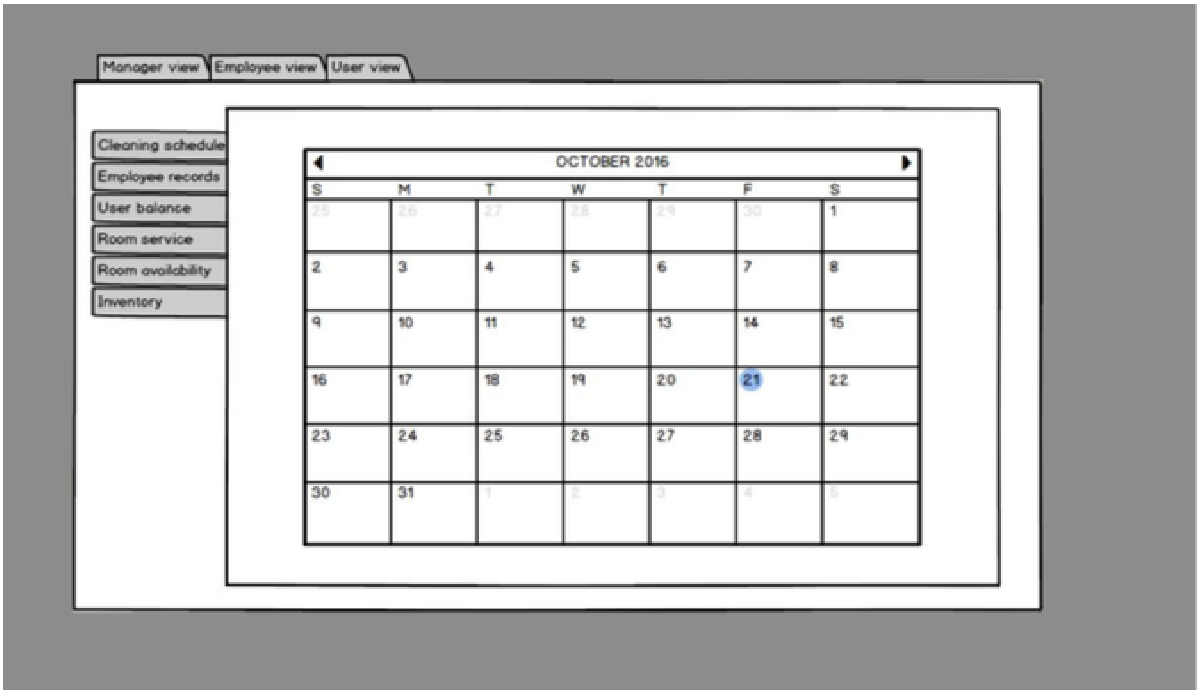
ALLIANCE HOTEL MANAGEMENT

By: SunRay MGMT



OCTOBER 2016

S	M	T	W	T	F	S
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Team Members: Jared Ariche, Heenali Patel, Jerry Vu,
Vontez Heard, Andrew Bolaji

Date: April 24,2017

CSC 4350 Spring 2017

Table of Contents

Case Study Problem Statement: Alliance Hotel Management (AHM)

1.0 AHM-Introduction:

The Alliance Hotel is a lively, new hotel opening up in downtown Atlanta. This new hotel is destined for greatness with a focus of high quality rooms at a low cost for consumers. Management has hired us, SunRay Management, to create a hotel management system that will efficiently handle booking rooms. The application will also serve as a platform for consumers to order room service. The application will be called Basic Alliance System. The BAS software will bridge the gap between room service, hotel booking, management and the consumer in order to improve the check-in process as well as room service. We at SunRay Management are working to ensure that your day shines.

1.1 Login/Logoff

All users (managers, desk attendants, housekeepers, and guests) shall have a separate login that will give each user different capabilities within the hotel.

AHM – Guest Privileges

2.0 Once a guest has called, gone online, or walked into the hotel and reserved a room, they shall automatically be provided with a user ID (last name) and password (room number) that allows them to request room service and access Wi-Fi information.

2.1 They shall be able to select items:

- A burger
- A drink
- More tissue
- Extra towel

That will be delivered by housekeeping.

2.2 This room service module shall be accessed on a tablet located inside of each room.

AHM – Employee Capabilities

3.0 Every employee shall be able to log into the AHM module.

3.1 This means that once they are finished with their shift, they log out.

Desk Attendant

3.2 Each desk attendant employee shall have access to view, book, modify and delete reservations. The desk attendants will handle all transactions.

3.3 Attendants shall also be able to view room service requests so that they can contact and notify the housekeeper of any pending requests that may happen to go unnoticed.

Housekeeper

3.4 Each housekeeping employee shall have the access to view, modify, and delete room service requests from each guest staying at the hotel.

3.5 Housekeepers shall also be able to view the inventory of items to ensure that all items are returned properly before the departure of a guest. This will also keep them informed about food that is or is not available.

3.6 Since the housekeepers will also handle all of the cleaning at the hotel, their log in to the application shall allow them to see (and delete once completed) maintenance orders that the managers may submit for them to do, such as taking out the trash or wiping down glass surfaces. This module will allow them to keep track of all manager and guest regrets.

Managers/Supervisor

3.7 All managers shall have the access to view all databases that are incorporated into the software, which means that they have the access to view everything to manage the entire hotel. They will monitor all reservations, requests, inventory, and employees of Alliance Hotel.

3.8 Managers shall have the sole privilege of hiring, firing, and scheduling all desk attendants and housekeepers of the hotel.

Alliance Hotel Management System: Requirements Traceability Matrix

Entry #	Paragraph	Requirement Description	Type	Use Case
1	1.1	All users shall have a separate login that will give each user different capabilities within the hotel.	SW	EmployeeLogin
2	2.0	They shall automatically be provided with a user ID (last name) and password (room number) that allows them to request room service and access Wi-Fi information.	SW	GuestRequestMeal
3	2.1	They shall be able to select items, such • A burger • A drink • More tissue • An extra towel • etc. That will be delivered by housekeeping.	SW	GuestRequestMeal
4	2.2	This room service module shall be accessed on a tablet located inside of each room.	SW	GuestRequestServices
5	3.0	Every employee shall log into the AHM module.	SW	Login
6	3.1	Employees shall log out at the end of their shift.	SW	Logout
7	3.2	Each desk attendant employee shall have access to view, book, modify and delete reservations.	SW	EmployeeView
8	3.3	Desk Attendants shall also	SW	GuestRequestServic

		be able to view room service requests.		es
9	3.4	Each housekeeping employee shall have the access to view, modify, and delete room service requests from each guest staying at the hotel.	SW	HousekeeperView
10	3.5	Housekeepers shall also be able to view the inventory of items to ensure that all items are returned properly before the departure of a guest.	SW	Inventory
11	3.6	Housekeepers shall allow them to see (and delete once completed) maintenance orders that the managers may submit for them to do, such as taking out the trash or wiping down glass surfaces.	SW	Admin
12	3.7	Managers shall have the access to view all databases that are incorporated into the software, for the hotel.	SW	HotelView
13	3.8	Managers shall have the sole privilege of hiring, firing, and scheduling all employees of the hotel.	SW	EmployeeSchedule

Use Cases:

Use Case 1:	**EmployeeLogin**
Participating Actors	Employee
Flow of Events	1. Employee selects Employee Login from home view. 2. Employee enters username and password into the prompted fields, or may select Cancel. If Cancel is selected, user is redirected to home view. 3. Employee clicks login button 4. After the Employee’s logins, their username and password will store in the database. 5. Page loads and displays the Employee view
Entry Condition	System must be at home view while no other users are logged in to the system.
Exit Condition	All values submitted are of correct type or user has selected Cancel.
Special Requirements	After submitting their credentials, values entered in the prompted fields must correspond to an active employee of type Admin in the database.

Rationale: The Employee must make sure they enter the correct Username and Password, for them to view their screen.

Use Case 2:	**GuestRequestMeal**
Participating Actors	Guest
Flow of Events	1. On the Guest main page, Guest will select the ‘Request Meal’ tab. 2. Guest can specify what they would like to order from the given menu. 3. Guest can press submit to complete the process be sent back to the Guest main page. 4. Guest can press submit to stop the process and be sent back to the Guest main page. 5. Once the Guest sends the order, they will receive a receipt with their meal display on the screen.
Entry Condition	On the Guest main page, guest will go to the ‘Request Meal’ screen.
Exit Condition	Guest can click ‘submit’ or click ‘cancel’ to be sent back to the Guest main page.
Special Requirements	The Kitchen supervisor will be able to monitor the orders placed and make sure they are delivered within the quoted time.

Rationale: The guest must make sure they choose the food according to what they desire and submit order so they are taken back to the main tab in case they want to add anything else they missed.

Use Case 3:	**GuestRequestServices**
Participating Actors	Guest
Flow of Events	1. On the Guest main page, Guest will select the ‘Request Services’ tab. 2. Guest can specify what services they require from the hotel. 3. Guest can press submit to complete the process and be sent back to the Guest main page 4. Guest can press cancel to stop the process and be sent back to the Guest main page 5. Once the Guest submits the services they want the housekeeper will be notified. 6. If anything is pending as in the service the guest wants the attendants will contact and notify the housekeeper.
Entry Condition	On the Guest main page, guest will go to the ‘Request Service’ Screen.
Exit Condition	Guest can click ‘submit’ or click ‘cancel’ to be sent back to the Guest main page.
Special Requirements	The Attendants will coordinate with housekeeping and make sure everything the guest requested is received and then making sure the guest’s are satisfied.

Rationale: The guest should make sure they have everything they asked for and mention if they need any more services.

Use Case 4:	**Login**
Participating Actors	Employee and Manger
Flow of Events	1. On the Employee main view, Employee will log in. 2. Type in their username and password. 3. Employee can press submit to validate their information. 3.1 An alert box will appear saying “Log in Sucessful”. 3.2 The Employee will press ‘Okay’ and be sent back the Employee main view.
Entry Condition	Clock-In/Clock-Out Tab on the Employee main view.
Exit Condition	Employee can click ‘submit’. If the number is correct, then Employee will press ‘Ok’ to be sent back to the Guest main page.
Special Requirements	The manager will make sure that the employees work according to their schedule and they don’t work overtime.

Rationale: Employees must check in as soon as they come in to prevent future conflicts, which arise when employees don’t sign in causing errors in their salary.

Use Case 5:	**Logout**
Participating Actors	Employee and Manger
Flow of Events	1. On the Employee main view, Employee will log out once their shift has been completed. 2. Type in their username and password. 3. Employee can press submit to validate their information. 3.1 An alert box will appear saying “Log out sucessful”. 3.2 The Employee will press ‘Okay’ and be sent back to the Employee main view.
Entry Condition	Clock-In/Clock-Out Tab on the Employee main view
Exit Condition	Employee can click ‘submit’. If the number is correct, then Employee will press ‘Ok’ to be sent back to the Guest main page.
Special Requirements	Managers make sure that each employee is given their check according to the hours they worked during the month.

Rationale: Employees must make sure they are done with all their duties before they check out. They should also check their schedule for the following day.

Use Case 6:	**EmployeeView**
Participating Actors	Employee
Flow of Events	1. On the Employee main view, Employee will go to the “New Guest” tab. 2. The Employee will type in the guest information as in name, address, and payment type. 3. The Guest can tell the desk attendants which room they prefer to have. 4. The payment will then be process. 5. Employee can click ‘submit’ which will print/display a receipt for the guest.
Entry Condition	Employee will go to the “New Guest” tab.
Exit Condition	Employee can click ‘submit’ which will print a receipt for the guest.
Special Requirements	The desk attendants will handle all transactions.

Rationale: The Employee has access to view, book, modify and delete reservations.

Test Cases:

Functional Test Case 1:	**EmployeeLogin_DeskAttendant**
Test Location	C:\Users\jaric\OneDrive\Documents\NetBeansProjects\Alliance\src
Features to be Tested	Employee enters login credentials into the system; Desk Attendant UI will be onscreen.
Features Pass/Fail Criteria	Test passes if Desk Attendant UI comes to view after login credentials are submitted.
Means of Control	WelcomeName() class will be called once login credentials are entered and login button is pressed.
Data	1. Username 2. Password Information is matched with information within the database.
Test Procedure	User enters login credentials into jtxtfield boxes. If information is matched within the database, then Desk Attendant UI appears.
Special Requirements	Manager needs to specify the employee occupation with a D in employee database in order for proper UI to appear.

Rationale: Manager must make sure employee is saved into the database with correct occupation so that proper UI appears. Also employee must make sure they have the correct login credentials.

Functional Test Case 2:	**EmployeeLogin_HousingRequest**
Test Location	C:\Users\jaric\OneDrive\Documents\NetBeansProjects\Alliance\src
Features to be Tested	Employee enters login credentials into the system; Housing Request UI will be onscreen.
Features Pass/Fail Criteria	Test passes if House Request UI comes to view after login credentials are submitted.
Means of Control	HouseKeep() class will be called once login credentials are entered and login button is pressed.
Data	1. Username 2. Password Information is matched with information within the database.
Test Procedure	User enters login credentials into jtxtfield boxes. If information is matched within the database, then HouseKeep UI appears.
Special Requirements	Manager needs to specify the employee occupation with a H in employee database in order for proper UI to appear.

Rationale: Manager must make sure employee is saved into the database with correct occupation so that proper UI appears. Also employee must make sure they have the correct login credentials.

Functional Test Case 3:	**EmployeeLogin_Admin**
Test Location	C:\Users\jaric\OneDrive\Documents\NetBeansProjects\Alliance\src
Features to be Tested	Employee user enters login credentials into the system, Administrator UI will be onscreen.
Features Pass/Fail Criteria	Test passes if Admin UI comes to view after login credentials are submitted.
Means of Control	Admin() class will be called once login credentials are entered and login button is pressed.
Data	1. Username 2. Password Information is matched with information within the database
Test Procedure	User enters login credentials into jtxtfield boxes. If information is matched within the database, then Administrator UI appears.
Special Requirements	Manager needs to specify the employee occupation with an A in employee database in order for proper UI to appear. Software comes initially with.

Rationale: Manager must make sure employee is saved into the database with correct occupation so that proper UI appears. Also employee must make sure they have the correct login credentials.

Functional Test Case 4:	**GuestLogin**
Test Location	C:\Users\jaric\OneDrive\Documents\NetBeans Projects\Alliance\src
Features to be Tested	Guest enters login credentials into the system; Guest Room Service UI will be onscreen.
Features Pass/Fail Criteria	Test passes if Guest Room Service UI comes to view after login credentials are submitted.
Means of Control	roomservices() class will be called once login credentials are entered and login button is pressed
Data	1. Last Name 2. Room Number Information is matched with information within the database
Test Procedure	User enters login credentials into jtxtfield boxes. If information is matched within the database, then room service UI appears.
Special Requirements	Guest must have booked a room and be in the database in order to have access to database.

Rationale: Guest must make sure they enter the correct login credentials.

Functional Test Case 5:	**ServiceRequest_DeleteRequest**
Test Location	C:\Users\jaric\OneDrive\Documents\NetBeansProjects\Alliance\src
Features to be Tested	Employee will be able to view service request sent to the housing database by guest and delete the request once the request has been tended to.
Features Pass/Fail Criteria	Test passes if request is deleted off screen and from the database if the delete button is pressed.
Means of Control	Delete Method()
Data	Data that populates JTable is pulled from the databases HouseKeeping table.
Test Procedure	User selects which request to delete by clicking on the request to be deleted from the JTable and hitting the delete button.
Special Requirements	Information needs to be in HouseKeeping table in database in order to view and delete the request.

Rationale: Housekeeping must make sure that the request has been tended to before deleting it from the system.

Functional Test Case 6:	**ServiceRequest_UpdateRequest**
Test Location	C:\Users\jaric\OneDrive\Documents\NetBeansProjects\Alliance\src
Features to be Tested	Employee will be able to view service request sent to the housing database by guest and update the table if new request has been sent to the database.
Features Pass/Fail Criteria	Test passes if new request that have been sent to the database are viewed once update button is pressed.
Means of Control	Update() method
Data	Data that populates JTable is pulled from the databases HouseKeeping table.
Test Procedure	User presses update button and new request that have been submitted to the database will populate the JTable.
Special Requirements	HouseKeeping table is needed in order to view requests

Rationale: Housekeeping must hit the button every 10 minutes or once they finish a service request.

Functional Test Case 7:	**BookRoom**
Test Location	C:\Users\jaric\OneDrive\Documents\NetBeansProjects\Alliance\src
Features to be Tested	Employee will enter guest information into the UI which will then be sent to sqlite database.
Features Pass/Fail Criteria	Test passes if information is sent to the BookRoom table in the Alliance database once submit button is pressed.
Means of Control	Information is obtained through textfields and dropdown selection boxes. Information is submitted once continue button is pressed only if information has been entered into textfields.
Data	The information collected from the UI is as follows: First Name, Last Name, Address, City, Country, room type, Smoking Allowed, Number of Beds, Check In Date, Check Out Date
Test Procedure	1.Enter information into UI. 2.Press Continue button.
Special Requirements	Application needs to be connected to database in order to store guest information. Also, user must at least fill out the first name box or information from UI will not be entered into the database.

Rationale: Desk Attendants must ensure that they enter the correct information into the UI, especially the last name so that the guest will have access to room service application.

Functional Test Case 8:	**DeskAttendant_HomePage**
Test Location	C:\Users\jaric\OneDrive\Documents\NetBeansProjects\Alliance\src
Features to be Tested	Employee will have the option to book a room for a guest, check status of room, logout the system dependent on which button is pressed.
Features Pass/Fail Criteria	Test fails if UI that corresponds with each button comes to view once it the button is pressed.
Means of Control	Three buttons will control which UI opens. Book Room button will call BookRoom() class UI, Check Status button will call Status() class UI, and Logout button will call Login() class UI.
Data	No data is needed for this test case.
Test Procedure	1.Press Book Room button 2.Press Check Status button 3.Press Logout button
Special Requirements	Login() class UI must be successfully validated for user to have access to DeskAttendant() class UI.

Rationale: Employee will have the option choose make three separate decisions based on the menu items presented.

Rationale:

The Alliance Hotel, opening up in downtown Atlanta this May, offers premium four-star hotel service. The hotel is set to have 3 floors, with 100 rooms per floor. This adds up to 300 rooms total. The rooms will be divided into two sections: Smoking and Non-Smoking. These two sections will be divided into 75 Single rooms and 75 Double rooms. Check-in times are the standard 3 o'clock, and checkout times are the standard 11 O'clock. With 300 rooms, constantly being left and then reoccupied, it is essential that our hotel management system updates in real- time. Double booking rooms can lead to a bad reputation, a loss of money, or even a confused staff. But our fool proof Hotel Management System will never fail.

The Hotel Management System will be linked to a database. The Basic Alliance System must comply with Alliance Hotel's check-in services. The hotel management
System will be loaded onto the Alliance Check-in Service's desktop computers. The room service software will be loaded onto a kiosk and placed in each room of the hotel. The room service software will be strictly for the user and the hotel management system will be strictly for hotel staff.

When people check in, their name and room number they are in will be added to the database, as well as how long they will be staying, and how many guests are with them. Upon receiving this information, the room given back by the system will suit their needs. It will also be occupied in the database, so whoever comes along will not be booked into the same room, causing a disaster.

The system will also notify the staff which room needs to be cleaned, since a guest has just vacated it. Until housekeeping logs the room into the database as cleaned, no guests will be checked into that room. It is essential that a management system has different access areas for different users, either costumer, staff, or management. Different user permissions are set by using passkeys, so that regular customers cant access information meant to be private to management. Employees will have a limited view in comparison to that of management.

The information provided above is in line with all the customer requirements specified by SunRay Hotels, before the system was produced.

Use Case Rationale

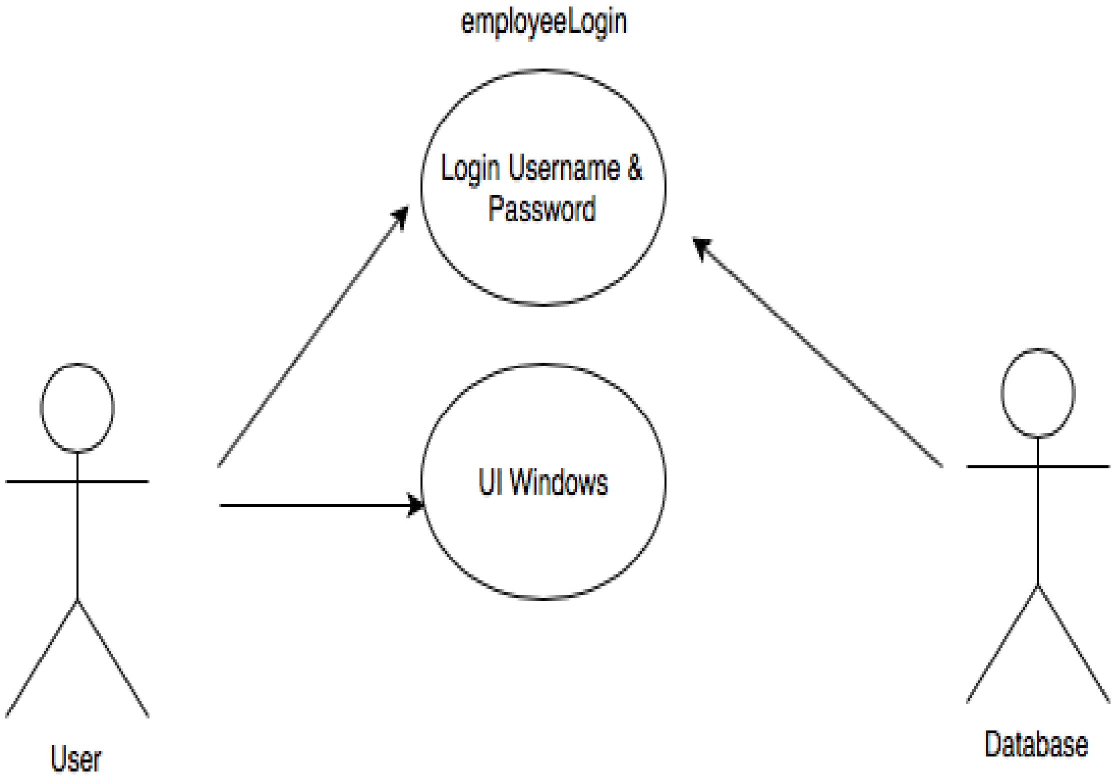
1. The Employee must make sure they enter the correct Username and Password, for them to view their screen.
2. The guest must make sure they choose the food according to what they desire and submit order so they are taken back to the main tab in case they want to add anything else they missed.
3. The guest should make sure they have everything they asked for and mention if they need any more services.
4. Employees must check in as soon as they come in to prevent future conflicts, which arise when employees don’t sign in causing errors in their salary.
5. Employees must make sure they are done with all their duties before they check out. They should also check their schedule for the following day.
6. The Employee has access to view, book, modify and delete reservations.
7. The hotelView has access to the entire database that incorporated into the software.
8. The Inventory will always be updated on a certain date, which will allow the supplies to be delivered.

Test Case Rationale

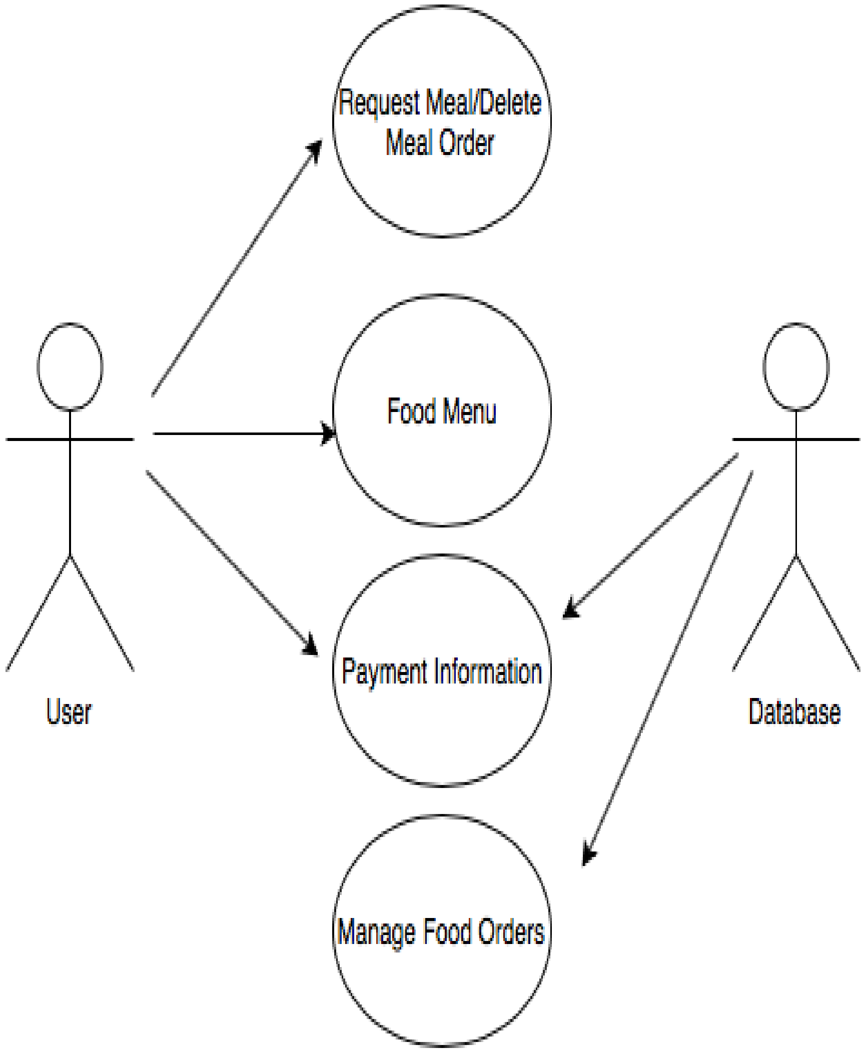
1. Upon opening of application, all employees must then be able enter their log in information into the text fields and then click the log in button, which will prompt a popup window stating if the log in was successful or unsuccessful.
2. After clicking okay, the employee should see their corresponding UI to complete job tasks.
3. All Managers must be able to book rooms, manage employees, or view service requests with the click of a button.
4. Managers are the only employees that can manage other employees.
5. Managers must be able to add, update, and delete any employee with specific info in text fields.
6. Desk attendants must be able to book a room or check status of room once they have logged in.
7. When booking a room, a manager or desk attendant must be able to enter a guest's information into available text fields.
8. They must ensure that they enter the correct information into the UI, especially the last name so that the guest will have access to room service application.
9. Managers/Desk Attendants also must be able to specify room type, smoking capability, and number of beds through the selection of a combo box.
10. Once all of a guests' information has been inserted, the employee shall be able to click the continue button which shall open a payment information window so that payment can be received.
11. The manager/desk attendant should be able to enter payment information into text fields.
12. The manager/desk attendant should be able to specify payment method through selection of combo box.
13. Once this information is inserted, the employee shall click pay now and the amount due should come up at the top of the window.
14. While on the payment information window, the employee shall be able to go back to the book room or home menu.
15. When desk attendants click the check status button on the home/welcome window, they should be sent to the room status UI.

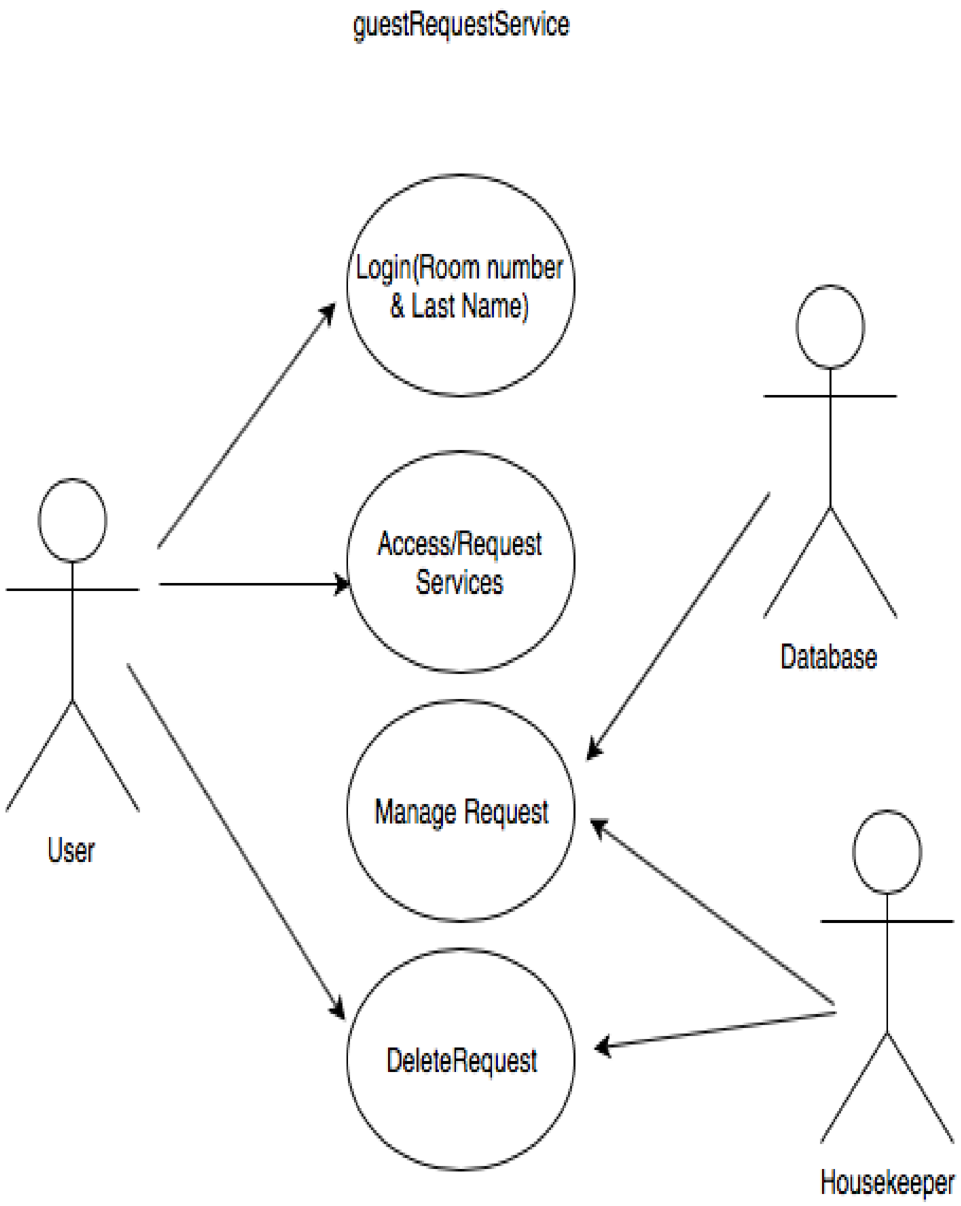
16. On the room status UI, they will enter a first name and last name and click search customer so that subsequent room status and information will appear on the screen. They will also be able to go back to the home window on this UI.
17. Housekeeping must be able to select any need(s) and delete it from the system and subsequently from the database.
18. Housekeeping must also be able to update (refresh) the requests table to see if any new requests have populated from guests.
19. Once a housekeepers job is done, they shall be able to log out.
20. Guests should automatically be populated into the database once they have booked a room. This will allow them to log into the room service module located on a kiosk in their rooms.
21. Once a guest has logged into the room service module, they shall be sent to a room service UI where various needs/services and food/beverage items will be available for them to select from.
22. They shall be able to single click a item and click the select button or double click an item.
23. Once a need is selected, a guest will then be prompted to specify quantity.
24. Once a food is selected, the guest will be prompted to specify type and then quantity.
25. A guest should be able to remove an selected item or items from the table at any time using the remove button.
26. When the user had selected specifications, the selected items should populate in the table below.
27. Once they are finished selecting items, the user must then be able to enter room number in the appearing text field and then hit submit, which will clear table and send requests to the database where housekeepers can receive the update on their end.
28. There is also a restaurants nearby button that will send users to a UI with pictures of nearby restaurants.
29. Once requests have been submitted, guests shall be able to log out.

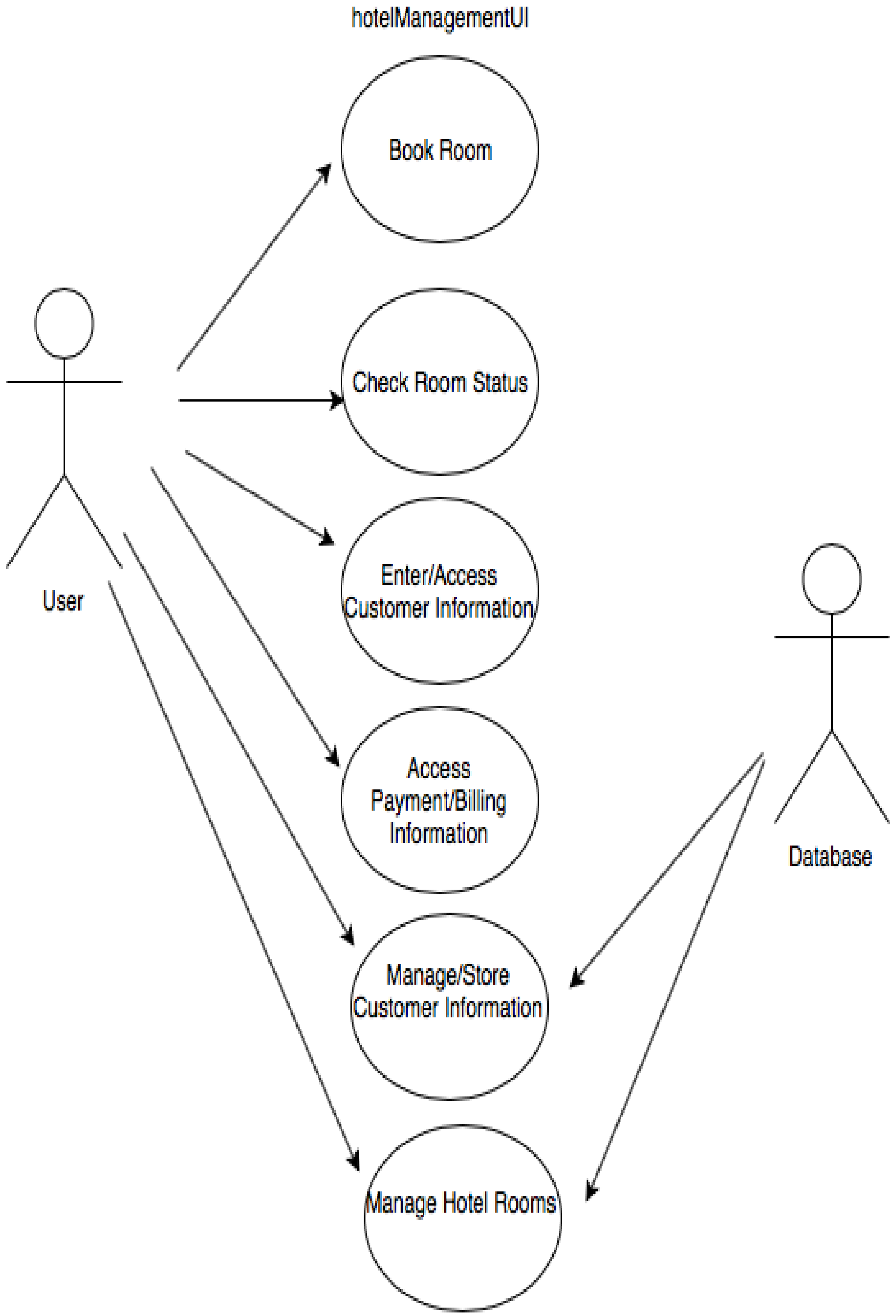
Use Case Diagrams:

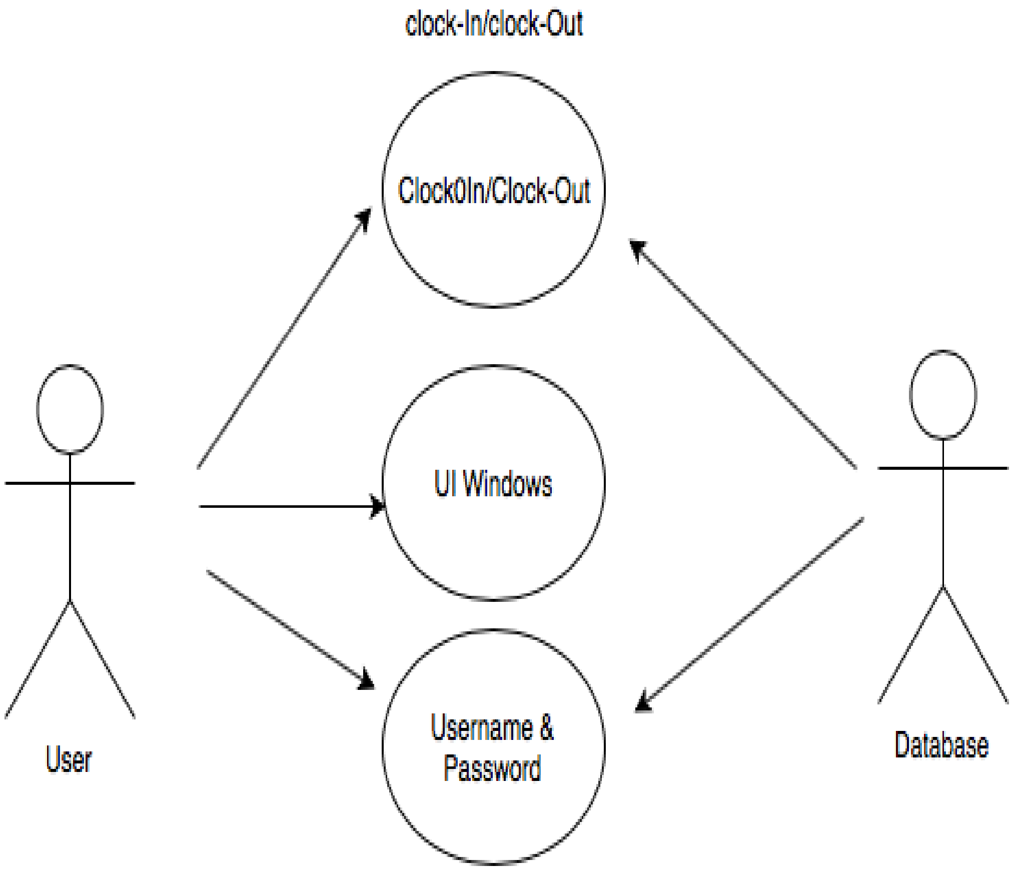


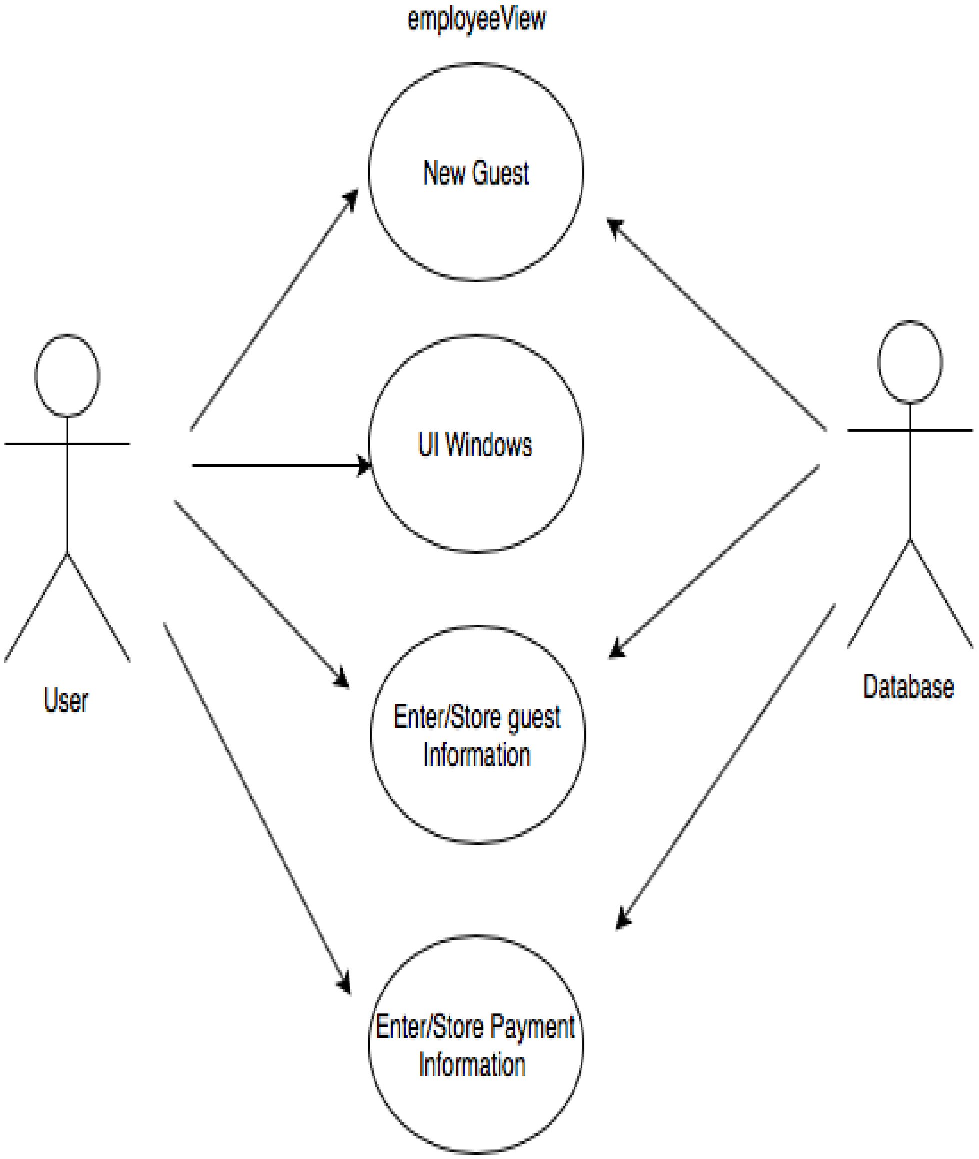
guestRequestMeal

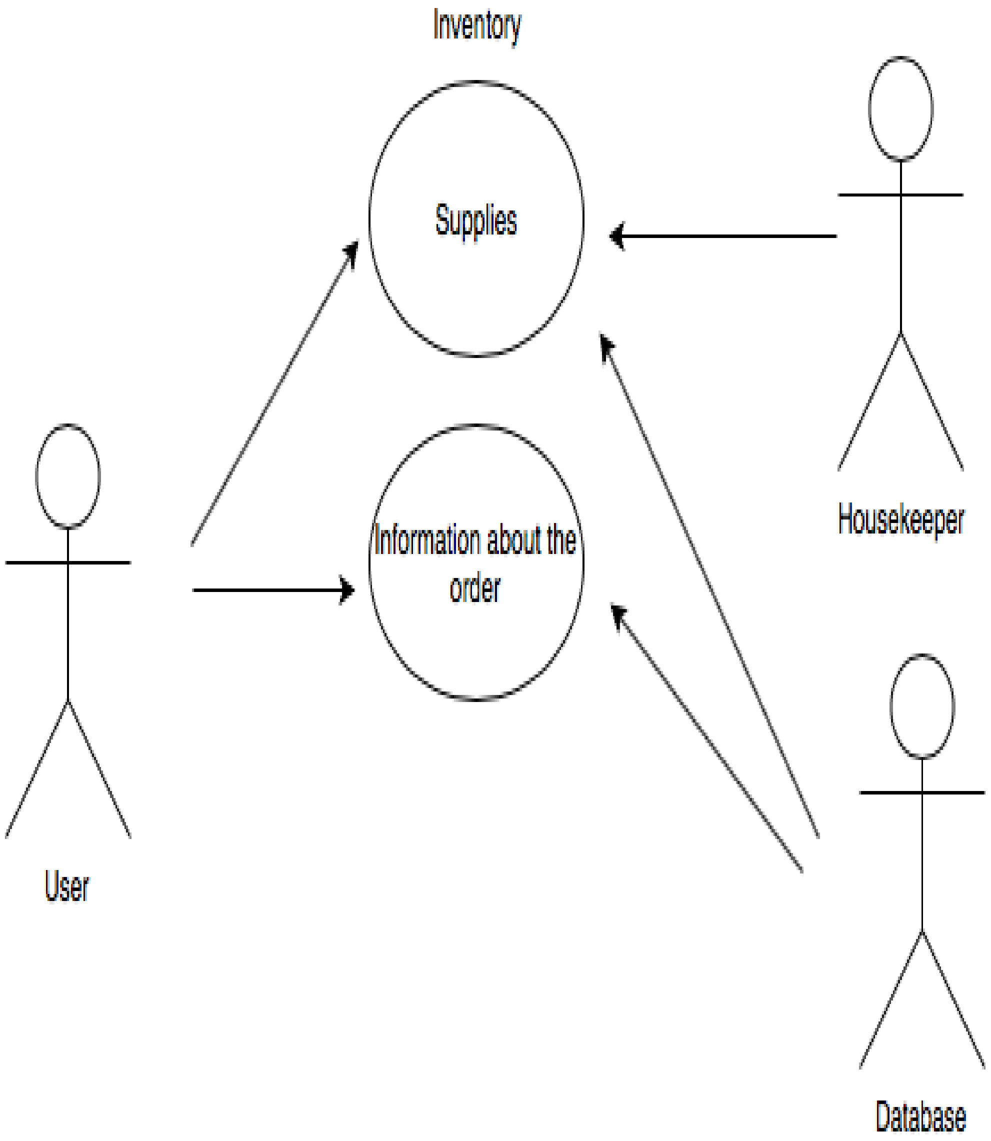


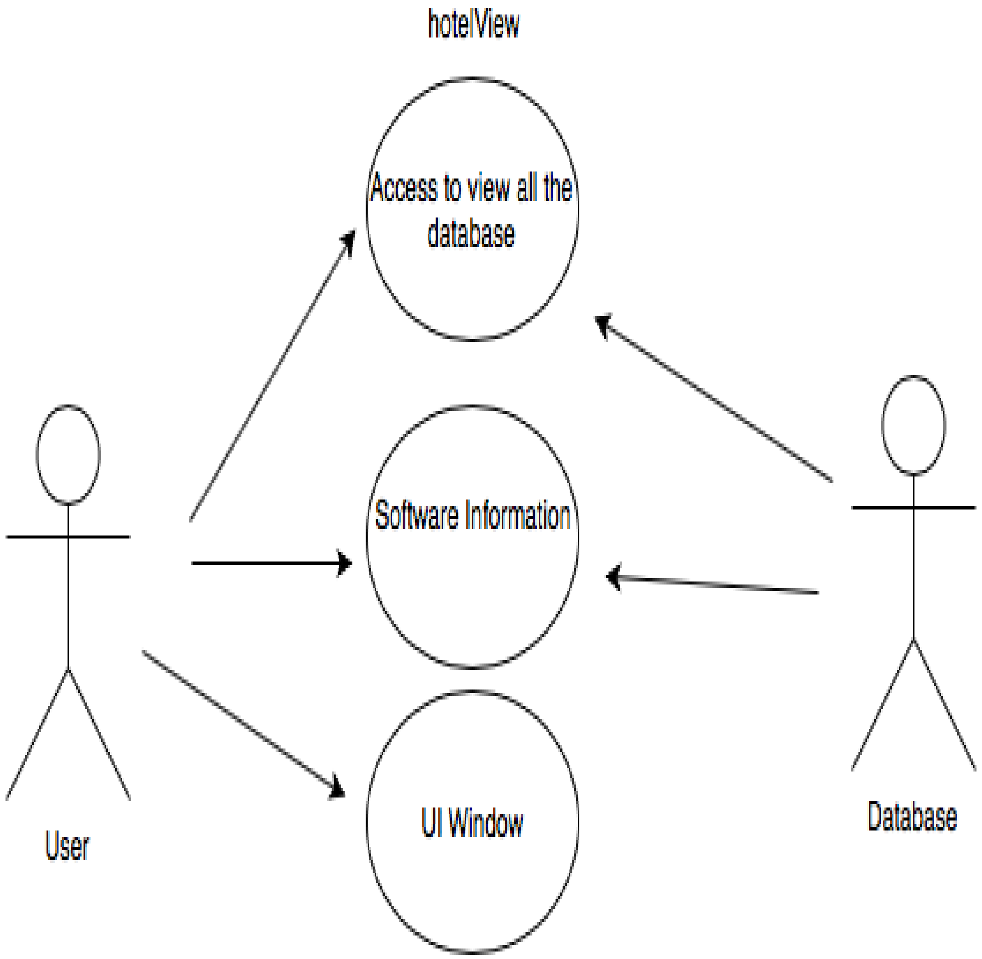

guestRequestService
Login(Room number & Last Name)
Access/Request Services
Manage Request
DeleteRequest
User
Database
Housekeeper

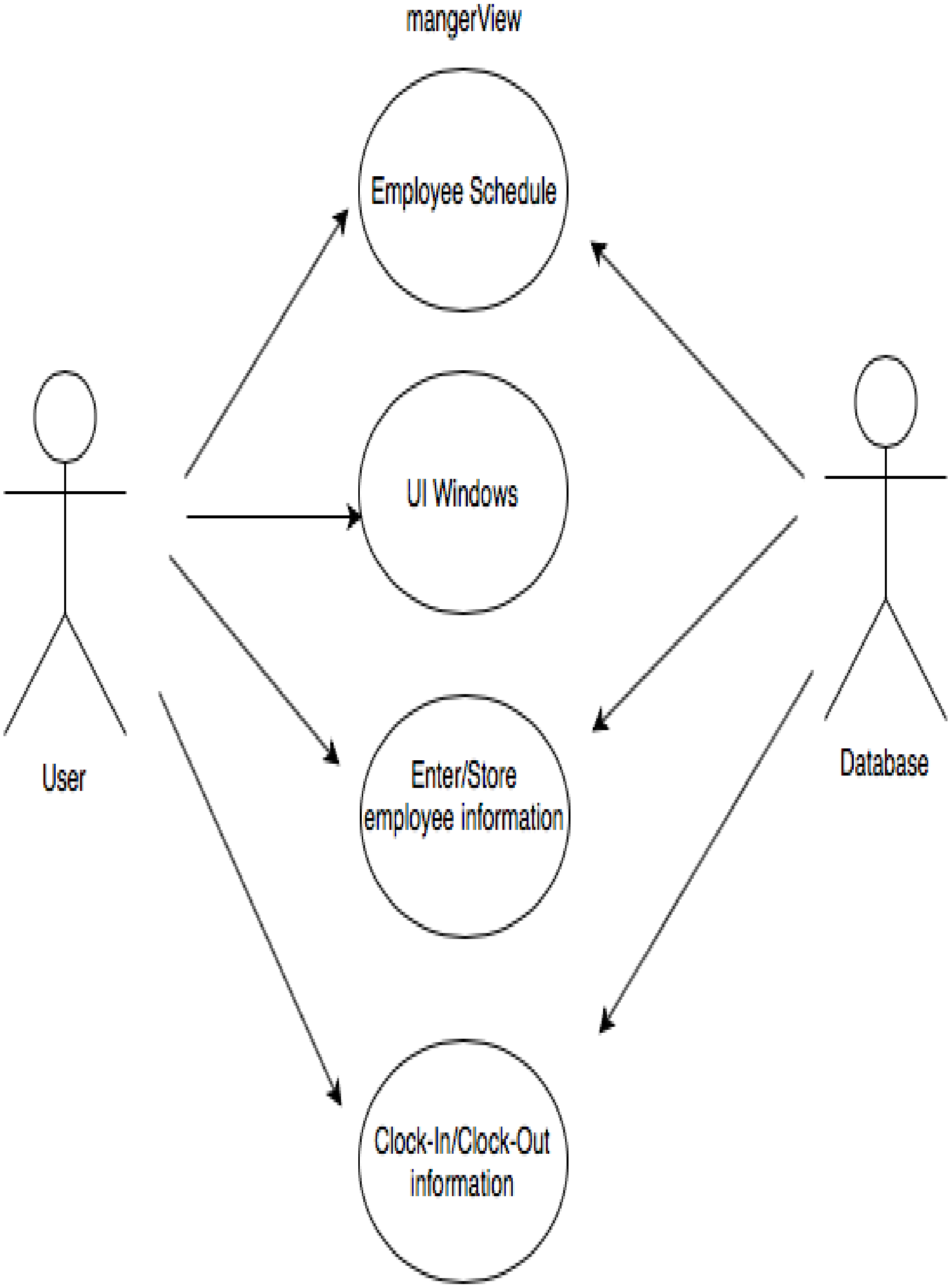

hotelManagementUI
Book Room
Check Room Status
Enter/Access Customer Information
Access Payment/Billing Information
Manage/Store Customer Information
Manage Hotel Rooms
User
Database


clock-In/clock-Out
Clock0In/Clock-Out
UI Windows
Username & Password
User
Database

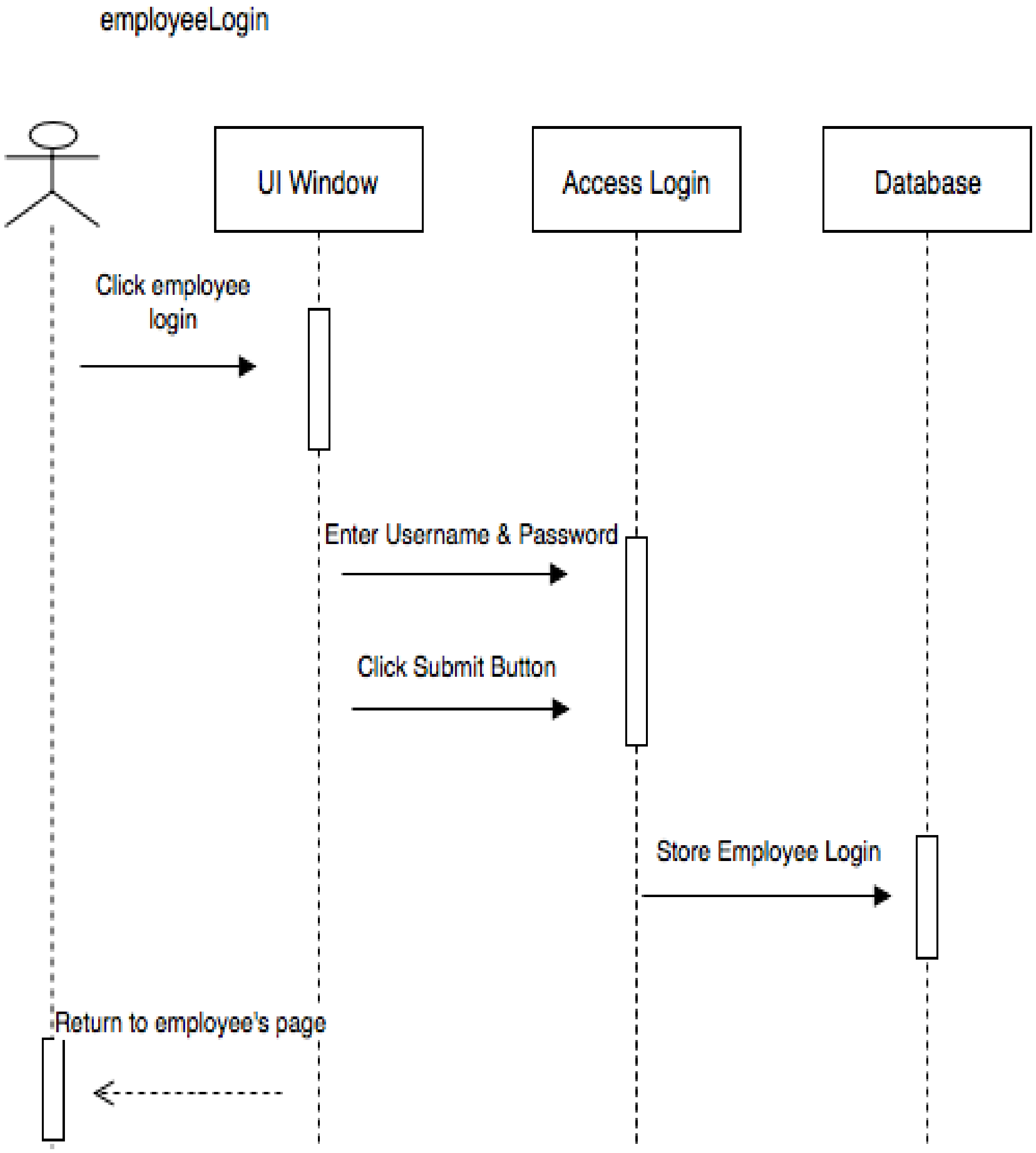

employeeView
New Guest
UI Windows
Enter/Store guest Information
Enter/Store Payment Information
User
Database


Inventory
Supplies
Information about the order
User
Housekeeper
Database

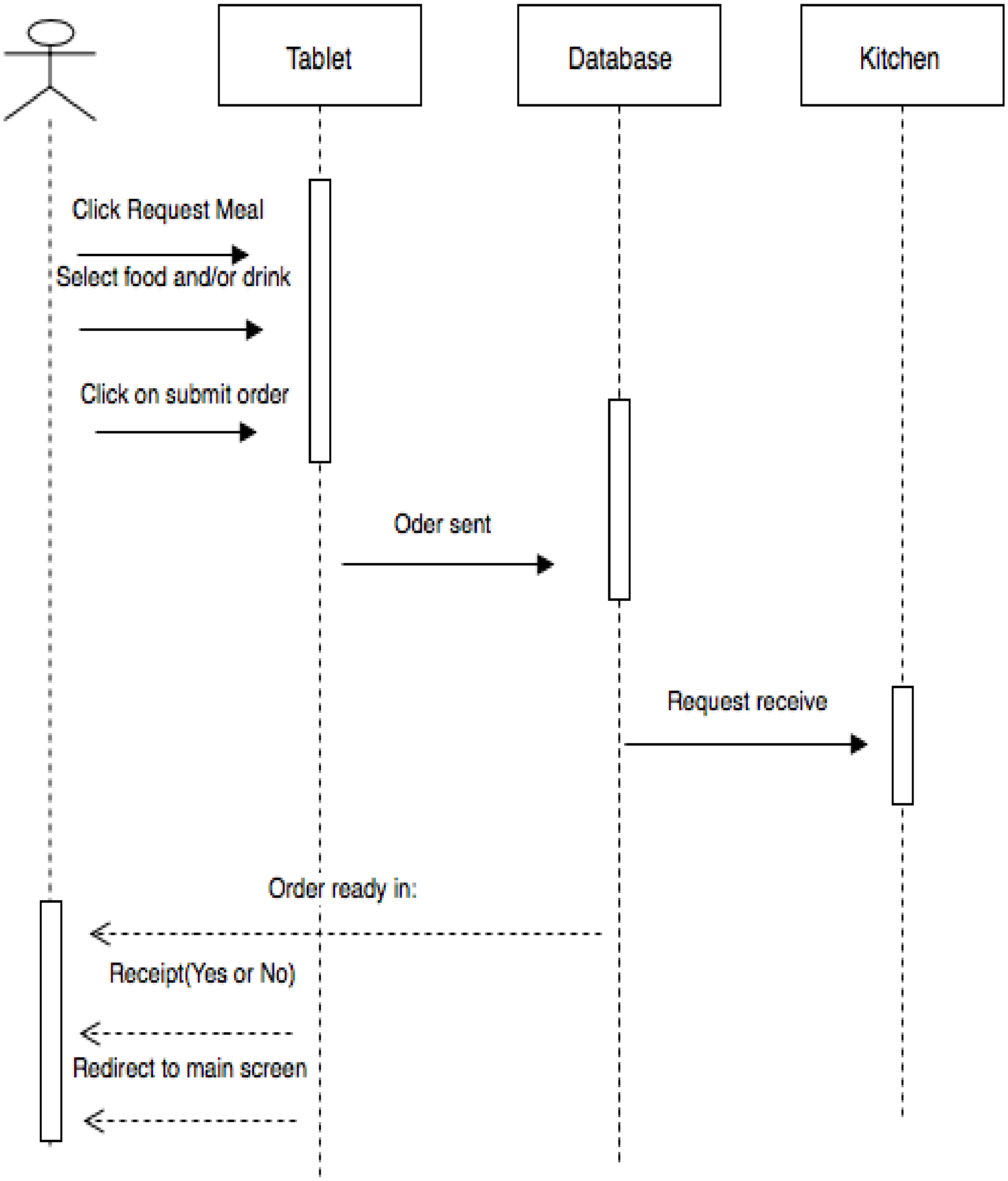

hotelView
Access to view all the database
Software Information
UI Window
User
Database

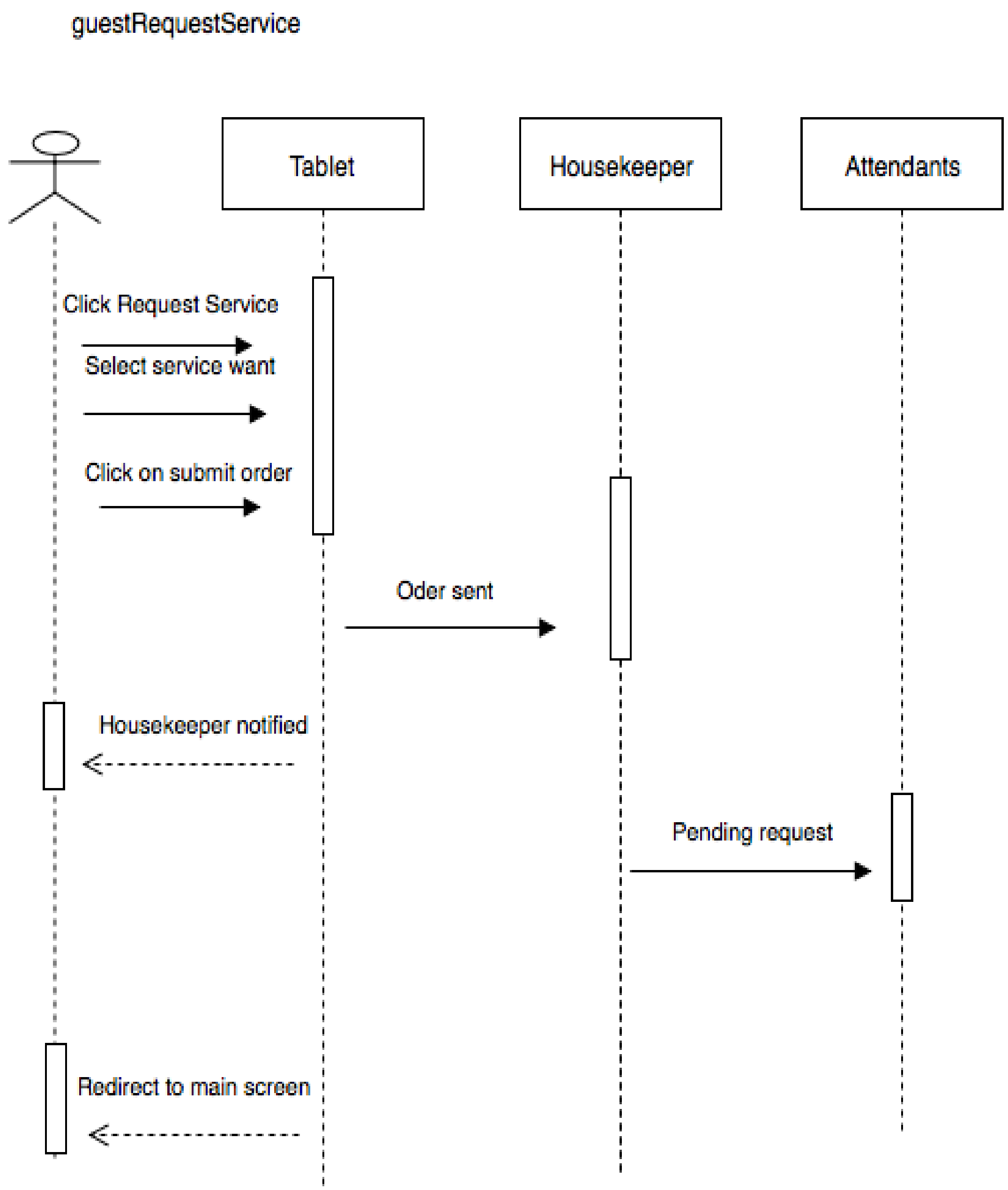

mangerView
Employee Schedule
UI Windows
Enter/Store employee information
Clock-In/Clock-Out information
User
Database

Sequence/Interaction Diagram:

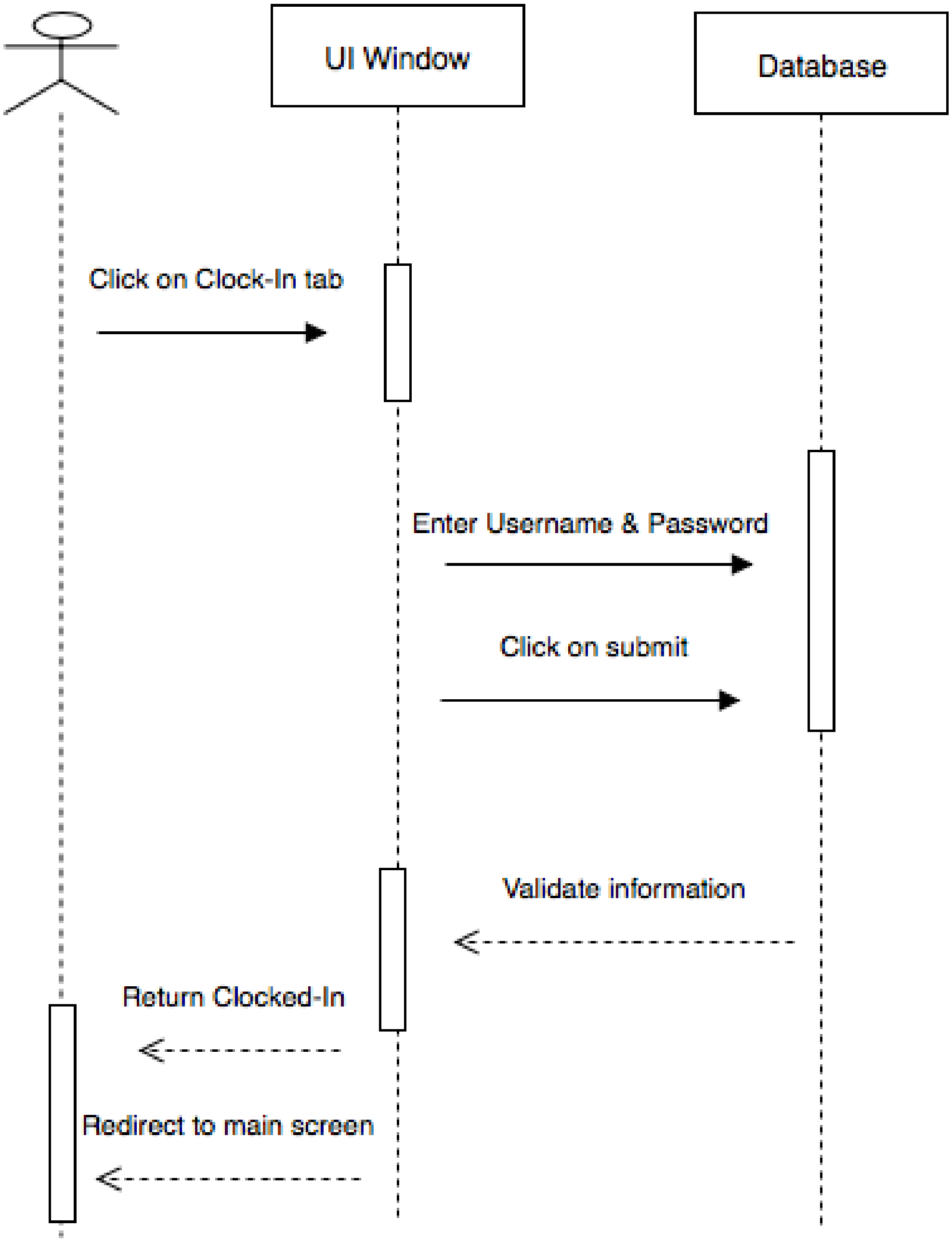


guestRequestMeal

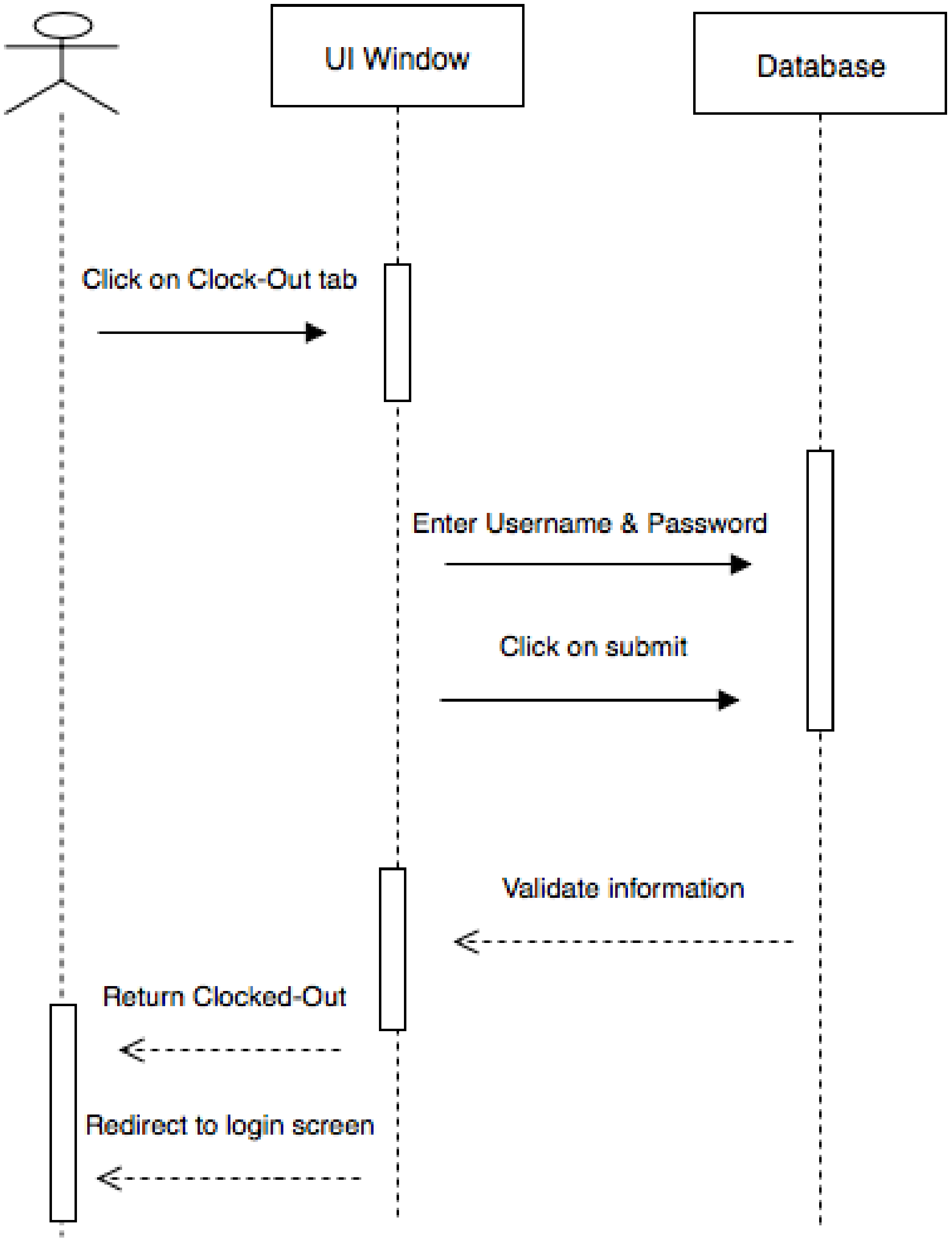



guestRequestService
Tablet
Housekeeper
Attendants
Click Request Service
Select service want
Click on submit order
Oder sent
Housekeeper notified
Pending request
Redirect to main screen

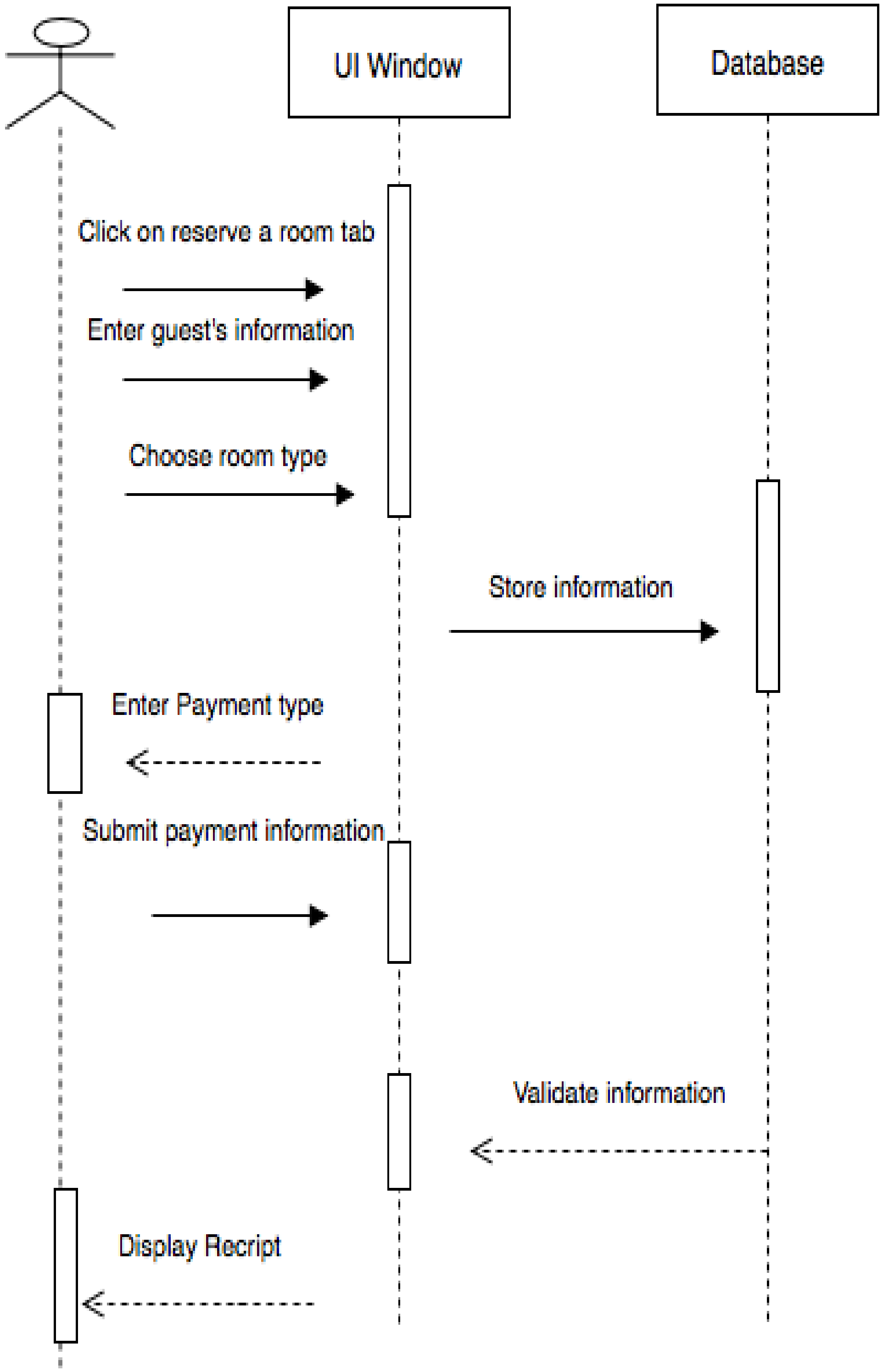
clock-In

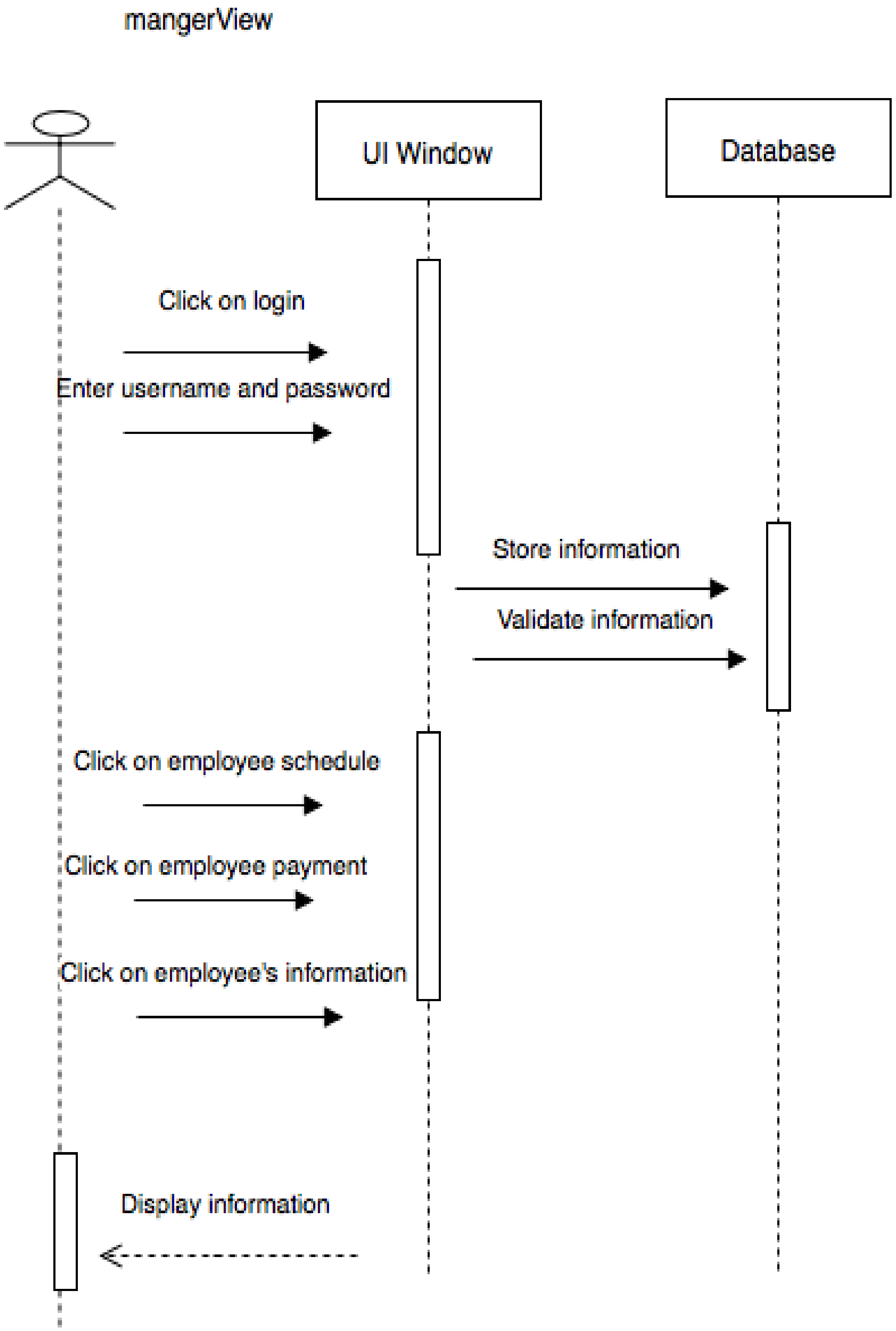

UI Window
Database
Click on Clock-In tab
Enter Username & Password
Click on submit
Validate information
Return Clocked-In
Redirect to main screen

clock-Out


UI Window
Database
Click on Clock-Out tab
Enter Username & Password
Click on submit
Validate information
Return Clocked-Out
Redirect to login screen

employeeView




mangerView
UI Window
Database
Click on login
Enter username and password
Store information
Validate information
Click on employee schedule
Click on employee payment
Click on employee's information
Display information

Function Point Cost Analysis:

A software package is to be developed for a Hotel Management System

The Hotel Management System will contain and manage a hotel with 300 rooms. The software will allow employees to book and reserve rooms for customers and check customers out. The GUI will also allow the manipulation of the Hotel Management System's database. The program will have the ability to query for previously existing customers in the database. A room service application will be incorporated into the system as well to improve customer service efficiency. The software is designed to help with the management process of a hotel by keeping things organized.

After analyzing the requirements we ended up with the following Weighted Factor Estimation. We aim to keep it within the average weighting factor

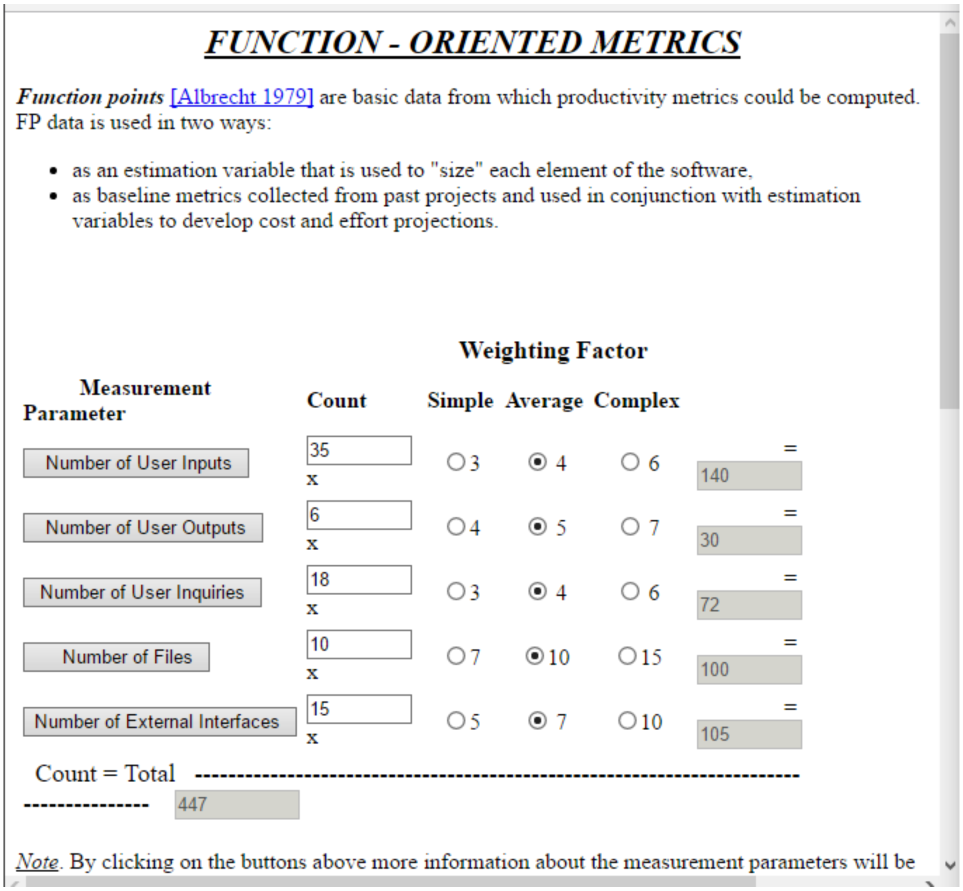
Measurement Parameter	Count	Simple Average Complex	Quantity
Number of user inputs	35	4	140
Number of user outputs	6	5	30
Number of user inquires	18	4	72
Number of files	10	10	100
Number of external interfaces	15	7	105

Total: 447

Factor	**Value**
Does the System Require Reliable Backup and Recovery	4
Are data communications required	5
Are there distributed processing functions	0
Is performance critical	4
Will the system run in an existing, heavily utilized operational environment	4
Does the system require on-line data entry	2

Does the on-line data entry require the input transaction to be built over multiple screens or operations	3
Are the master files updated on-line	2
Are the inputs outputs files or inquiries complex	2
Is the internal processing complex	5
Is the code reusable	4
Are conversion and installation included in the design?	0
Is the system designed for multiple installations in different organizations	4
Is the application designed to facilitate change and ease of use by the user	3

478 FP



FUNCTION - ORIENTED METRICS

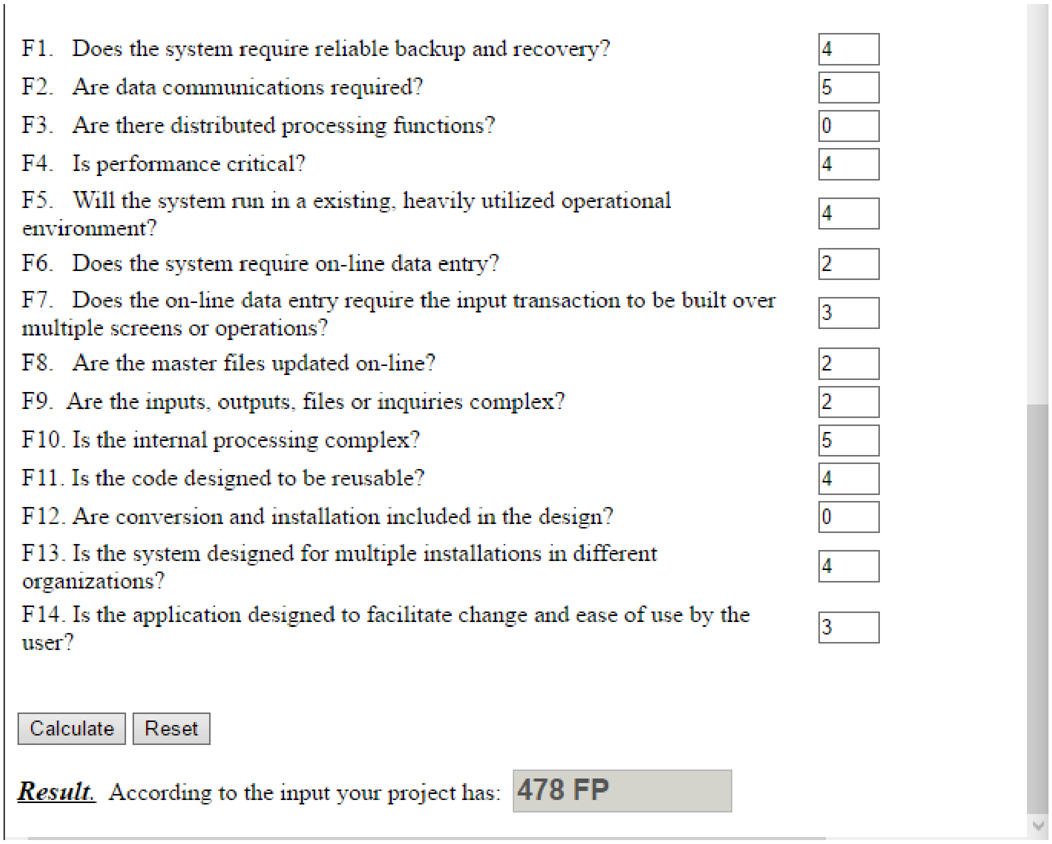
Function points [Albrecht 1979] are basic data from which productivity metrics could be computed. FP data is used in two ways:

- as an estimation variable that is used to "size" each element of the software,
- as baseline metrics collected from past projects and used in conjunction with estimation variables to develop cost and effort projections.

			Weighting Factor				
Measurement Parameter	Count		Simple	Average	Complex		
Number of User Inputs	35	x	○ 3	● 4	○ 6	=	140
Number of User Outputs	6	x	○ 4	● 5	○ 7	=	30
Number of User Inquiries	18	x	○ 3	● 4	○ 6	=	72
Number of Files	10	x	○ 7	● 10	○ 15	=	100
Number of External Interfaces	15	x	○ 5	● 7	○ 10	=	105

Count = Total -- --------------- 447

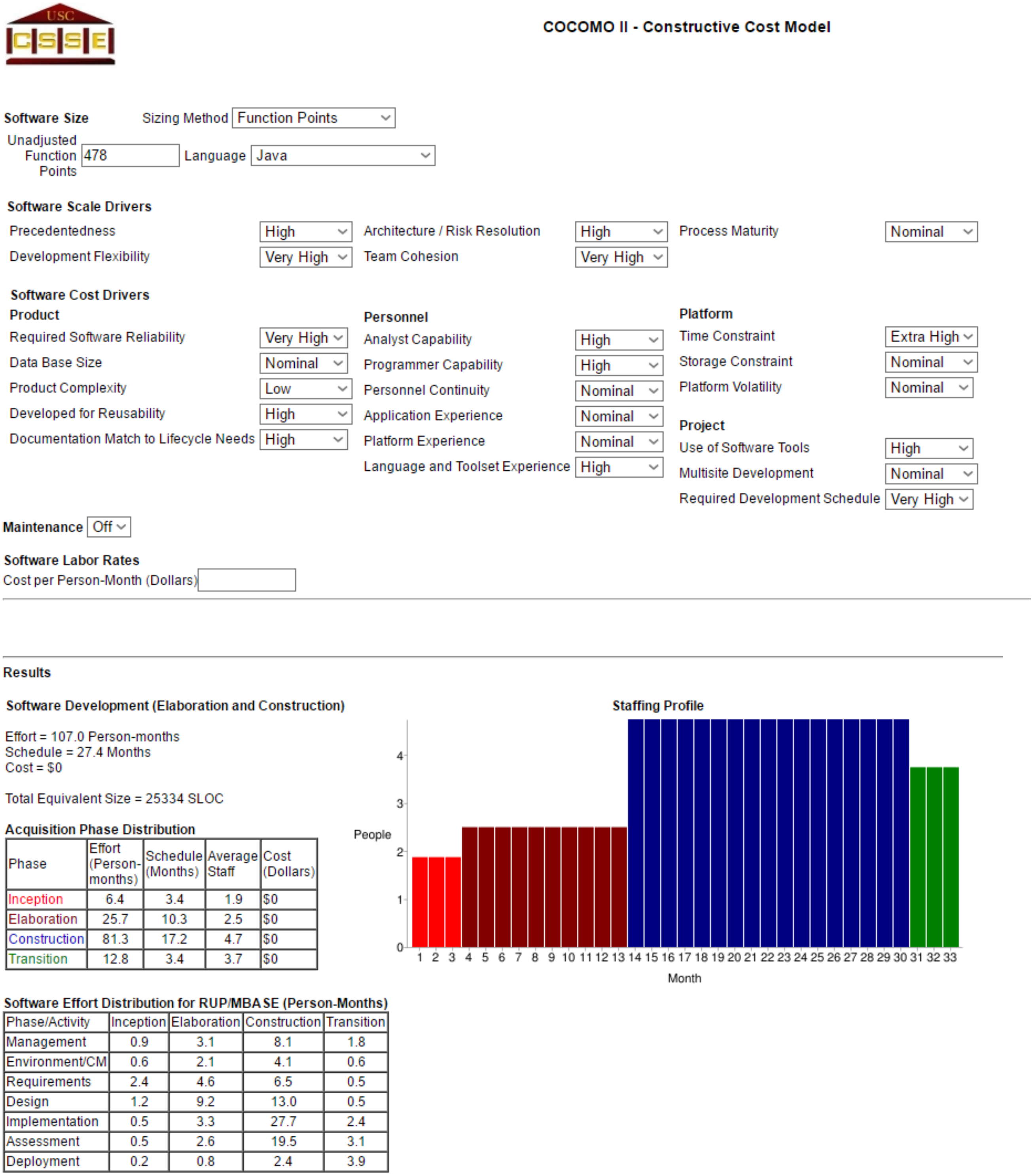
Note. By clicking on the buttons above more information about the measurement parameters will be



F1. Does the system require reliable backup and recovery?	4
F2. Are data communications required?	5
F3. Are there distributed processing functions?	0
F4. Is performance critical?	4
F5. Will the system run in a existing, heavily utilized operational environment?	4
F6. Does the system require on-line data entry?	2
F7. Does the on-line data entry require the input transaction to be built over multiple screens or operations?	3
F8. Are the master files updated on-line?	2
F9. Are the inputs, outputs, files or inquiries complex?	2
F10. Is the internal processing complex?	5
F11. Is the code designed to be reusable?	4
F12. Are conversion and installation included in the design?	0
F13. Is the system designed for multiple installations in different organizations?	4
F14. Is the application designed to facilitate change and ease of use by the user?	3

Result. According to the input your project has: **478 FP**

COCOMO Model:



COCOMO II - Constructive Cost Model

Software Size Sizing Method Function Points

Unadjusted Function Points 478 Language Java

Software Scale Drivers

Precedentedness	High	Architecture / Risk Resolution	High	Process Maturity	Nominal
Development Flexibility	Very High	Team Cohesion	Very High		

Software Cost Drivers

Product		**Personnel**		**Platform**	
Required Software Reliability	Very High	Analyst Capability	High	Time Constraint	Extra High
Data Base Size	Nominal	Programmer Capability	High	Storage Constraint	Nominal
Product Complexity	Low	Personnel Continuity	Nominal	Platform Volatility	Nominal
Developed for Reusability	High	Application Experience	Nominal	**Project**	
Documentation Match to Lifecycle Needs	High	Platform Experience	Nominal	Use of Software Tools	High
		Language and Toolset Experience	High	Multisite Development	Nominal
				Required Development Schedule	Very High

Maintenance Off

Software Labor Rates

Cost per Person-Month (Dollars)

Results

Software Development (Elaboration and Construction)

Effort = 107.0 Person-months
Schedule = 27.4 Months
Cost = $0

Total Equivalent Size = 25334 SLOC

Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	6.4	3.4	1.9	$0
Elaboration	25.7	10.3	2.5	$0
Construction	81.3	17.2	4.7	$0
Transition	12.8	3.4	3.7	$0

Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.9	3.1	8.1	1.8
Environment/CM	0.6	2.1	4.1	0.6
Requirements	2.4	4.6	6.5	0.5
Design	1.2	9.2	13.0	0.5
Implementation	0.5	3.3	27.7	2.4
Assessment	0.5	2.6	19.5	3.1
Deployment	0.2	0.8	2.4	3.9

User Guide:

Any user who chooses to use our application will quickly find out that it is very easy to use. We have provided the users with a simple, yet serviceable application that make the daily responsibilities of managing a hotel extremely straightforward. There are particular **graphical user interfaces** for each type of user, which will allow one to undertake proper tasks or requests. The specific GUI viewed will be contingent upon how the user is stored in the database. The manager must store the employee information in the department section as follows: a – admin, h – housekeepers, d – desk attendant.

A manager will log in and have three options of booking a room, managing employees, or viewing room service requests. The main duty of the manager is to manage all employees and since they are a manager, they will also be able to assist desk attendants in booking a room or view and inform housekeepers of pending requests. Hiring (adding) and firing (deleting) of employees can be done if the manager selects the manage employees option. The manager can also edit an employee's information in the case of their being some incorrect stored information or a change in position.

A desk attendant has to ability to book rooms by simply entering a guest's general identification and payment information. The desk attendants also have the ability check the status of a guest's room.

Housekeepers will use the application to view/update and pending requests.

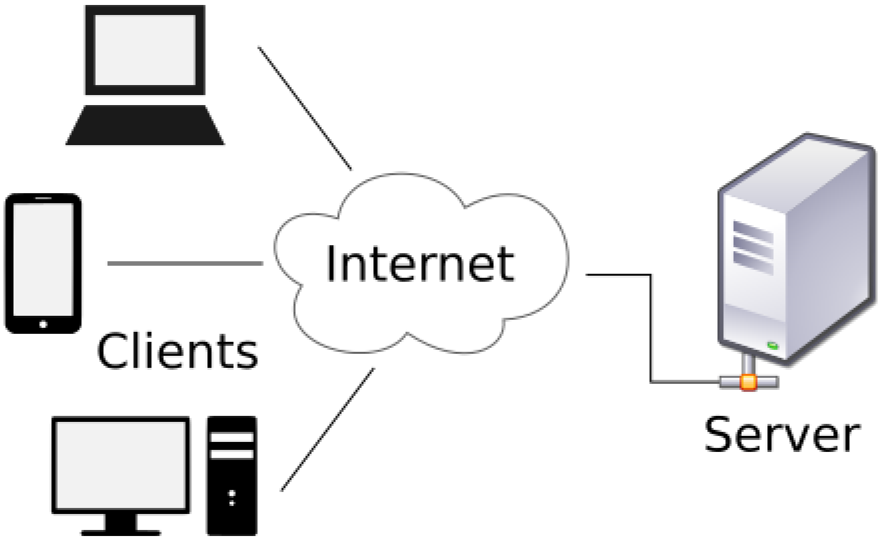
Guests will have a separate module they log in to make room service requests (services, needs, foods, etc.). They will submit their requests so that the housekeepers will be able to view them.

Database Used:

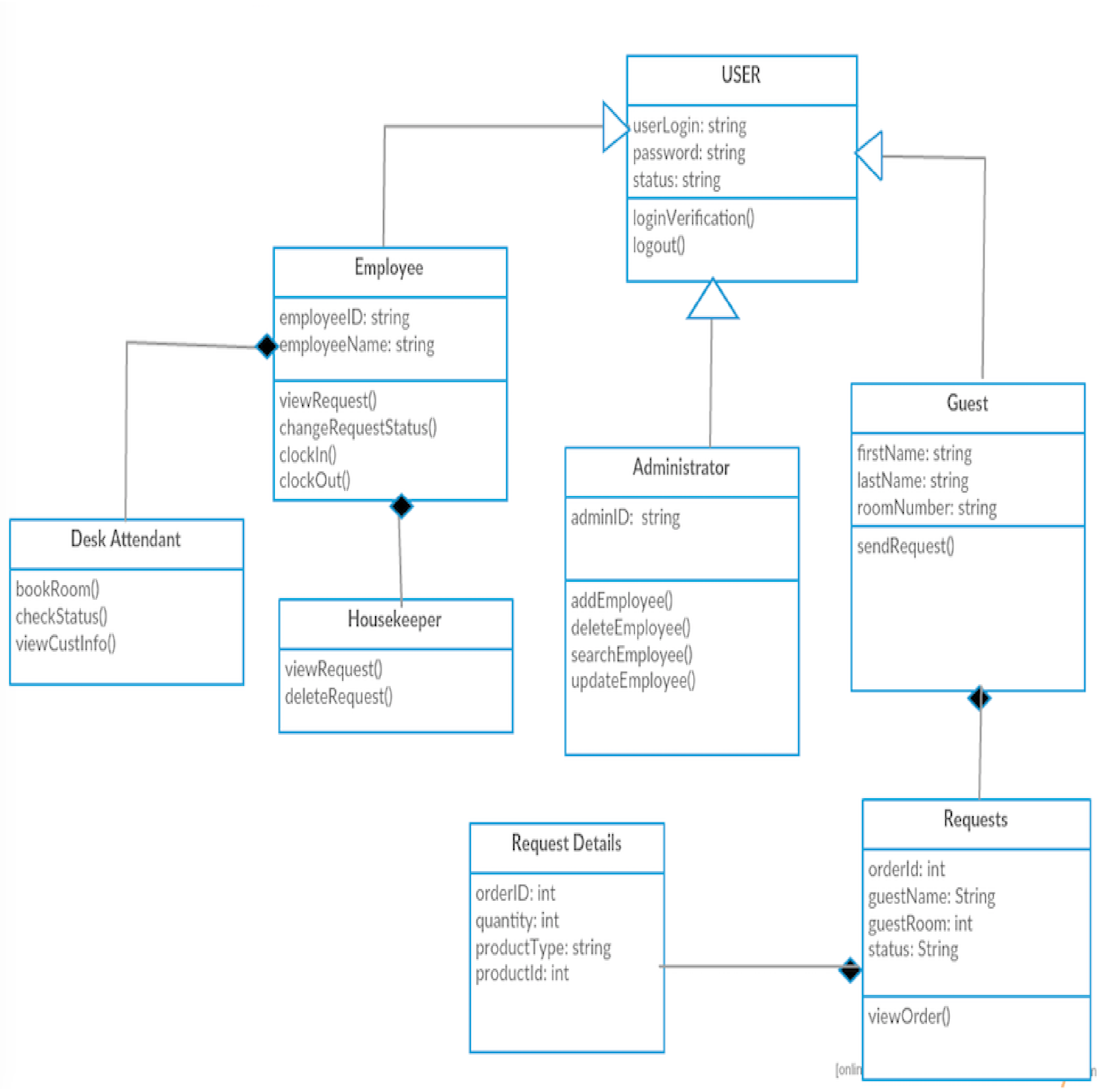
SQLite will be used to manage the database. One database will be created to maintain four separate tables used for the Alliance Hotel System. The four separate tables will be as follows: Employee/User login, Hotel Rooms, Room Service Requests, and Customer Information. A graphical user interface will be created in order for the user to interact with the database. Employees will be able to access, add and process data stored within the database. The customers will have login privileges for the room service portion of the software. They will also have access to their room and billing information. The employee will have their own personal login information, access to the hotel services, and the customer's information. Manager will have the privilege to access and manipulate all data contained within the database.

Software Architecture:

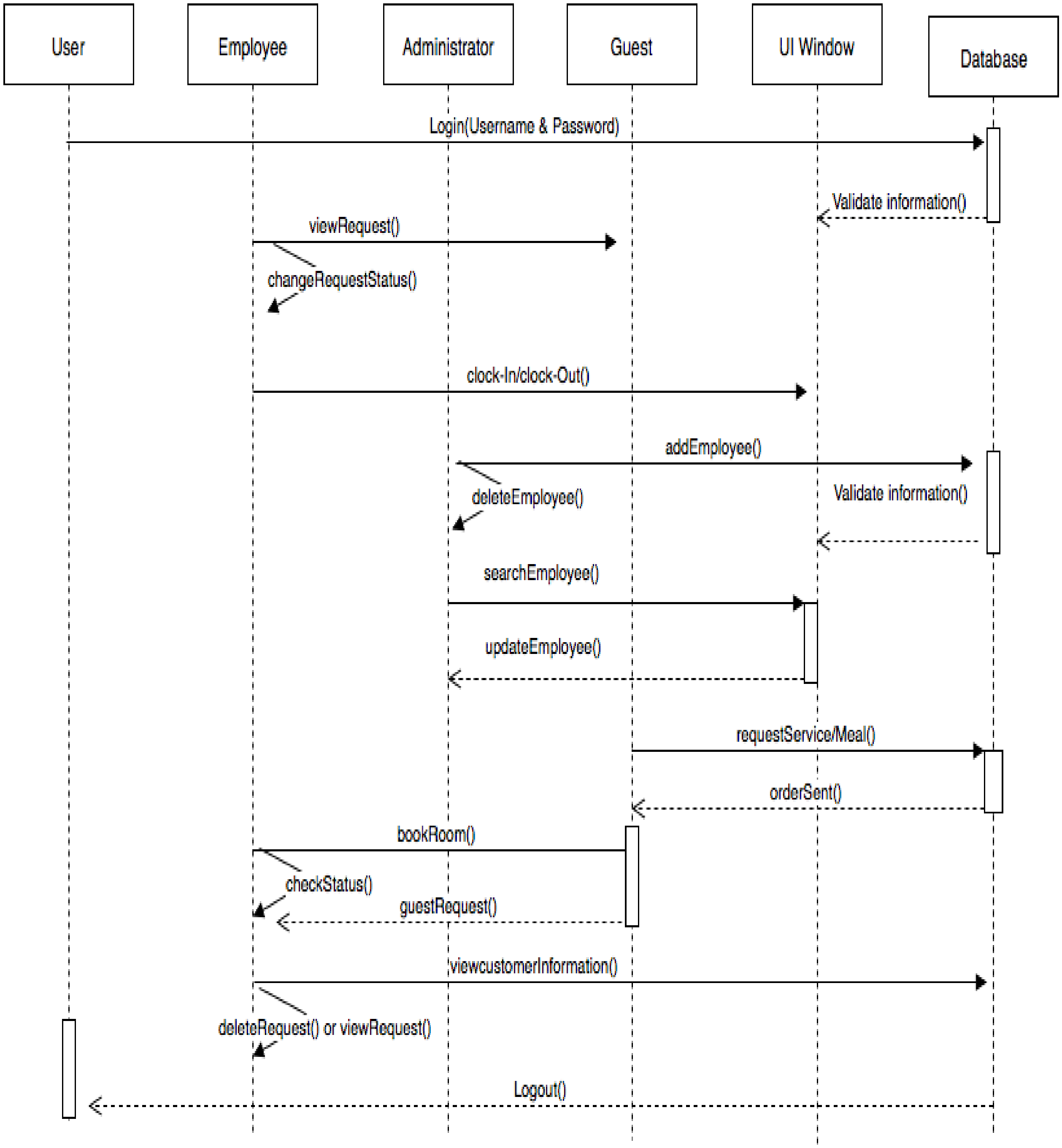
For the Basic Alliance System, our software will be an implementation of the Client-Server model as depicted below. The Employee/customer will interact with the server by sending information through the software's interface. That information will be stored in a database. The software application will be installed into the computers and kiosk that have been placed throughout the hotel.



Object Design:

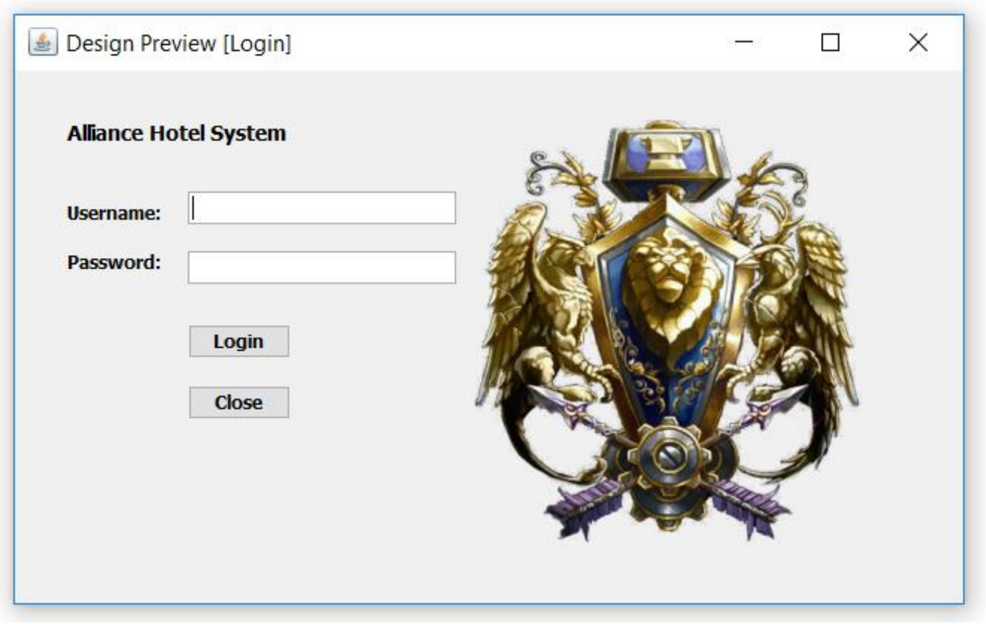


UML Category Interaction Diagram:

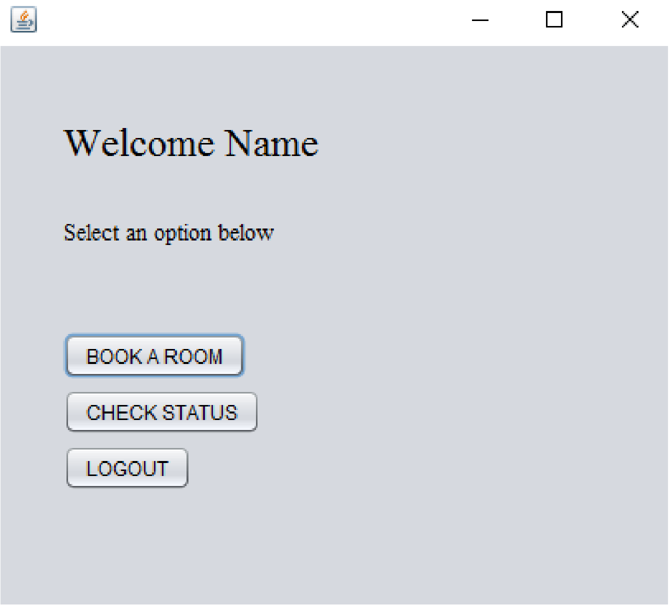


Horizontal Prototype:

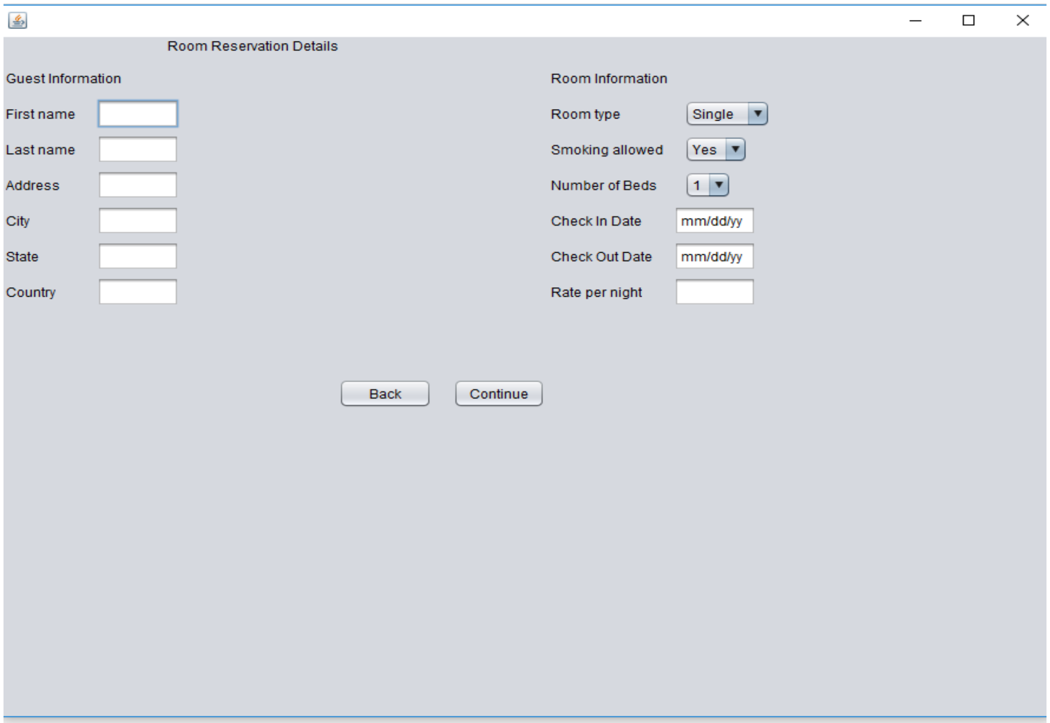
Employee Login UI



Home Screen UI



Room Reservation UI



Room type: Single

Smoking allowed: Yes

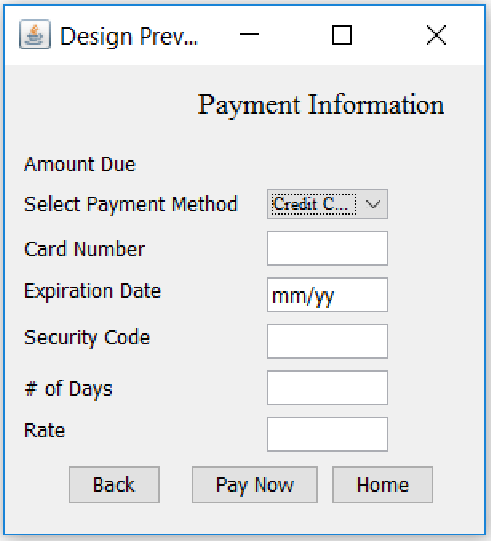
Number of Beds: 1

Check In Date: mm/dd/yy

Check Out Date: mm/dd/yy

Rate per night

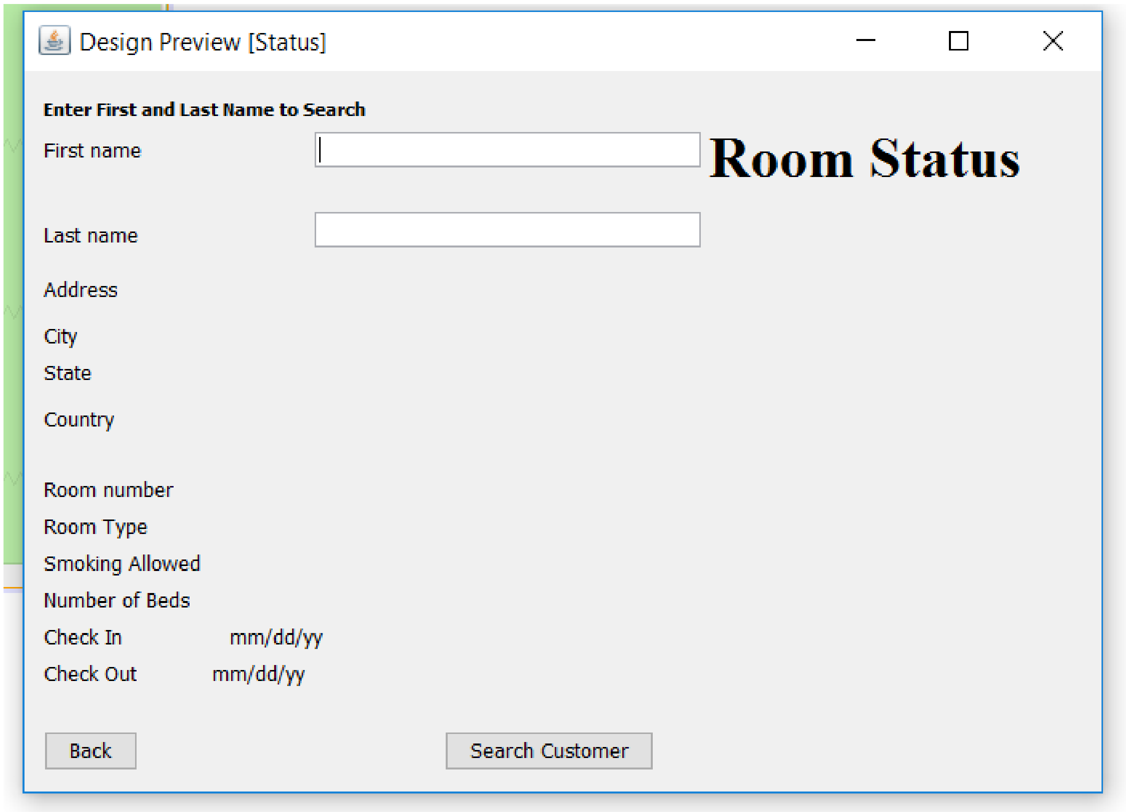
Payment UI



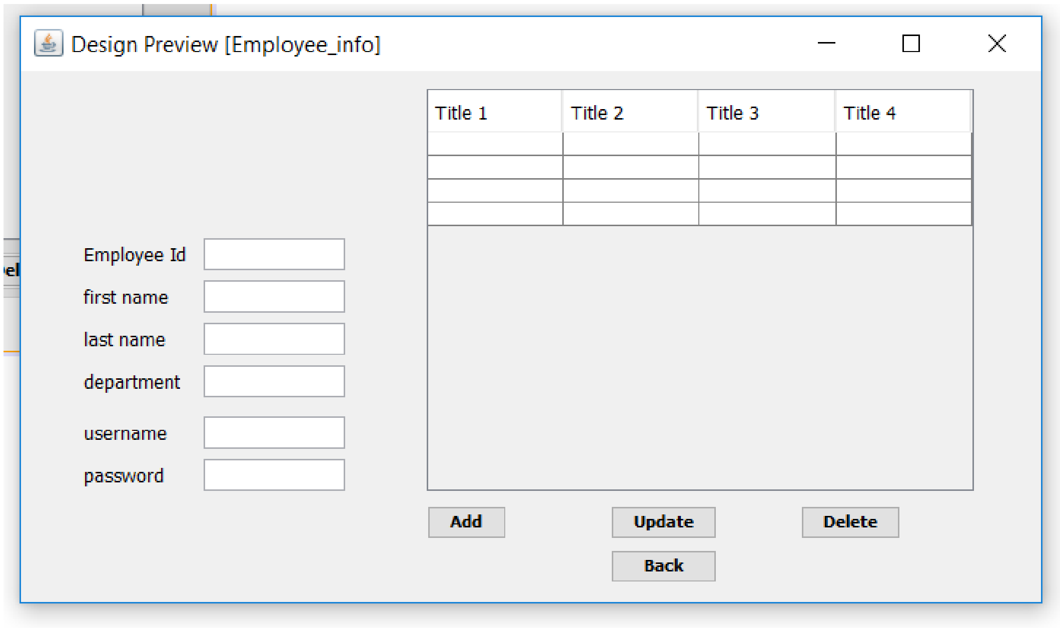
Select Payment Method: Credit C...

Expiration Date: mm/yy

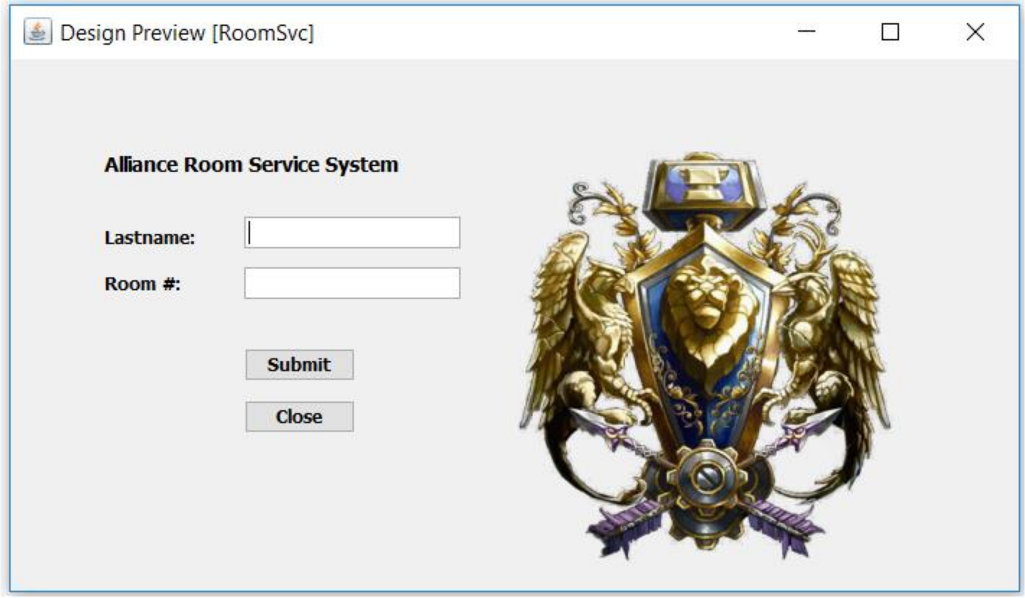
Room Status UI



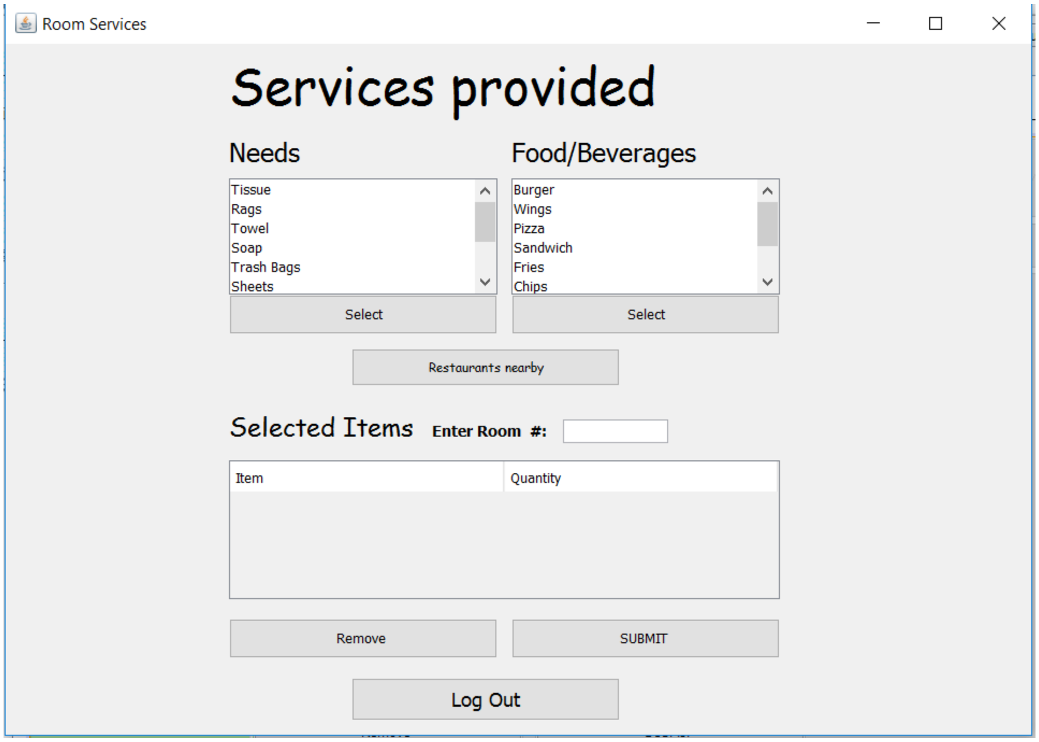
Employee Database UI



Room Service Login UI



Room Service UI



Legacy:

- Learned communication was an extremely important necessity
- Improved critical thinking when initially decided what type of system to build and how we would collaborate and make it work for the entire group
- Setting aside a time where everyone in the group could meet was challenging so we started to use Group Me to communicate more often
- Becoming a better problem solver and improving on coding skills
- Learning how to divide work between everyone equally so everyone was satisfied

Team: SunRay MGMT

Members:

- Jared Ariche
- Jerry Vu
- Heenali Patel
- Vontez Heard
- Andrew Bolaji

Topic Chosen

Hotel Management Software System

Team Coordinator

Jared Ariche

Team Roles

Heenali Patel	Documentation UML Quality Assurance (Performance)
Jared Ariche	Team Coordinator Lead Developer (Login) Database Manager Documentation
Jerry Vu	Developer (Guest Information) Documentation User Interface
Vontez Heard	Developer (Room Services) Documentation Quality Assurance (Tester)
Andrew Bolaji	Developer Documentation

Group Members Resumes:

Jared K. Ariche

141 Piedmont Ave NE, Atlanta, GA 30303 | (252) 340-3730 |
jaredk.ariche@gmail.com

OBJECTIVE: To work full-time as a software programmer related to information technology.

EDUCATION:

Georgia State University – Atlanta, GA
May 2017

Bachelor of Science, Computer Science | Cumulative GPA: 3.40

Dean's List

Spring 2015, Fall 2015, Fall 2016

Morehouse College – Atlanta, GA
August 2012 - May 2013

Computer Science | Cumulative GPA: 3.19

Relevant Coursework: Data Structures, Big Data Programming, System-Level Programs, Computer Architecture, Algorithm Design/Analysis, Software Engineering, Embedded Systems, Operating Systems

COMPUTER/TECHNICAL SKILLS:

- **Programming Languages:** Java, C++, HTML/CSS, PHP, JavaScript, SAS, MySQL,
  Visual Basic, Java Virtual Machine Language, MATLAB, UNIX, Python
- **Application Certification:** Microsoft Word 2010 and PowerPoint 2010 – Certified in 2012
- **Foreign Languages:** Spanish - Conversational

PROGRAMMING PROJECTS:

- **Software Engineering:** Created calculator GUI application using Java Swing API

- **MATLAB:** Created MP3 player GUI application using MATLAB functions and commands
- **System Level Programming:** Used bash script and C to simulate a company payroll database
- **Web Programming:** Used HTML/CSS, PHP, JavaScript to create ecommerce website that sold shoes

TECHNICAL EXPERIENCE:

Girls Who Code – Instructor
January 2016 - Current

- Taught basic computer programming concepts through Python
- Facilitated creation of Python application and online class modules with up to 10 students

Morehouse College – Atlanta, GA
September 2012 – May 2013

Computer Science Department Teacher Assistant

- Communicated with all levels of professional staff including three corporate companies
- Assisted with technology support in areas such as troubleshooting and data entry to improve work efficiency

WORK EXPERIENCE:

Top Shelf Hospitality – Manteo, NC

July 2013 – August 2016

Associate Manager

- Supervised and trained 4 employees to fulfill dessert and drink orders and provide customer service.
- Provided technical support to troubleshoot and calibrate computer software issues to improve efficiency.

RELATED PROFESSIONAL AFFILIATIONS/ORGANIZATIONS:

- **Association for Computing Machinery (ACM):** Member
  January 2016 - Current
- **Institute of Electrical and Electronics Engineers (IEEE):** Member
  February 2014 – Current
- **IEEE Computer Society:** Member
  February 2014 – Current

RELATED VOLUNTEER EXPERIENCE:

Panther Hackers – Atlanta, GA
October 2015 – Current

- ♦ Worked in a team of 4 students in HackATL 48-hour coding competition for college students.
- ♦ Developed and designed a logo and MAC OS application which created recipes based on common household ingredients

Heenali Patel

415 Emerald Trace Jonesboro, GA 30236
678-629-2250 hpatel106@student.gsu.edu

Education

Georgia State University, Atlanta, GA Expected Graduation Fall 2017

Bachelor of Science, Computer Science

Technical Skills

Programming Languages:	Java, Html/CSS
Used IDEs:	Eclipse, NetBeans
Office Applications:	Microsoft Office, Word, PowerPoint, Excel
Design Applications:	Photoshop
Operating Systems:	Windows 10, Mac OS X, UNIX, and IOS
Information technology:	Computer troubleshooting, Software installation

Related Experience

Relevant Courses Taken:

- **System-Level Programming**–An introduction to programming at the level of the operating system.
- **Principles of Computer Programming**–Fundamental principle of computer programming.
- **Theoretical Foundations of CSC**-Covers the basic theoretical foundations required studying various sub-disciplines in computer science.
- **Digital Image Processing**-Fundamentals of image processing, including image digitization, description, image transforms, filtering, restoration, coding, and retrieval.

Projects:

- Helped create and develop a hotel reservation web application with a team using an agile development cycle

Jerry Vu

3193 Henderson Walk Atlanta, GA 30340 | 678-468-1140 | jerryvu89@gmail.com

Objective

An internship or research opportunity that will allow me to utilize my problem solving skills and attention to detail to further build my professional career in the field of computer science.

Education

Georgia State University

- Major: B.S. in Computer Science, Expected May 2018 GPA: 3.12
- Minor: AS in Science
- Related coursework: Principles of computer science I and II, Data Structures, System-Level Programming, Matlab.

Skills / Strengths

- **Programming languages:** Java, C, C++, UNIX, Matlab
- **Software:** Microsoft Office and Adobe Photoshop
- **Operating Systems:** Windows (95, 98, 2000, XP, 7, and 10) and Mac OS
- **Professional:**

Project

- Matlab: Grocery Budget System (Fall 2016)
  - Created a program that displays the cheapest groceries from popular grocery stores.

Activity

GSU Hackathon, 2016

- Participated in the competition hosted by GSU Computer Science Department's ACM and IEEE student organizations.

Association for Computing Machinery 2016

- World's oldest and largest educational and scientific computing society.

Vontez Heard

3496 Pleasant Brook Village Lane, Atlanta, Georgia 30340
Cell: 770-655-2624 | vheard2@student.gsu.edu

Education

Georgia State University, Atlanta | Expected graduation – Dec. 2017
Bachelor of Arts & Sciences in Computer Science **|** *GPA: 3.1*
Apalachee High School, Winder, Georgia | graduated in 2013 with Honors while playing football and basketball year-round

Experience

Georgia State University Library, Atlanta, GA
PC Systems Assistant, 08/17/2015 – Present

- Manage, update, and troubleshoot over 500 computers and printers/scanners
- Provide instantaneous technical customer service to Library Staff through help desk
- Thoroughly research, communicate with vendors, provide comprehensive documentation, etc. when troubleshooting an issue

Skills

Technical

- Skilled in Java, HTML/CSS
- Knowledge of C, C++, Python, Unix Shell Scripting, and SQL programming languages
- Experience with SCCM, Active Directory, GPO, Munki, Ghostcast, DeepFreeze, and various other mass-management software
- Efficient in Microsoft Office Software: Access, MS Word, PowerPoint, Excel, Visual Studio, and Publisher

Accomplishments/Activities

Relative Activities/Coursework

- **Software Engineering**: developed a functional room service GUI (using Java Swing and SQLite) for a Hotel Management Software; increased understanding of UML methodology
- Team Leader of my team in Home Depot's CodeAthalon in 2015

Andrew Bolaji

3004 Fairview Road • Covington, GA 30016
404-437-4881 • andrewbolaji@gmail.com

Objective

Aiming to utilize my strong academic SQL, relational database, and Java knowledge and experience for an employer by means of a full time Opportunity. Entry level, but I am ready to learn job specifics quickly, aggressively and passionately.

Education

Georgia State University – Atlanta, GA **Graduation- May 2017**
Bachelor of Science, Computer Science

Georgia Perimeter College – Atlanta, GA **Graduation-May 2016**
Associate in Science, Computer Science

Technical Skills

- SQL, Java, Linux. Basic C# and .NET. Knowledge in Oracle, SQL server, T-Sql and Agile.
- Windows OS, Mac OS, and Microsoft Office
- Currently Training for Microsoft SQL Server 2012

Relevant Coursework

- Database Systems, Data Structures, Web Programming, Computer Programming, System Level Programming,

Significant Projects

Game Database Development Project
Fall 2016

- Collaborated as a team to plan, design, and develop a full development process project to help optimize a game and store player information efficiently
- Project and tasks included complex queries, tables, and diagrams and use of oracle 11g
- Needs and user requirements fully satisfied, met and covered
-

Hotel Management Software System Project
Spring 2017

- Full-scale System designed to allow customers to check in rooms, as well as employees to manage customer and room information.
- Connected SQLite Database with Java
- Designed advanced Graphical User Interfaces using Java Swing

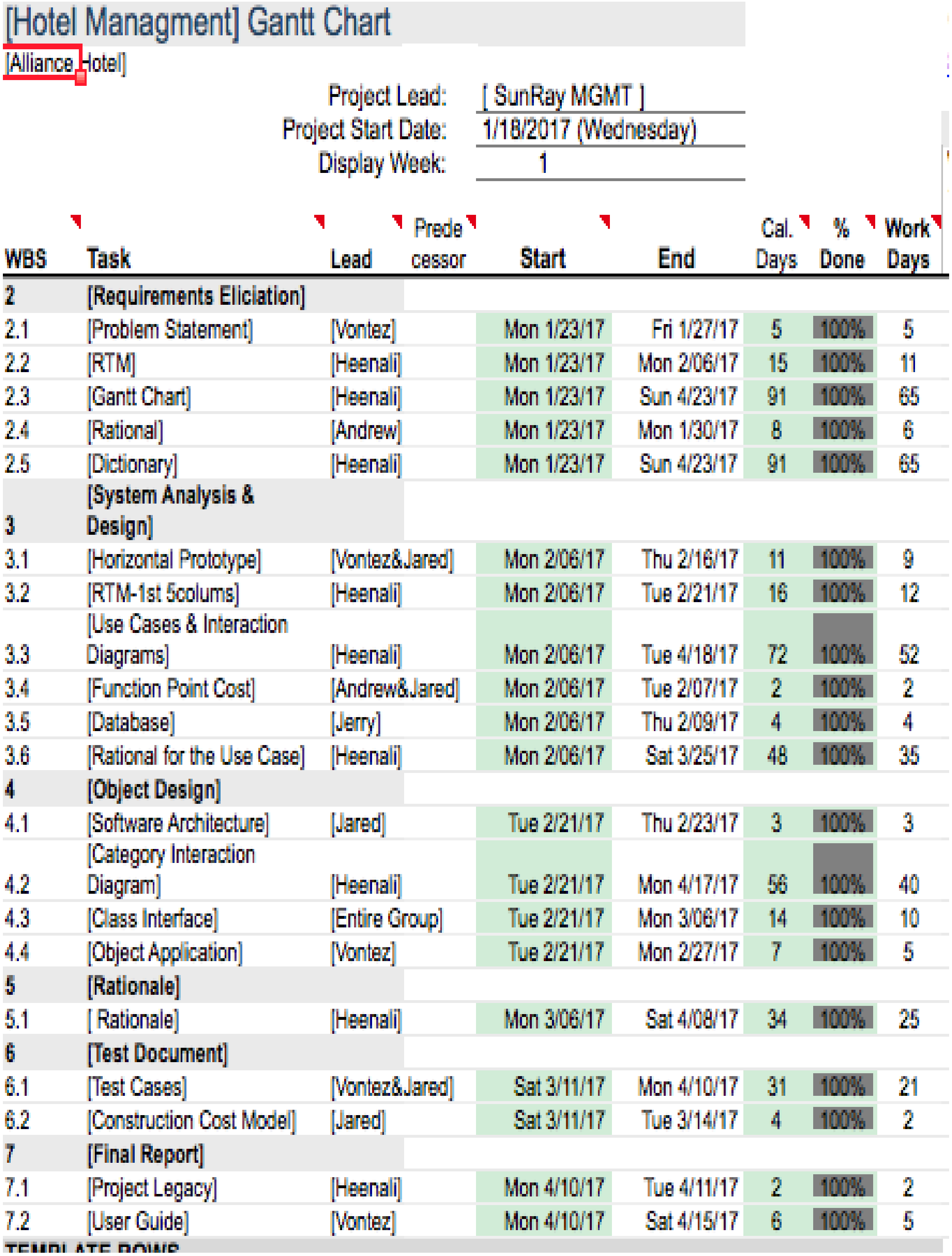
Work Experience

Brand Ambassador **October 2015- Present**

A la Carte marketing- Lawrenceville, GA

- Customized marketing plans and strategies to cater to various companies needs
- Promoted deals and specials for a national telecommunications company
- Provided marketing and sales services for businesses in Georgia, Alabama, and Tennessee

Gantt Chart:



[Hotel Managment] Gantt Chart

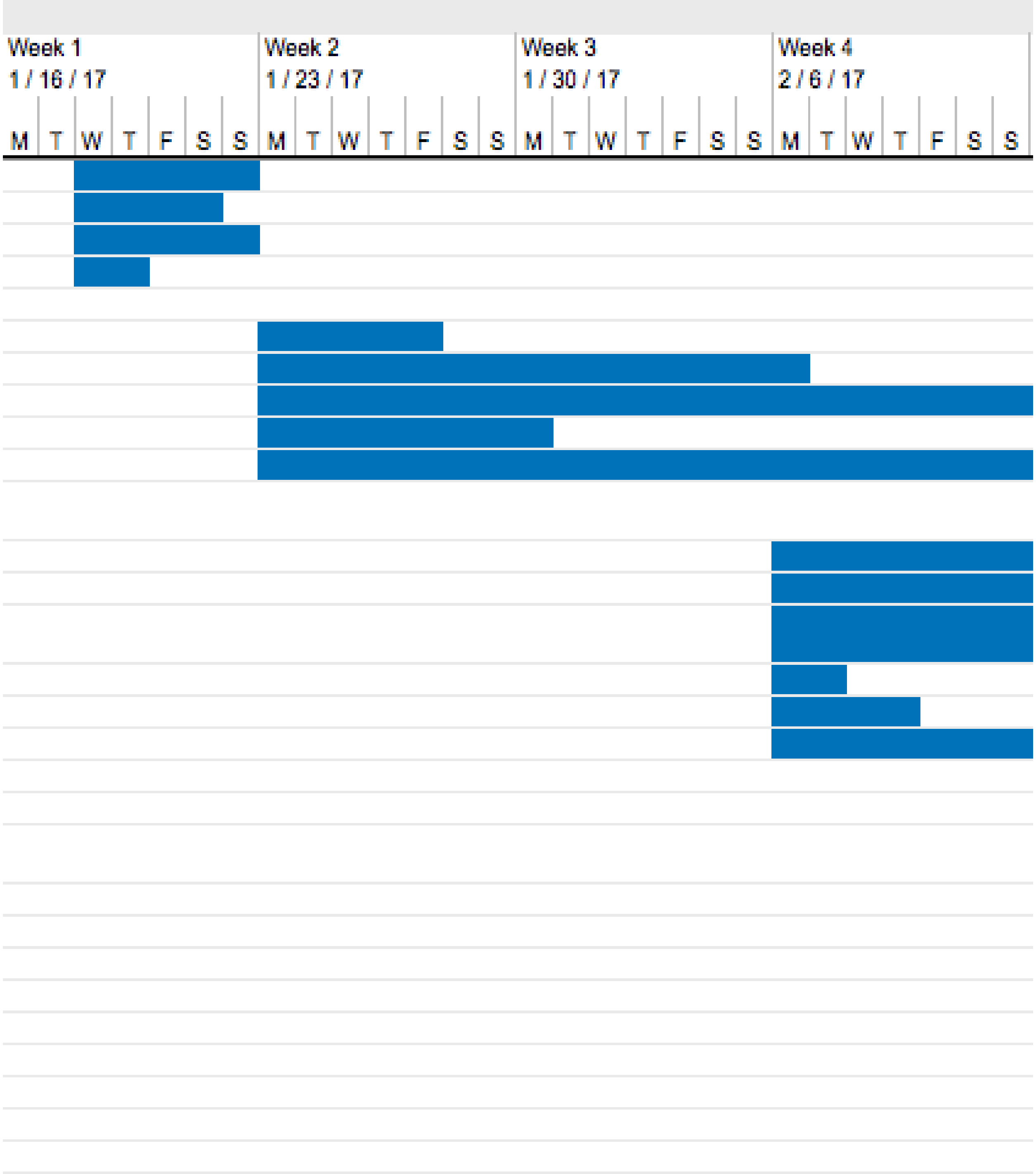
[Alliance Hotel]

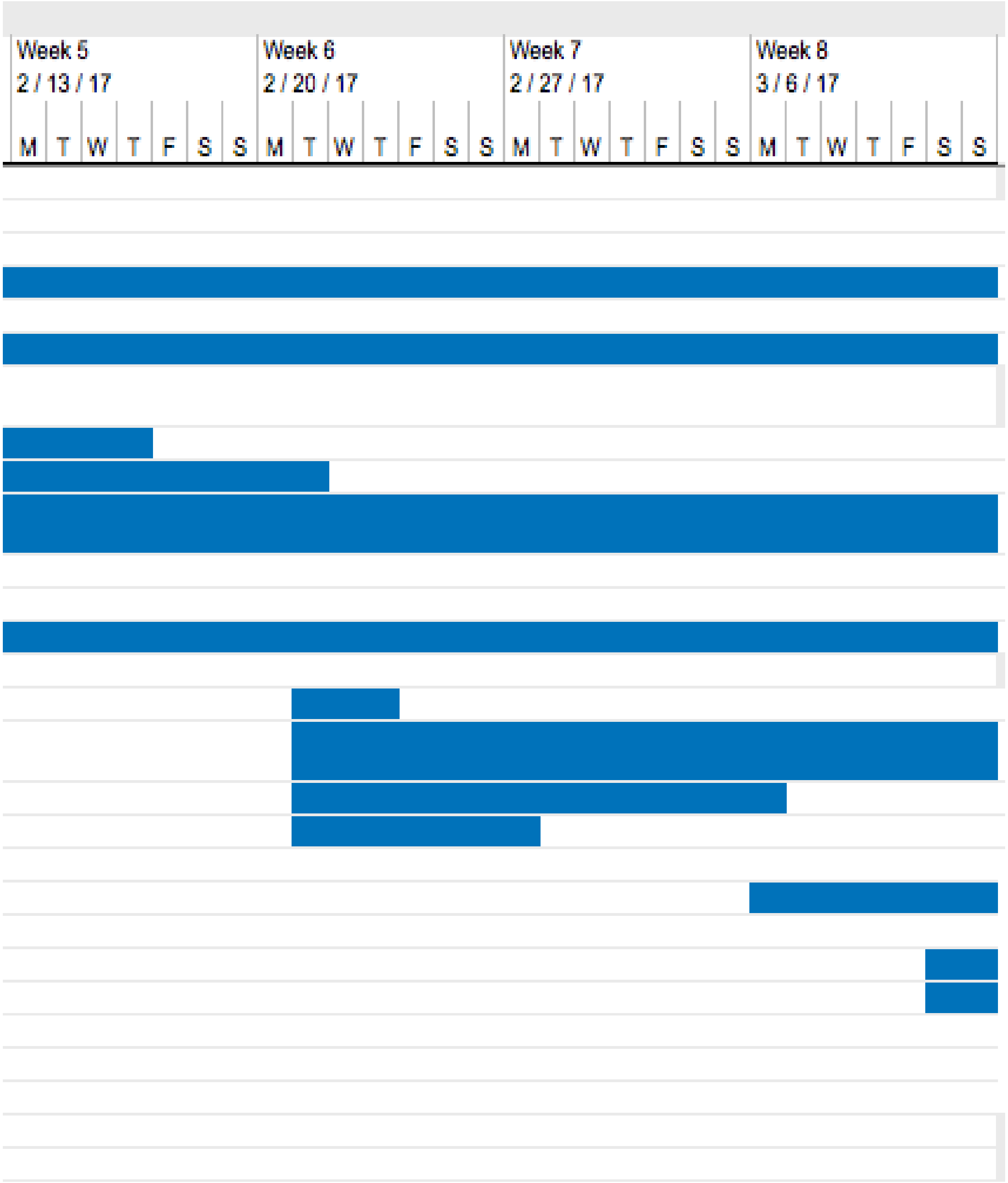
Project Lead: [SunRay MGMT]
Project Start Date: 1/18/2017 (Wednesday)
Display Week: 1

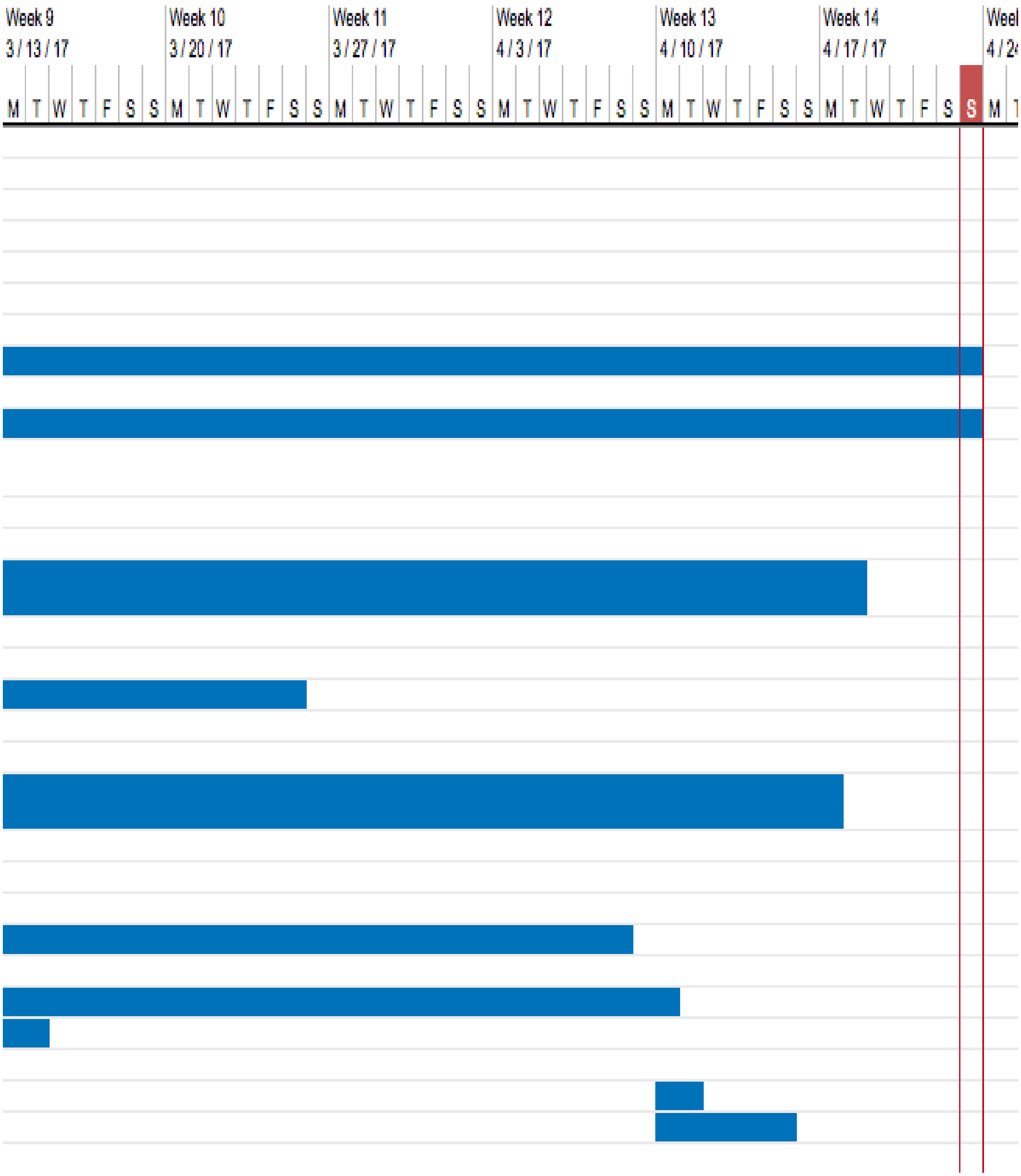
			Prede			Cal.	%	Work
WBS	Task	Lead	cessor	Start	End	Days	Done	Days
2	**[Requirements Eliciation]**							
2.1	[Problem Statement]	[Vontez]		Mon 1/23/17	Fri 1/27/17	5	100%	5
2.2	[RTM]	[Heenali]		Mon 1/23/17	Mon 2/06/17	15	100%	11
2.3	[Gantt Chart]	[Heenali]		Mon 1/23/17	Sun 4/23/17	91	100%	65
2.4	[Rational]	[Andrew]		Mon 1/23/17	Mon 1/30/17	8	100%	6
2.5	[Dictionary]	[Heenali]		Mon 1/23/17	Sun 4/23/17	91	100%	65
3	**[System Analysis & Design]**							
3.1	[Horizontal Prototype]	[Vontez&Jared]		Mon 2/06/17	Thu 2/16/17	11	100%	9
3.2	[RTM-1st 5colums]	[Heenali]		Mon 2/06/17	Tue 2/21/17	16	100%	12
3.3	[Use Cases & Interaction Diagrams]	[Heenali]		Mon 2/06/17	Tue 4/18/17	72	100%	52
3.4	[Function Point Cost]	[Andrew&Jared]		Mon 2/06/17	Tue 2/07/17	2	100%	2
3.5	[Database]	[Jerry]		Mon 2/06/17	Thu 2/09/17	4	100%	4
3.6	[Rational for the Use Case]	[Heenali]		Mon 2/06/17	Sat 3/25/17	48	100%	35
4	**[Object Design]**							
4.1	[Software Architecture]	[Jared]		Tue 2/21/17	Thu 2/23/17	3	100%	3
4.2	[Category Interaction Diagram]	[Heenali]		Tue 2/21/17	Mon 4/17/17	56	100%	40
4.3	[Class Interface]	[Entire Group]		Tue 2/21/17	Mon 3/06/17	14	100%	10
4.4	[Object Application]	[Vontez]		Tue 2/21/17	Mon 2/27/17	7	100%	5
5	**[Rationale]**							
5.1	[Rationale]	[Heenali]		Mon 3/06/17	Sat 4/08/17	34	100%	25
6	**[Test Document]**							
6.1	[Test Cases]	[Vontez&Jared]		Sat 3/11/17	Mon 4/10/17	31	100%	21
6.2	[Construction Cost Model]	[Jared]		Sat 3/11/17	Tue 3/14/17	4	100%	2
7	**[Final Report]**							
7.1	[Project Legacy]	[Heenali]		Mon 4/10/17	Tue 4/11/17	2	100%	2
7.2	[User Guide]	[Vontez]		Mon 4/10/17	Sat 4/15/17	6	100%	5
TEMPLATE ROWS								

Gantt Chart Template © 2016 by Vertex42.com.

See info on Gantt Chart Template Pro




Week 5
2 / 13 / 17
Week 6
2 / 20 / 17
Week 7
2 / 27 / 17
Week 8
3 / 6 / 17
M T W T F S S M T W T F S S M T W T F S S M T W T F S S


Week 9
3 / 13 / 17
Week 10
3 / 20 / 17
Week 11
3 / 27 / 17
Week 12
4 / 3 / 17
Week 13
4 / 10 / 17
Week 14
4 / 17 / 17
Weel
4 / 2
M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T

Glossary:

Basic Alliance System - bridge the gap between room service, hotel booking, management and the consumer in order to improve the check-in process as well as room service.

Check-In - Anytime after 3:00 P.M

Checkout - Before 11:00 A.M

Desk Attendant - Employees have access to view, book, modify and delete reservations.

Employee Login UI - Employee’s login page.

Graphical user interface - will be created in order for the user to interact with the database

Housekeeping - Employees have the access to view, modify, and delete room service requests from each guest staying at the hotel

Managers - Employees have the access to view all databases that are incorporated into the software.

SQLite - a relational database management system contained in a C programming library