HOTEL MANAGEMENT SYSTEM

System Requirements Documentation

Kristen Thomas

INFO-C451

Contents

Customer Problem Statement

Hotel management systems are utilized by most hotels around the world to handle the large volume of guests, rooms, and other information they have to handle securely. Since hotels are the most utilized accommodations when traveling, it is important that they can accurately store all of the guests personal and booking information. And as a lot of guests will check in immediately at their designated check in time, it is important to have a way to streamline this process. Creating a kiosk or app where guests can access their information and check in or check out will allow for an expedited process when staying or leaving a hotel.

System Requirements

- Objectives: The objectives of the system are to allow for express check in and check out, secure storage of guest and booking information, and payment processing. Managers will also be able to utilize this system to help guests with their information, but this system will ultimately free up their time to attend to other matters. For front desk workers, it will decrease the wait times of guests checking in and can provide help to those who need it.
- **System Requirements:** The system needs to be able to access guest information quickly, be secure when storing information, and be scalable to larger hotel chains.
- Typical Customers: Any travelers and hotel guests can use this system. Avid travelers will definitely make good use of this system to expedite the check-in process. People who don't like to wait in lines or don't want to interact with the front desk staff will also benefit from this system.
- Project planning:
    - Software: Notepad ++, XAMPP, Apache, MySQL
        - Languages: HTML, JavaScript, PHP, CSS
    - Hardware: PC
    - Network Requirement: high-speed internet connection with cellular backup, cloud storage
- Development approach: I will most likely being using Java or JavaScript, and some SQL server to store the information.

Functional Requirements Specification

| No.	Priority Weight	Description
REQ-1	High	Should be able to book guests in all 2,000 rooms.
REQ-2	High	Different room types: 1 king bed, 2 queen beds, accessible rooms, deluxe suites
REQ-3	High	Guests must be at least 21 to check in to the hotel
REQ-4	High	Guests can't check in if hotel is full **or** don't have proper identification **or** payment is insufficient
REQ-5	High	Hotel is cash/paper free, must use debit/credit cards to pay incidental
REQ-6	Medium	Customers can use a room key or the hotel app to access room
REQ-7	Medium	Online receipts must be visible to guests to view or email
REQ-8	Medium	Guests can go back and search up their own room information
REQ-9	Low	Display hotel information
REQ-10	High	Collect incidental charge
REQ-11	High	Backend information for hotel staff including how many cards the kiosk contains and system statuses
REQ-12	Medium	Room must be calculated at nightly rate, if not already paid

System Sequence Diagram

- Secure Payment Authorization
  - o Actor: Customer
  - o Objects: Credit Card, Payment
    1. Customer inserts credit card
    2. Payment system receives credit card information
    3. Card is ejected and payment is processed
       - If approved, payment is placed on card and the customer is notified
       - If denied, error message is sent
- Management retrieving guest and booking information
  - o Actor: Management
  - o Object: Booking system/database, guest information
    1. Management logs in with manager ID
    2. Manager will enter minimum required information
    3. If the information is in the database, manager can view and edit info
    4. If the information is not in the database, error message is returned

Management retrieving guest
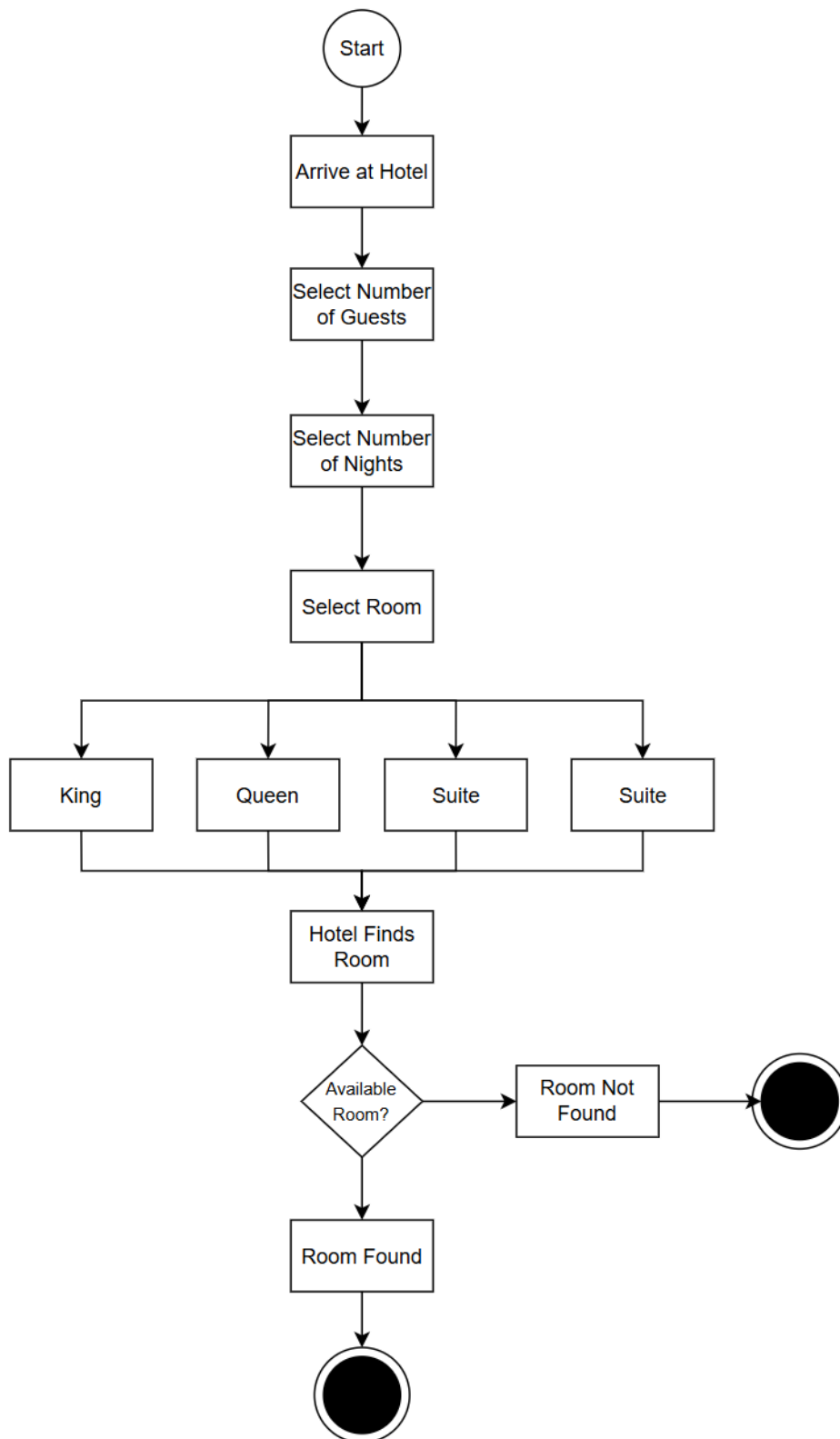and booking information
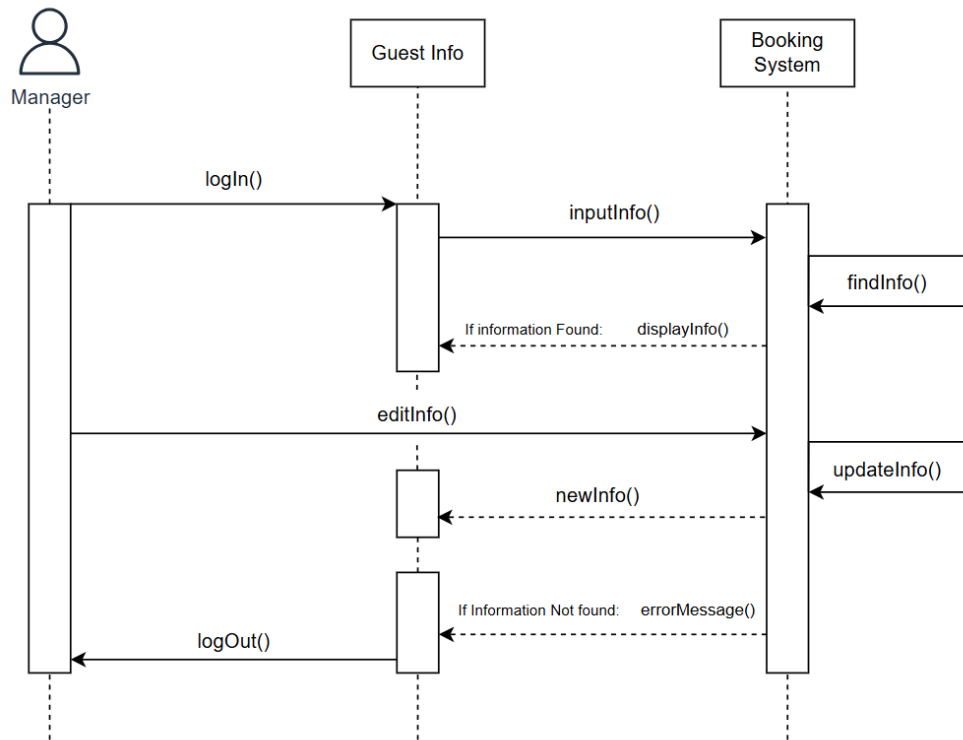
Payment Authorization
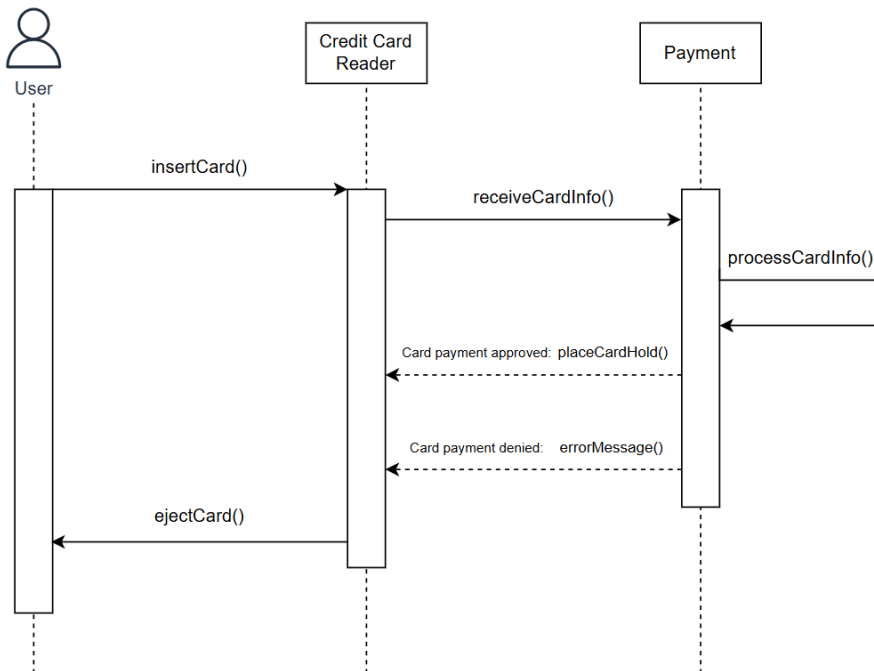
Activity Diagram

Guest books room

- States
    - Initial State: Customer arrives at the hotel
    - Final State:
        1. Customer successfully books room
        2. There are no more rooms available to book
- Actions
    - The customer arrives at the hotel. They select room type, party size, and total night stay. They are assigned a room based off of their input.

Guest uses self-check in
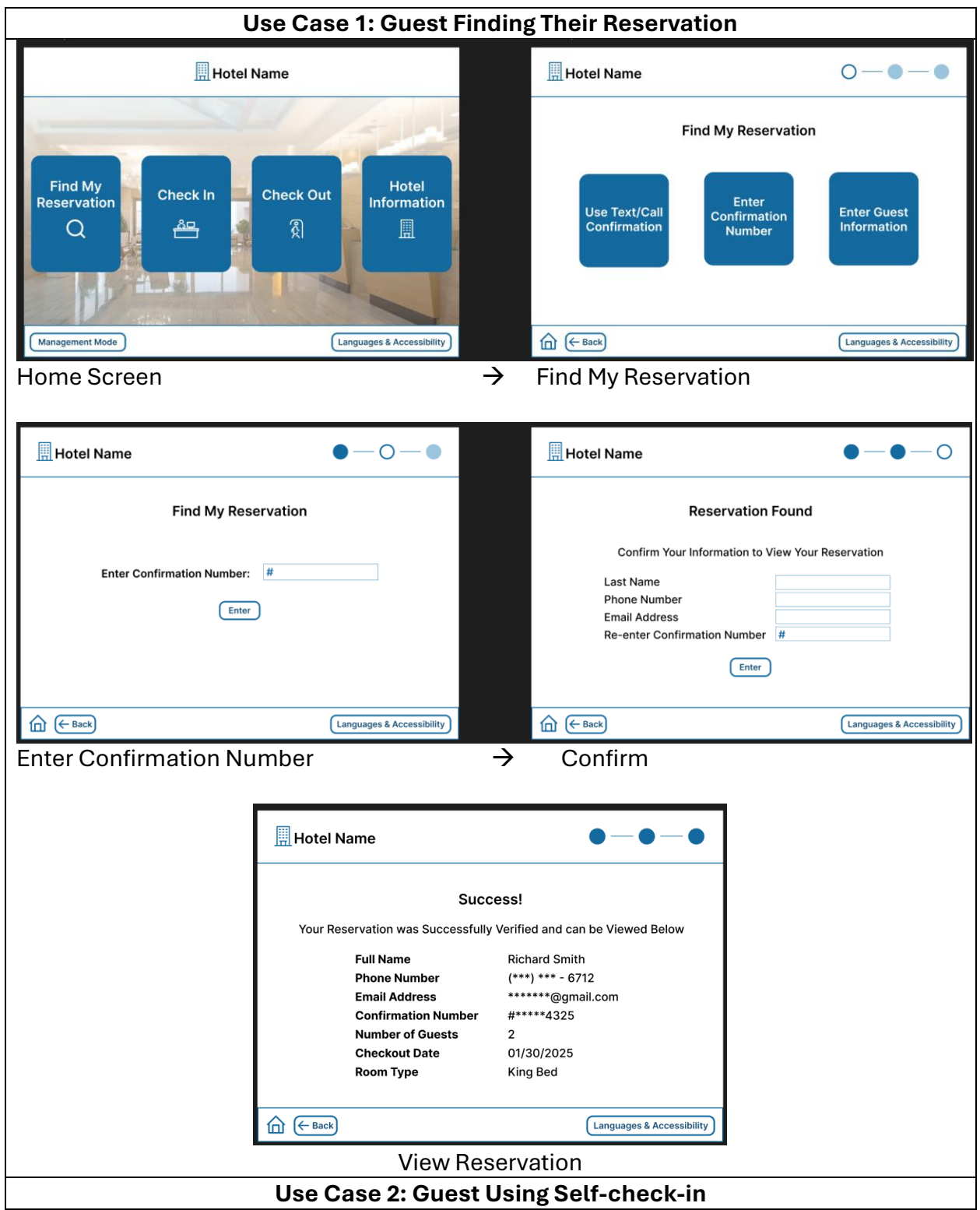
- States
    - Initial State: Guest arrives at the hotel
    - Final State:
        1. Guest successfully checked in
        2. Guest does not check in
- Actions
    - Guest arrives at the hotel. Guest shows their valid ID and enters their credit card information to pay the incidental for the room. Guest receives room number and room key through the app. Guest receives e-receipt

Guest Books Room

```
                          Start

                    Arrive at Hotel

                    Select Number
                      of Guests

                    Select Number
                      of Nights

                     Select Room

        King      Queen      Suite      Suite

                    Hotel Finds
                       Room

                   Available          Room Not          ●
                    Room?      →        Found      →

                   Room Found

                       ●
```

Guest Uses Self-Check-In

```
                    ┌─────────┐
                    │  Start  │
                    └─────────┘
                         │
                         ▼
                 ┌────────────────┐
                 │ Arrive at Hotel│
                 └────────────────┘
                         │
                         ▼
                 ┌────────────────┐
                 │  Show Valid ID │
                 └────────────────┘
                         │
                         ▼
                   ◇ Valid ID? ◇ ──No──► ┌──────────────┐
                         │               │ Guest cannot │
                        Yes              │   check in   │
                         │               └──────────────┘
                         ▼
                 ┌────────────────┐
                 │  Valid Credit  │
                 │      Card      │
                 └────────────────┘
                         │
                         ▼
                   ◇ Valid CC? ◇ ──No──► ┌──────────────┐
                         │               │ Guest cannot │
                        Yes              │   check in   │
                         │               └──────────────┘
                         ▼
                 ┌────────────────┐
                 │ Pay Incidental │
                 └────────────────┘
                         │
                         ▼
                   ◇  Paid?  ◇ ──No──► ┌──────────────┐
                         │             │ Guest cannot │
                        Yes            │   check in   │
                         │             └──────────────┘
                         ▼
                 ┌────────────────┐
                 │   Get Room     │
                 │   Number &     │
                 │   Room Key     │
                 └────────────────┘
                         │
                         ▼
                 ┌────────────────┐
                 │  Check Into    │
                 │     Room       │
                 └────────────────┘
                         │
                         ▼
                       ● End ●
```
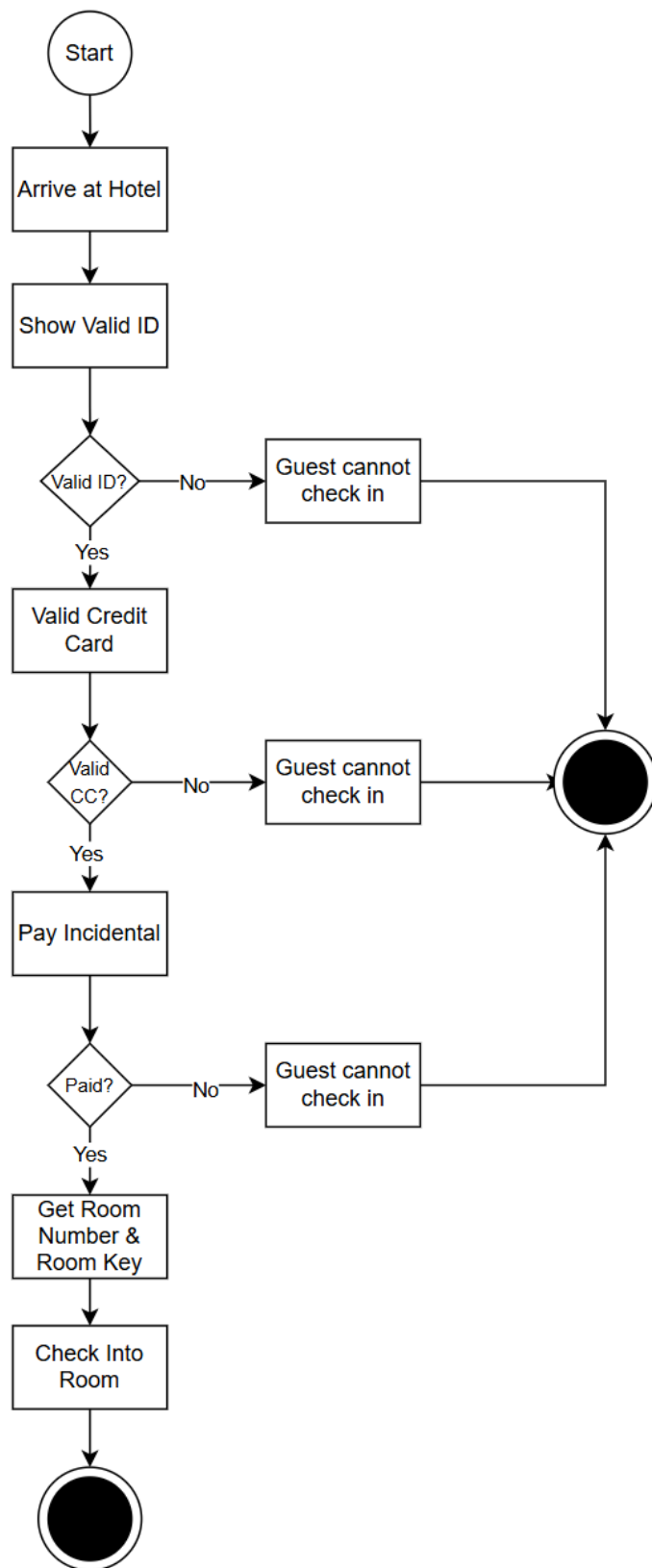
User Interface Specification

Use Case 1: Guest Finding Their Reservation

Home Screen → Find My Reservation

Enter Confirmation Number → Confirm

View Reservation

Use Case 2: Guest Using Self-check-in

Hotel Name

Check In

Start the Check-in process by entering your Confirmation Number or Name on the account.

Enter Confirmation Number: [#]

OR

Enter Full Name: [ ]

[Enter]

🏠 [← Back]   [Languages & Accessibility]

Hotel Name

Check In

Next, Scan or Enter your Driver's License or Accepted ID number below.

Enter ID Number: [#]

[Enter]

🏠 [← Back]   [Languages & Accessibility]

Enter Info Screen   →   ID information

Hotel Name

Check In

Hi Mr. Smith, Thanks for Staying with Us! Please Verify and Confirm the Following Information is Correct for Your Stay.

Full Name	Richard Smith
Phone Number	(***) *** - 6712
Email Address	*******@gmail.com
Confirmation Number	#*****4325
Number of Guests	2
Checkout Date	01/30/2025
Room Type	King Bed

[Confirm]

🏠 [← Back]   [Languages & Accessibility]

Hotel Name

Check In - You're All Set!

You are successfully checked in and assigned to
King Bed Room **2337**

The room key will be dispensed below, but you can also you use the app to enter your room and check out.

Have a Nice Day!

🏠 [← Back]   [Languages & Accessibility]

Verify/Confirm Info   →   Successful Check In

Use Case 3: Guest Pays Incidental

Hotel Name

Incidental Fee

Today, we will collect an incidental fee of **$75** for the duration of your stay. The full amount will go back on your card after you check out, should there be no damages or purchases. For any questions, please contact the main desk.

Scan or Enter Your Credit Card Info Below:

CC/Debit Card Number: [ ]

Expiry Date: [/]

Code: [ ]

[Enter]

🏠 [← Back]   [Languages & Accessibility]

Hotel Name

Incidental Fee

Processing Payment

🔄

🏠 [← Back]   [Languages & Accessibility]

Enter Information   →   Payment Processing

Hotel Name

Incidental Fee- Payment Successful!

Your Payment was Successfully posted and you may now complete the check-in process.

🏠 ← Back | Languages & Accessibility

Payment Successful

Use Case 4: Express Checkout

Hotel Name

Express Checkout

Please Enter Your Room Number and Last Name to start the check out process.

Enter Room Number: [   ]

Enter Last Name: [   ]

Enter

🏠 ← Back | Languages & Accessibility

Hotel Name

Success!

You are successfully checked out. You may keep your keys or drop them in the box below.

Thank you for staying with us!

🏠 ← Back | Languages & Accessibility

Enter information → Successful checkout

Use Case 5: Management Views Guest Information

Hotel Name

Management

Enter your Manager username and PIN to proceed.

Username: [   ]

PIN: [   ]

Enter

🏠 ← Back | Languages & Accessibility

Hotel Name

Management

Create/Edit a Reservation | Kiosk Management | Search by Guest Information | Search by Confirmation Number

🏠 ← Back | Languages & Accessibility

Management log in → Management interface

Hotel Name

Guest Search

Please enter any available information to search for a Guest.

Last Name
Phone Number
Email Address
Phone Number

Enter

⌂ ← Back   Languages & Accessibility

Hotel Name

Guest Found

1 Guest was found from the search results.

Full Name	Richard Smith
Phone Number	(***) *** – 6712
Email Address	*******@gmail.com
Confirmation Number	#*****4325
Number of Guests	2
Checkout Date	01/30/2025
Room Type	King Bed

⌂ ← Back   Languages & Accessibility

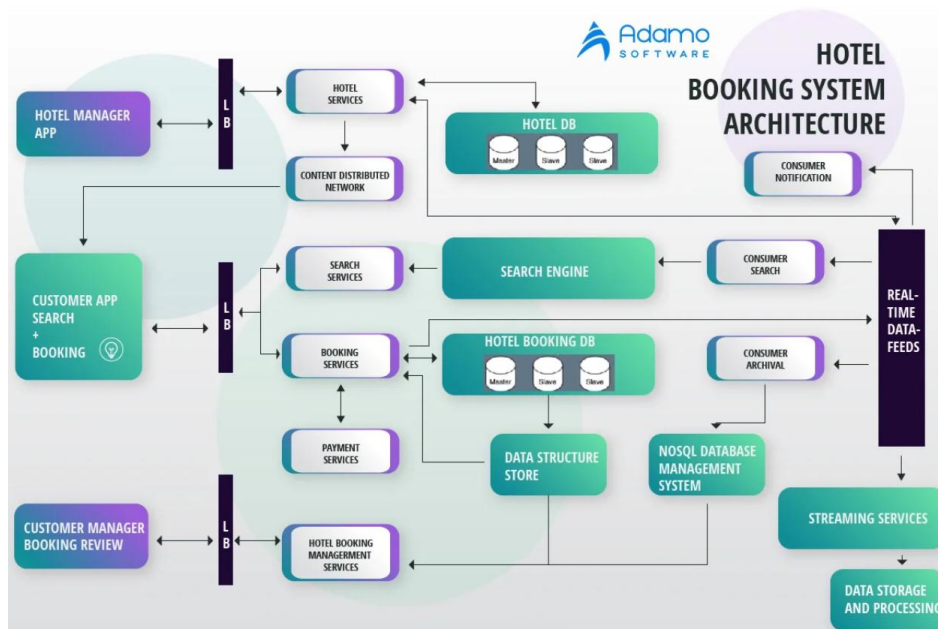Search by guest info   →   Guest Search Results

Implementation Planning

- Implementation Strategy
  - **Implementation**- A direct cutover approach will be used since there will be no current system to replace, in terms of in-lobby kiosk system.
  - **User training**- The entire office will have one or two formal training sessions and then have access to user training manuals and materials for regular access.
  - **User support**- A vendor support representative will be available for questions and troubleshooting for six months after system deployment
  - **Implementation timeline**-

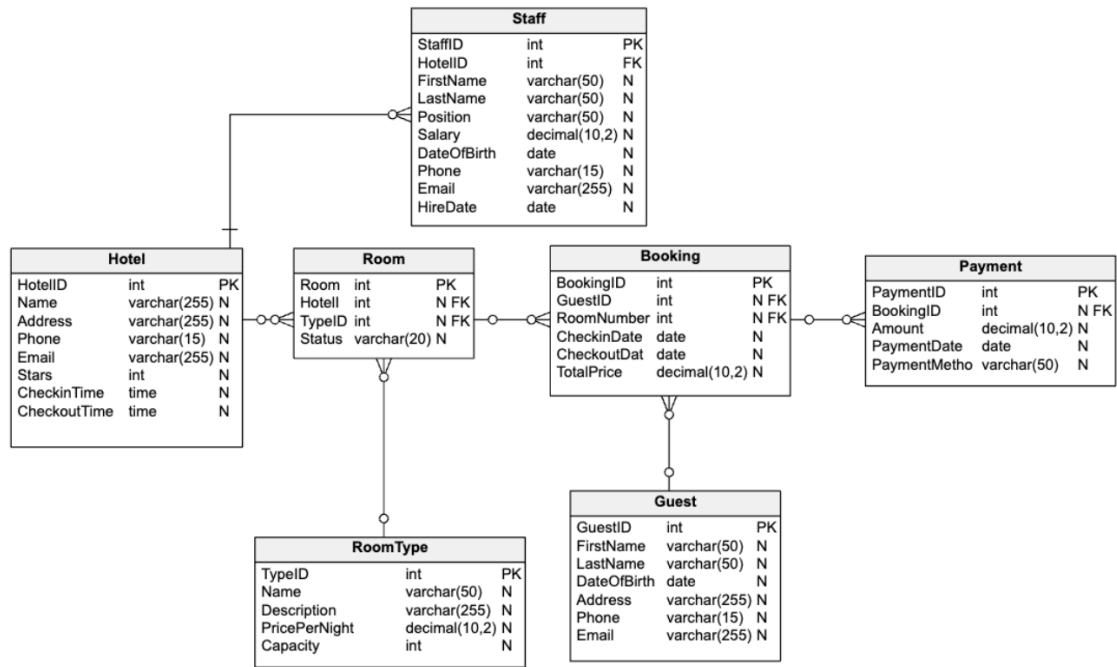| Week	Task
Week 1	Final system configuration and data migration
Week 2	Final testing
Week 3	Employee Training
Week 4	Cutover- Start new kiosk system
Week 5	System monitoring and error handling
Week 6	Post-implementation evaluation and final report

System Architecture and System Design

- A similar architecture will be constructed such as the image above: https://adamosoft.com/blog/travel-software-development/online-hotel-reservation-system/
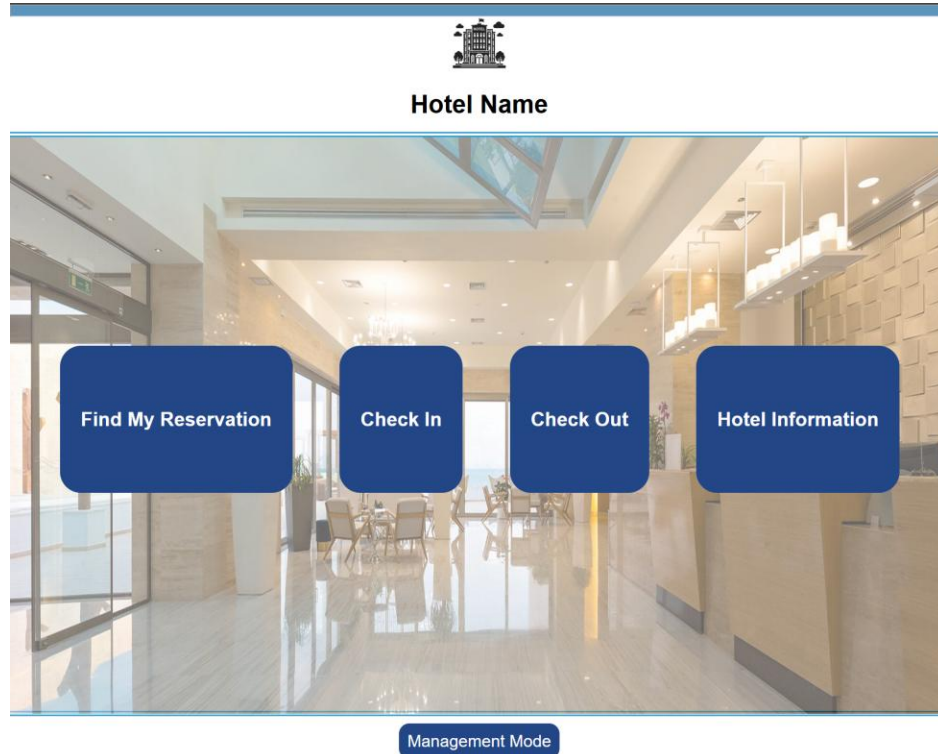
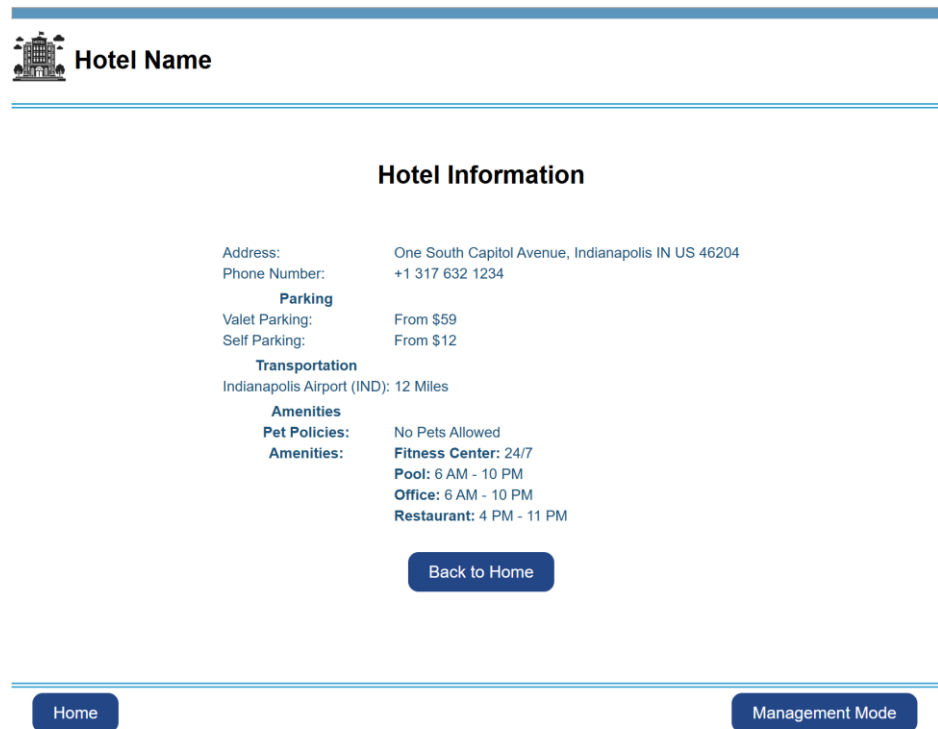## Algorithms and Data Model Implementation

- **ERD-**

Reference: https://vertabelo.com/blog/data-model-for-hotel-management-system/

User Interface Design and Implementation

- **Home Screen**

  ○
- **Hotel Info**


  ○

Design of Tests

- **Testing Plan:**
    - o **Unit testing:** To test the units, I will run each PHP file individually to ensure every unit is operating as intended. Isolating each class to check for syntax, logic and other errors to make sure they produce their intended outputs will start at the beginning and continue throughout the entire process of coding the system.
    - o **Integration testing:** I will begin to test the homepage unit, check in unit, and find reservation units first to see if they are properly interacting with each other. This will continue throughout the rest of the coding process as each file is developed.
    - o **System testing:** After the entire hotel management system is complete, I will use test cases to ensure that all of the modules are working correctly all together. The use cases will represent real life situations and tasks that will need to be completed. I will also ensure that MySQL is working correctly in accordance with the code,
- **Two test Cases:**
    - o **Test case 1:** Add a new booking reservation
        - ▪ Management will log in with their manager PIN which will take them to the manager dashboard
        - ▪ Click "Create/Edit Reservation"
        - ▪ Add all of the booking details for a new guest
        - ▪ Submit the reservation and update the database
        - ▪ Go back to the dashboard
        - ▪ Search up the new guest in "Search for bookings"
        - ▪ Look up one of the guest's new details
    - o **Case 1 expected outcome:** The system should add the new booking to the system and then populate the booking details from the database on the kiosk interface as prompted.
    - o **Test case 2:** Check a guest in and out of the system
        - ▪ Go to the check in screen
        - ▪ Enter a guest's details
        - ▪ Go to find a reservation to ensure they are checked in
        - ▪ Go to express check out
        - ▪ Go back to find a reservation to ensure they are checked out
    - o **Case 2 expected outcome:** The system should be able to successfully update the guest's status to checked in and checked out, respectively.

Error Handling and Debugging

- **Error Logging:** Log files with date and time of error, the error message, context/how it occurred, manager ID and/or guest confirmation number, and any other user notes.
- **Handling exceptions**: SQL try-catch using PHP, using user validation techniques, and user feedback
- **Debugging tools:** Inspect Elements using the browser, using queries to debug

Collaboration and Code Integration

- **Version Control:** Use GitHub to track and manage all of the versions of the system's code.
- **Team collaboration:** multiple users can develop code in GitHub, developers can commit their changes with comments.
- **Code reviews:** Other users can review their team's code for basic desk checking as well as ensuring developers are following proper coding standards.
- **Git branching workflow:**

- Reference: https://www.pablogonzalez.io/salesforce-git-branching-strategies/
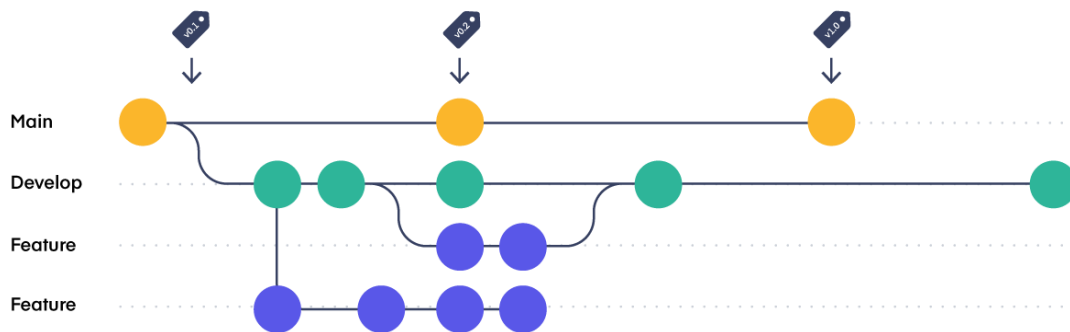
Performance Optimization Plan

- **Potential performance bottlenecks:**
    - Network delays slowing down the system
    - Large datasets/Slow Queries
    - Poorly optimized code
- **Optimization strategies:**
    - Reduce requests to the server
    - SQL pagination- smaller and more manageable queries and results
    - Indexing- better organization and structuring of data
    - Code refactoring- make more reusable code and improve structure of code

Maintenance and Support Strategy

- Post-deployment support:
  - **Types of maintenance:**
    - **Corrective:** fixing bugs in code and improving configuration settings
    - **Adaptive:** add online and cloud computing capabilities
    - **Perfective:** replace outdated hardware, upgrade networks, and improve UI
  - **Change management and version control:**
    - **Change management:** Obtain approval from team leads and supervisors for big changes to code, implementing proper training for employees, and using proper planning.
    - **Version control:** Git/GitHub with proper documentation of versions of the system.
  - **Documentation and user feedback mechanisms:**
    - **Documentation:** Create program documentation for the logic of each unit and system documentation for the entire system's program. User documentation will include videos, user manuals, and help screens that pop up when an error occurs.
    - **User feedback:** Ticket system so staff and guests can report bugs and errors with the system.

References

- http://lms.southeasterntech.edu/CIST/CIST2921/Systems%20Analysis%20and%20Design,%2011th%20Edition.pdf
- https://www.pablogonzalez.io/salesforce-git-branching-strategies/
- https://adamosoft.com/blog/travel-software-development/online-hotel-reservation-system/
- https://vertabelo.com/blog/data-model-for-hotel-management-system/