

SBI Management system

Kevin Patel

Jeff Jordan

Abstract

SBI Management System, named after the flagship Stone Brook Inn, is a program for use in the commercial Hospitality business in order to easily navigate the processes of booking, reserving, and maintaining rooms in an effective way to alleviate stresses of overbooking and unavailable rooms within businesses.

1. Introduction

The purpose of the SBI Management system is to remove the clerical errors of record keeping by hand and to cut down on the potential mistakes that could happen with note taking with rotating cast of managerial staff and employees. It is designed around the idea of condensing and centralizing all reservation and booking information into one simple, easy-to-use application in order to remove the stress of record keeping and room tracking.

On the backend, it will keep track of non-confidential customer information. The purpose of this is to implement features such as a customer rewards program, discounts, personalized service, etc. In the future, it will implement encryption for payment information.

1.1. Background

A member of our group's family owns a hotel. This project is being designed with the needs of an actual hotel business in mind. The design will be centered around a simple GUI to quickly assign rooms and check availability.

While the primary purpose of the project is for the Stone Brook Inn, we will design it to be implemented by any local small hotel or motel. The idea is to keep the functionality universal while making design edits simple and easy to understand.

1.2. Challenges

Integrating legal and secure encryption for web payments will be the ultimate challenge for the project. Ensuring that the program doesn't allow for any booking of unavailable rooms will be a challenge since there will be a lot of test cases that might go unnoticed. Designing a universal program that could be adjusted to fit any business model might present itself to be more challenging than we can currently foresee.

2. Scope

The scope of this program will be extensive. It will handle guest checkins and checkouts, it will feature an intuitive and powerful interface, it will be adjustable to any hotel, it will ensure no overbooking issues arise, will keep track of customer rewards points, and lastly, feature a cross-shift note system.

2.1. Requirements

As part of fleshing out the scope of your requirements, you'll also need to keep in mind both your functional and non-functional requirements. These should be listed, and explained in detail as necessary. Use this area to explain how you gathered these requirements.

2.1.1. Functional.

- User needs to have a private shopping cart – this cannot be shared between users, and needs to maintain state across subsequent visits to the site
- Users need to have website accounts – this will help track recent purchases, keep shopping cart records, etc.
- You'll need more than 2 of these...

1	Release occupancy
2	Fill vacancy

TABLE 1. USE CASES

2.1.2. Non-Functional.

- Security – user credentials must be encrypted on disk, users should be able to reset their passwords if forgotten
- you'll typically have fewer non-functional than functional requirements

2.2. Use Cases

Use Case Number: 1

Use Case Name: Booking and Maintaining Rooms

Description: Administrators at motel and hotel businesses will be able to block rooms once they are booked by customers or repairs/cleaning needs to take place.

Use Case Number: 2

Use Case Name: Reservations

Description: Interact with a calendar in order to properly plan out and schedule rooms to be booked and blocked to prevent overbooking.

Use Case Number: 2

Use Case Name: Checkout

Description: A shopper on our site has finished shopping. They will click on a "Checkout" button. This will kick off a process to calculate cart total, any taxes, shipping rates, and collect payment from the shopper.

You will then need to continue to flesh out all use cases you have identified for your project.

2.3. Interface Mockups

At first, this will largely be completely made up, as you get further along in your project, and closer to a final product, this will typically become simple screenshots of your running application.

In this subsection, you will be showing what the screen should look like as the user moves through various use cases (make sure to tie the interface mockups back to the specific use cases they illustrate).

3. Project Timeline

Go back to your notes and look up a typical project development life cycle for the Waterfall approach. How will you follow this life cycle over the remainder of this semester? This will usually involve a chart showing your proposed timeline, with specific milestones plotted out. Make sure you have deliverable dates from the course schedule listed, with a plan to meet them (NOTE: these are generally optimistic deadlines).

4. Project Structure

At first, this will be a little empty (it will need to be filled in by the time you turn in your final report). This is your chance to discuss all of your design decisions (consider this the README's big brother).

4.1. UML Outline

4.2. Design Patterns Used

Make sure to actually use at least 2 design patterns from this class. This is not normally part of such documentation, but largely just specific to this class – I want to see you use the patterns!

5. Results

This section will start out a little vague, but it should grow as your project evolves. With each deliverable you hand in, give me a final summary of where your project stands. By the end, this should be a reflective section discussing how many of your original goals you managed to attain/how many desired use cases you implemented/how many extra features you added.

5.1. Future Work

Where are you going next with your project? For early deliverable, what are your next steps? (HINT: you will typically want to look back at your timeline and evaluate: did you meet your expected goals? Are you ahead of schedule? Did you decide to shift gears and implement a new feature?) By the end, what do you plan on doing with this project? Will you try to sell it? Set it on fire? Link to it on your resume and forget it exists?

References

- [1] H. Kopka and P. W. Daly, *A Guide to L^AT_EX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.