Project Proposal: Concierge Software

Team D

Our Team

Team Leader Heather Young has five years of experience working as a full-stack software engineer, including extensive experience working on front-end applications using modern JavaScript frameworks. She also has experience working on APIs in Node, C# and Java, as well as some experience working with databases at the level of performing SQL queries. In addition, she is well-versed in Agile tools and methodologies and has done her fair share of DevOps implementations and troubleshooting.

John Fritter is relatively inexperienced in this type of software development, having spent most of his time as a student focused on designing embedded systems in C. But he's worked with Node before, done stuff with databases and in his free time likes to play with computers. He will just try to read the instructions and do his best.

Omar Omer is a computer science student at Fort Hays State University. Outside of school work he spends most of his reading and playing sports. Although Omar does not have any real world experience he still loves to learn about technologies in his own time. He has experience with some programming languages such as C, C++, Python, and Java.

Panharith Menh recently finished Information Technology of Management (ITM) at AUPP, and currently is a Computer Science student at Fort Hays State University (FHSU). During his studies at AUPP, he worked on React developing mobile applications, and worked on designing websites for school projects. He has experience with some programming languages such as JavaScript, React JS, and PHP.

Introduction

In an era of digital transformation, the hospitality industry is constantly evolving to meet the ever-changing needs of both guests and hotel management. Today, we present an innovative solution – Concierge Software – designed to revolutionize the guest experience at resort hotels and streamline operations for hotel management.

At the heart of our project lies a dual challenge: the perspective of the guest and that of hotel management. For guests, planning activities during a vacation can be a time-consuming endeavor, often resembling the complexities of a work schedule. Additionally, hotel management grapples with the need to optimize operations, reduce costs, and make informed decisions. We aim to simplify this process, enhancing the relaxation and enjoyment of guests while simultaneously providing resort hotels with a cost-effective alternative to traditional concierge services, reducing the burden on front desk staff, and offering valuable insights through automated reports for data-driven decision-making.

Outlined Steps

Problem Diagnosis

The problem our project addresses can be looked at from two points of view, that of the guest/customer and that of hotel management and staff. On the one hand, guests at resort hotels need a way to schedule their activities while on vacation that is easy and convenient for them. Coordinating activities while on vacation can be as much work as going to work is. So anything that can help ease that path and make time spent at a resort more restful, would be an advantage for the guests and a selling point for the resort hotel. In addition, customers like to feel special and they like to feel like they are getting extra value for their dollars spent. A digital concierge, properly designed and implemented, can address all these customer needs. At the same time, hotel management wants to provide added value to their guests while keeping costs down and profit margins up. Once installed, a digital concierge would be less costly to run than employing people to work as real concierges. It would also reduce demand on front desk and other staff who may fill those functions in the absence of an actual concierge being at the property. Hotel management would also benefit from easy access to automated reports that detail how the resort's services and amenities are being used. This would help with things like setting staffing schedules based on busy times and determining which services to keep or cut based on usage and popularity.

Proposed Treatment

Our Concierge software would provide a web-based and mobile application that could be accessed only by hotel guests. The landing page of the UI would offer a choice of two main modes: a calendar view, where guests can see activities available by date, and a service-based view containing a display of icons representing particular services or venues. The calendar view will default to whatever the current day is, but will provide the option to change the display to view by week or month. From there, the user can click on any of the activities to get more detailed information or to make a reservation. The service-based page is for users who know exactly what they want to sign up for and they want to check availability or make a reservation. Clicking on any of the service icons will take them to a homepage for that service. This may actually be the same page that the user arrives at by clicking on a link from the calendar. Although when navigating from the calendar view, date and time may be passed along to refine what users are shown on the service homepage.

There will be a management dashboard where hotel administration will be able to enter, modify and delete the activities, availability and capacity for services offered based on date and time frames.They will have the capability to add and remove services. We will also implement a reporting function that will allow administrators to download reports based on time frames and filtering by service.

Plan of Work

We will start by creating the framework for a web application, API and database. For the web application, we will use a modern JavaScript framework that is scalable and reactive, such as Vue or Angular. We will use Node.js for the API and Microsoft SQL for the database.

Then we will architect the solution and map out the needed database tables and schema based on the components we'll need to create. For example, each service will have its own corresponding database table with the service-specific attributes reflected in the column names or fields.

We will set up two repositories in GitHub, one for the web app and one for the API. We will set up a Jira board in Atlassian, which we will use to manage our workflow for the coding tasks we need to accomplish. We will create stories and assign them amongst ourselves to keep track of who is doing what, how much we've accomplished, and what we have left to complete.

We'll create a homepage for the app that reflects the requirements described above. We'll create the service homepages first, starting with one and modeling the others after that one, making modifications as necessary. We'll aim to include at least three services, with the idea that we can expand on that later as time allows. With those complete, one of us can work on the calendar view and one of us can work on the service icon view. At the same time, someone can set up the database tables and begin implementing the functionality for adding, deleting and modifying available events and reservations in the API.

The success of our Concierge Software project will be determined by a multifaceted approach aimed at ensuring the software's functionality, user experience, and alignment with project objectives. To assess the software's functionality and robustness, we will employ rigorous unit testing throughout the development process. This will involve systematically testing individual components and functions to verify their correctness, helping to identify and address issues at an early stage. Additionally, system testing will be crucial to evaluate the software as a whole, ensuring that all integrated components work harmoniously, meeting user requirements, and providing a seamless experience for guests and hotel management.

To gauge the software's success in terms of user experience and usability, we will conduct mock user testing. This will involve collecting feedback from potential users and classmates to assess the software's user-friendliness, efficiency, and overall satisfaction. Insights from these tests will be invaluable in refining the software and enhancing its user-centric design. Furthermore, our commitment to documentation quality, project management efficiency, and innovative problem-solving will serve as essential markers of success. Regular peer and instructor feedback, adherence to project timelines, and the ability to adapt and innovate in response to project challenges will collectively contribute to the project's success within the academic context. By combining these evaluation methods, we aim to deliver a Concierge Software solution that not only meets academic standards but also demonstrates excellence in functionality and user experience.