Group 11

CSCI441VA

September 11, 2022

Group 11 Software Proposal

Appointment Management System

Discord: https://discord.gg/CxUQynSe

GitHub: https://github.com/JamesChapmanNV/CSCI_441_group_11.git

Problem Diagnosis:

Our country is, or at least was, meant to be built on the backs of small businesses. Small businesses have a need for a fast and cheap management systems for their staff to make appointments with customers. These appointments would need to take the customer's needs and translate them into the necessary appointments with the correct entity within the business. An example would be that of a massage parlor. The front desk scheduler of the business would need to be able to organize available masseuses and customers as well as maintain all the information in an easy-to-access, readily-available format.

Carpe Diem Massage Company Carpe Diem Massage Company is small massage company that is expanding to a new building. They will have 4 separate rooms for massages, allowing for increased sales and more flexible scheduling. They need an appointment scheduler for the front desk. When a customer comes in (or calls), the front desk should be able to access available appointment times, collect necessary information from customer, and book

appointments. There should be some functionality for the customer to choose a preferred masseuse and to search available appointments for a specific masseuse. It should also record the clients preferred type of massage, such as deep tissue, Swedish, hot stones etc. A masseuse must be able to ask for their schedule to see which room they need to be in, and at what time. For small businesses customer retention can be an issue, therefor being able to keep track of a clients last massage would make it easier to reach out to those that haven't been in for a while with reminders or promotional deals.

Project Description:

Carpe Diem Massage Company is open Mon - Fri, 9AM - 5PM. Each appointment is 1 hour long and is only allowed to be scheduled on the hour. Each room will have 8 appointment slots per day (9, 10, 11, 12, 1, 2, 3, 4). Every day will have 32 available appointments. Each appointment slot shall be available or booked. To book an appointment, the front desk employee must assign a masseuse and customer through the appointment scheduler. Payments for the massages are handled between the customer and the masseuse. All massages (1 hour) are charged at a flat rate.

Each masseuse has a schedule of availability, which is decided at time of hire, and the schedule shall not change from week to week. A masseuse can only do one massage at a time, so there must be some record to prevent overlap. Other information to be stored for each masseuse include name, phone number, email, address, license number, employee number, etc.

Each customer should have basic information stored that is collected at the time of appointment schedule. This should include name, phone number, and email.

Potentially, this application should email reminders to customers 24 hours in advance of their appointment time.

Pull, add, remove, update, analyze would all be necessary functions to the scheduler in an appointment app. A search function will allow the front desk employee to quickly access a client's personal information. Each of these functions perform I/O, so a method of record storage will be used in the form of a MySQL database. To allow an end-user to easily and efficiently work with the records, a graphical user interface will be implemented to manage the data with minimal effort.

Plan of Work:

- Design and implement a multi-threaded main driver written in Python responsible for overseeing appointment management back-end and front-end functionality.
- From the driver, use threading to deploy a graphical user interface and conduct back-end operations in parallel.
- Incorporate a modular and responsive user interface with capabilities to authenticate/de-authenticate a user and allow administrative staff access to management-level functionality.
- Design and implement an interface to access and edit the task scheduler.
- Design and implement an interface to access and edit available client records and current staffing opportunities to service a client.

Tools and Resources:

- MySQL – Store and manage staff, customer, and clientele information.
- Python - Backend driver that displays front-end and is used to process data when needed.

Team Profile:

Dan – Proficient in C++, JAVA, JS. Beginner in PYTHON. Decent at professional and technical reading and writing. Needs improvement on server-side applications and GUI although relatively proficient at command line UIs.

James – Proficient in C++, Java, Python, and MySQL. Strong organization & design skills.

Landon – Proficient in Python with a small amount of commercial experience. Solid level of understanding with GUI design and development in Python using threading. Has worked with interfacing MySQL with Python, but lacks on the concepts of database design.

David – Proficient in Java, Python, and C++. 10+ years' experience running a small business.

Contributions So Far:

Dan – Proposal template and part of diagnosis section.

James – outline business needs and project description

Landon – Plan of work, tools and resources

David – Small contribution to problem diagnosis and project description