Appointment Management System

Carpe Diem Massage Company

Fort Hays State University - Department of Computer Science

Software Engineering Fall 2022

Group Number 11:

James Chapman - jachapman3@mail.fhsu.edu

Landon Crispin - lccrispin@mail.fhsu.edu

Daniel Ring - dmring2@mail.fhsu.edu

Project URLs:

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

Discord - https://discord.gg/qfC3P6bM

Submission:

25 September 2022 - Report One - Final

Table of Contents

Distribution of Work

- James Chapman
  - UML package diagram & ERD
  - Gantt Chart
- Landon Crispin
  - Connectors and Network Protocols
- Dan Ring
  - Editing and formatting of Report One – 50%
  - Architecture Styles – 100%
  - Mapping Systems to Hardware – 100%
  - Global Control Flow – 100%
  - Hardware Requirements – 100%
  - Project Size Estimation – 100%
  - Plan of Work – 100%
  - References – 50%

Customer Statement of Requirements

Problem Statement

Our country is, or at least was, meant to be built on the backs of small businesses. Small businesses have a need for 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 a 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 customers, and book appointments. There should be some functionality for the user to choose a preferred masseuse and to search available appointments for a specific masseuse based on the client's needs and desires. 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, therefore 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.

Carpe Diem Massage Company is open Mon - Fri, 9AM - 5PM. Each appointment is 1 hour long and must be scheduled at the top of 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 open 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 their basic information collected at the time of their first appointment scheduling. 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 work with the records easily and efficiently, a graphical user interface will be implemented to manage the data with minimal effort. A user

The business would enjoy the ability to add multiple users with different levels of access based on their position within the company. The owner should have Admin status and the ability to perform all functions within the software.

The software should have a pleasing display and easily accessible menus denoting all the necessary features. Minimum interface is required for functionality; however, should time and budget allow it, overall greater UI and software functionality in general would be appreciated.

Decomposition into Subproblems

1. Setting up Database for appointments
2. Implement Database Access
3. Design UI
4. Implement UI
5. Setup Database for masseuses, rooms, and services.
6. Update UI
7. Implement Admin/User access and privileges.
8. Implement email reminder system.

Glossary of Terms

Admin – The owner of the company and any other so designated by them. Has full access to all software functionalities.

Appointment - Agreed-upon time. Should be able to be canceled if need be.

Booked - Room, customer, and masseuse cannot be booked at the same time

Customer/Client - Person getting massage. Basic information needed.

Massage type - Available options for specialized massage.

Masseuse - A licensed employee of Carpe diem. Basic information and a weekly available schedule should be collected at time of hire.

Masseuse availability - Important job requirement to have same availability each week.

Open - Available for appointment to be scheduled at a certain room at a certain time.

Room - Labeled 1, 2, 3, 4

User – Employee of company with varying levels of access to the system up to and including Admin privileges. Minimal requirements are to make and remove appointments.

System Requirements

Enumerated Functional Requirements

Identifier	PW	Description
REQ-01	5	The system shall present an interface to manage appointments and allow the user to manage the appointments.
REQ-02	2	The system shall allow the user to Create, Read, Update, Delete customer information.
REQ-03	4	The system shall allow the user to Create, Read, Update, Delete masseuse information, scheduling, and availability.
REQ-04	4	The system shall interface between the front-end user interface and the cloud hosted SQL database.
REQ-05	2	The system shall allow for the 4 rooms to be booked at the same time, while each room, masseuse, and customer shall NOT be double booked
REQ-06	3	The system shall conduct GUI management and any operations involving data processing in two separate threads to allow for seamless and smooth experience to the end-user without delay.

Analysis of Enumerated Functional Requirements

The main functional requirement of this application is to allow a user to work with the appointment scheduling database and its appropriate tables (REQ-01). A user should be able to manipulate customer information (REQ-02) and manipulate masseuse availability and information as well (REQ-03). To add flexibility into the scheduling system, the system shall allow the booking of all rooms as required by demand and availability. The software shall not allow overbooking. (REQ-05).

In order to do these things, an interface will allow components of the GUI to interact with the backend database manager (REQ-04). Finally, because the GUI and database management interface are components which require their own processing power, two threads will be incorporated to distribute load and allow for simultaneous processing (REQ-06).

Enumerated Non-Functional Requirements

Identifier	PW	Description
REQ-07	1	The system should present the data in an easy to use, accessible format.
REQ-08	2	The system should allow appointments to be scheduled using hourly blocks and be restricted to certain periods of time.
REQ-09	1	The system should incorporate a login system which will provide the user access to the software with the appropriate level of permissions.
REQ-10	1	The system should use an admin management system to allow a manager to change certain parameters about the scheduling behavior or allow a manager to change users and permissions.
REQ-11	3	The system should allow appointments to be scheduled only if the time preferred is not already booked.
REQ-12	2	The system should allow the end-user to specify the type of massage to be serviced to the customer and other specifics about the type of work to be performed.
REQ-13	1	The system should provide a way to notify the customer of an upcoming appointment to remove the need for manual notification.

Analysis of Enumerated Non-Functional Requirements

Requirements listed here are not out of necessity; however, they will significantly aid in the quality of experience in using the software.

For scheduling, the system will allow scheduling to occur only in the blocks that they can be scheduled. It should allow for the changing of the number of rooms available and the store hours of operation in a global manner, (REQ-10) and only if the time to schedule is not already booked (REQ-13). Different UI panels for admin management (REQ-12) and user login (REQ-11) will be incorporated to assign and compartmentalize user access and permissions.

For miscellaneous yet important functionality of the software, an easy to use and accessible format shall be available to the end user (REQ-09). In addition to basic scheduling requirements, the scheduler shall indicate the type of massage/service to be performed on the scheduled time slot (REQ-14).

On-Screen Appearance Requirements

Identifier	PW	Description
REQ-14	2	The screen shall display separate pages to manage the schedule, customer information, masseuses and their availability in an easy-to-read format.
REQ-15	1	The screen shall be responsive and adapt to various sizes to be able to adjust to multiple different devices as necessary.
REQ-16	4	The screen shall incorporate a nav bar for ease of navigation among different features and functionality.
REQ-17	2	The screen shall display a management interface for a manager/operator to change and adjust parameters and configuration of the schedule to specify how the schedule will operate.

Analysis of On-Screen Appearance Requirements

The user interface of the software needs to provide functionality for the end-user and provide an appearance in such a way that it is easy to understand and easy to use.

Separate pages/windows will address functionality to each individual component (REQ-16) – a window to manage, view, and edit the schedule, a window to manipulate customer information, and a window to manage, view, and edit masseuses and their availability. A navigational bar will provide a way to access the different windows incorporated into the program (REQ-18).

The screen will be responsive, resizable, and be able to adapt to a number of different screen configurations and operating systems under the premise that Python is in use (REQ-17).

Functional Requirements Specification

Stakeholders

Front desk employee - The primary user of the appointment management software.

Masseuse employee - Person customers get scheduled with.

Customer - Person needing to be scheduled with masseuse employee.

Owner – Owner of Carpe Diem Massage Company.

Actors and Goals

User (Initiating) – Depending on extra privileges set by Admin, user's goals are to create and manage clients and appointments. Extra privileges include creating and managing masseuses and services as well as setting store hours and appointment length.

Admin (Initiating) – Goals are to create and manage users and their privileges. Also to create and manage clients, appointments, masseuses, and services as well as setting store hours and appointment length.

Database (Participating) – Goals are to authenticate and connect to SQL database allowing for the transfer and manipulation of data stored there.

ApptChecker (Initiating) – Goal is to check the current date against the date of all scheduled appointments and sends an email to the clients with appointments the next day.

ReminderEmail (Initiating) – Goal is to schedule email reminder to be sent 24 hours before appointment time. This shall be scheduled at time of scheduling.

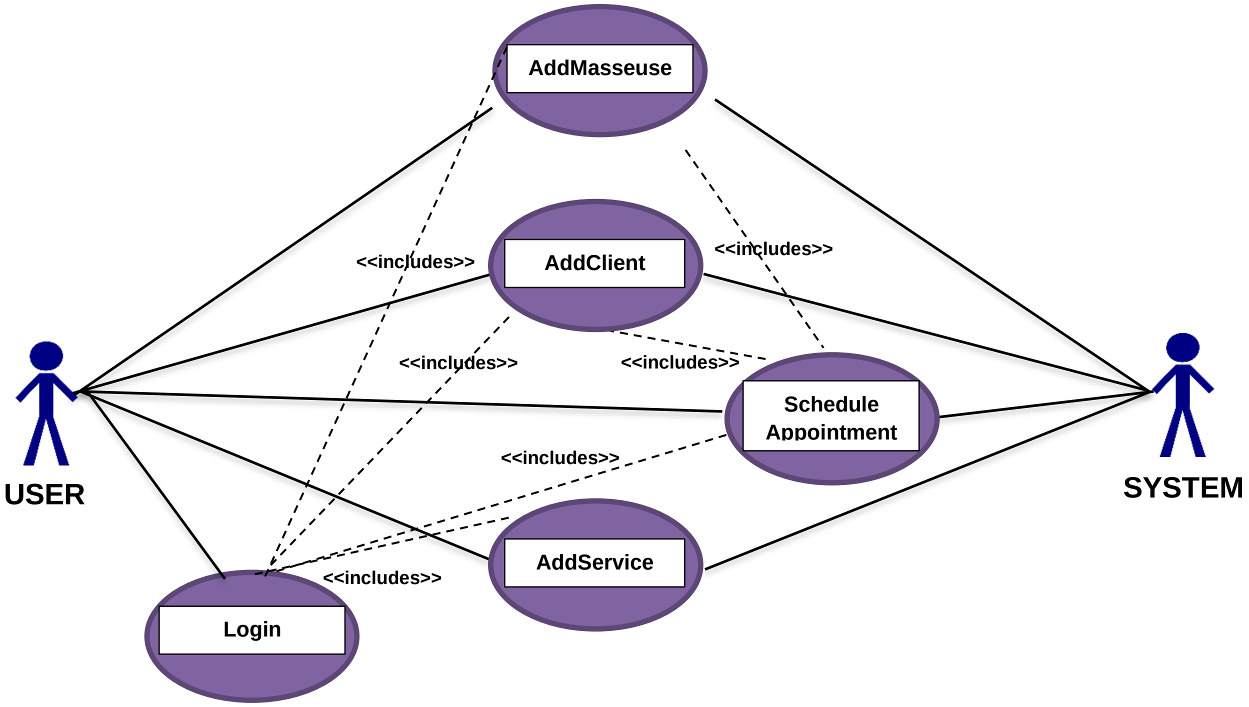
Use Cases

Casual Description

Use Case	Name	Description	Requirement
UC-01	Login	Ensure server connectivity and authenticate user.	REQ-01, REQ-04, REQ-09, REQ-10, REQ-14 - REQ17
UC-02	AddClient	Allows user to add new client information to database.	REQ-02, REQ-04, REQ-14 - REQ17
UC-03	AddMasseuse	Allows user to add new masseuse details.	REQ-03, REQ-04, REQ-14 - REQ17
UC-04	EditClient	Allows user to edit or delete existing client info.	REQ-02, REQ-04, REQ-14 - REQ17
UC-05	EditMasseuse	Allows user to edit or delete existing masseuse info.	REQ-03, REQ-04, REQ-14 - REQ17
UC-06	CheckSchedule	Allows the user to check the scheduled appts on a given day.	REQ-01, REQ-04, REQ-06, REQ- 07, REQ-14 - REQ17
UC-07	GetMasseuseSched	Allows the user to check the scheduled appts of a given masseuse.	REQ-03, REQ-04, REQ-06, REQ- 07, REQ-14 - REQ17
UC-08	MakeAppt	Allows the user to schedule an appointment with desired services and desired masseuse.	REQ-01, REQ-04, REQ-05, REQ-08, REQ-11, REQ-12, REQ-14 - REQ17
UC-09	RemoveAppt	Allows user to delete an appointment.	REQ-01, REQ-04, REQ-05, REQ-14 - REQ17
UC-10	SetandGetShopHours	Allows the user to set and get hours of operation.	REQ-07, REQ-08, REQ-10, REQ-14 - REQ17
UC-11	SetandGetTimeslotLen	Allows the user to set and get the appointment length. (Same for all appointments.)	REQ-07, REQ-08, REQ-10, REQ-14 - REQ17
UC-12	EmailReminder	Emails client 24 hours before appointment.	REQ-13
UC-13	SetandGetServices	Allows the user to set and get the services	REQ-12, REC-14 – REC 17

		offered at the location.	
UC-14	Admin	Super user that is able to add/update/remove users and their privileges.	REQ-04, REQ-09, REQ-10, REC-14 – REC 17

Use Case Diagram



Traceability Matrix

	PW	UC-01	UC-02	UC-03	UC-04	UC-05	UC-06	UC-07	UC-08	UC-09	UC-10	UC-11	UC-12	UC-13	UC-14
REQ-01	5	X					X		X	X					
REQ-02	3		X		X										
REQ-03	4			X		X		X							
REQ-04	4	X	X	X	X	X	X	X	X	X					
REQ-05	3								X	X					
REQ-06	3						X	X							
REQ-07	1						X	X			X	X			
REQ-08	2								X		X	X			
REQ-09	1	X													X
REQ-10	1	X									X	X			X
REQ-11	3								X						
REQ-12	2								X					X	
REQ-13	1												X		
REQ-14	2	X	X	X	X	X	X	X	X	X	X	X		X	X
REQ-15	1	X	X	X	X	X	X	X	X	X	X	X		X	X
REQ-16	4	X	X	X	X	X	X	X	X	X	X	X		X	X
REQ-17	2	X	X	X	X	X	X	X	X	X	X	X		X	X
Total PW		20	16	17	16	17	22	21	28	21	13	13	1	11	11

Fully-Dressed Description

UC-# and Name	UC-01 Login
Requirements	REQ-01, REQ-04, REQ-09, REQ-10, REQ-14 - REQ17
Initiating Actor	User, Admin
Actor’s Goals	To ensure user is allowed to use the software.
Preconditions	System must be powered on with active internet connection.
Postconditions	All ready for use depending on privileges.
Flow of Events Ask user for username and password. Send username and password to mySQL server via Amazon RDS. If username and password is incorrect ask again, else welcome/home screen.	

UC-# and Name	UC-02 AddClient
Requirements	REQ-02, REQ-04, REQ-14 - REQ17
Initiating Actor	User, Admin
Actor’s Goals	To add new clients to the database.
Preconditions	Valid user or Admin.
Postconditions	New client added to SQL database.
Flow of Events Ask user for customer name, address, email, phone number. Send info to mySQL server via Amazon RDS. Verify customer is not repeated, assign customer ID.	
UC-# and Name	UC-03 AddMasseuse
Requirements	REQ-03, REQ-04, REQ-14 - REQ17
Initiating Actor	User, Admin
Actor’s Goals	
Preconditions	Valid user or Admin.
Postconditions	New masseuse added to SQL database.
Flow of Events Ask user for masseuse name, address, email, phone number, masseuse license number Send info to mySQL server via Amazon RDS. Verify customer is not repeated, assign masseuseID	

UC-# and Name	UC-08 MakeAppt
Requirements	REQ-01, REQ-04, REQ-05, REQ-08, REQ-11, REQ-12, REQ-14 - REQ17
Initiating Actor	User, Admin
Actor’s Goals	Create a valid appointment.
Preconditions	Valid user or Admin.
Postconditions	SQL database updated with appointment information.
Flow of Events Ask user for the date of appointment. Search all masseuse availability tables for all timeslots on that date, Remove available time slot where masseuses are already booked, Remove all timeslots where rooms are already booked, Return available masseuses name for each timeslot/room. Create display of available time slots/room/masseuse that are available/booked	

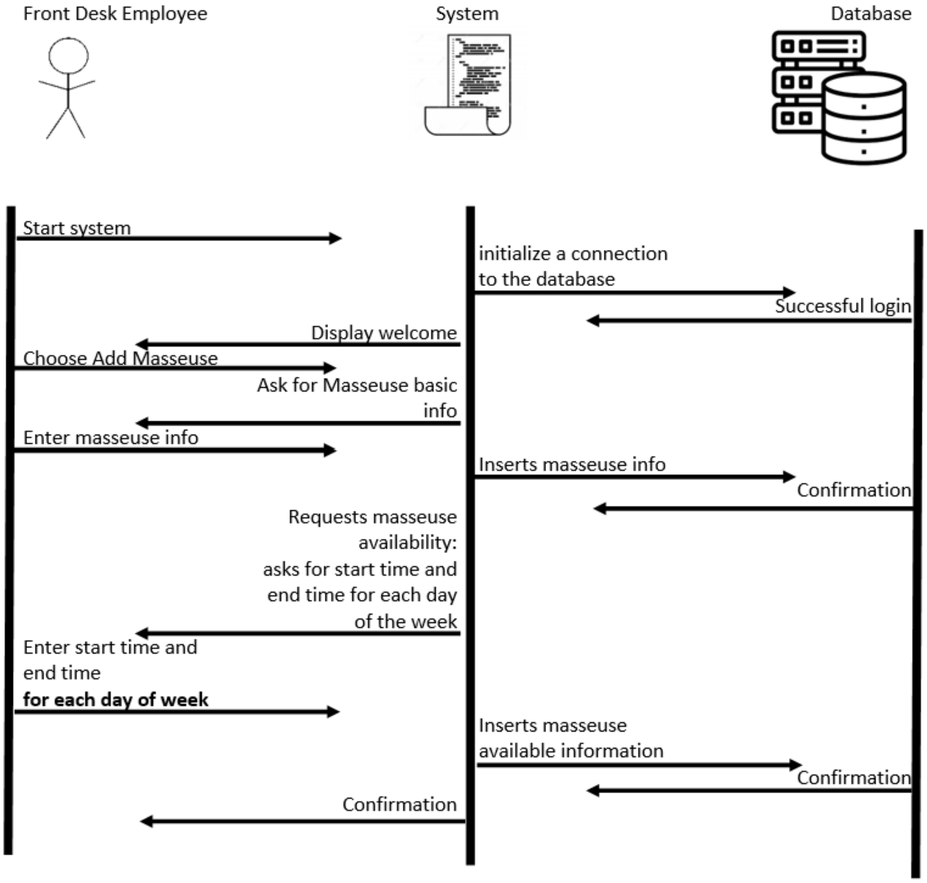
Repeat if user chooses different day Once user chooses appointment, ask for customer ID/name If the customer is new, add new customer Verify customer does not have appointment booked at that time Insert appointment with customer ID, masseuse ID, and room

UC-# and Name	UC-09 RemoveAppt
Requirements	REQ-01, REQ-04, REQ-05, REQ-14 - REQ17
Initiating Actor	User, Admin
Actor's Goals	To delete an appointment from the database, freeing up the slot as available.
Preconditions	Valid user or Admin.
Postconditions	Appointment is deleted and available for new appointment.
Flow of Events Ask for date or name of appointment Display all appointments on that date Choose appointment, display appointment details Ask for confirmation of delete appointment Delete appointment row in appropriate table of mySQL server via Amazon RDS. Key to remove all appointment reminders!	

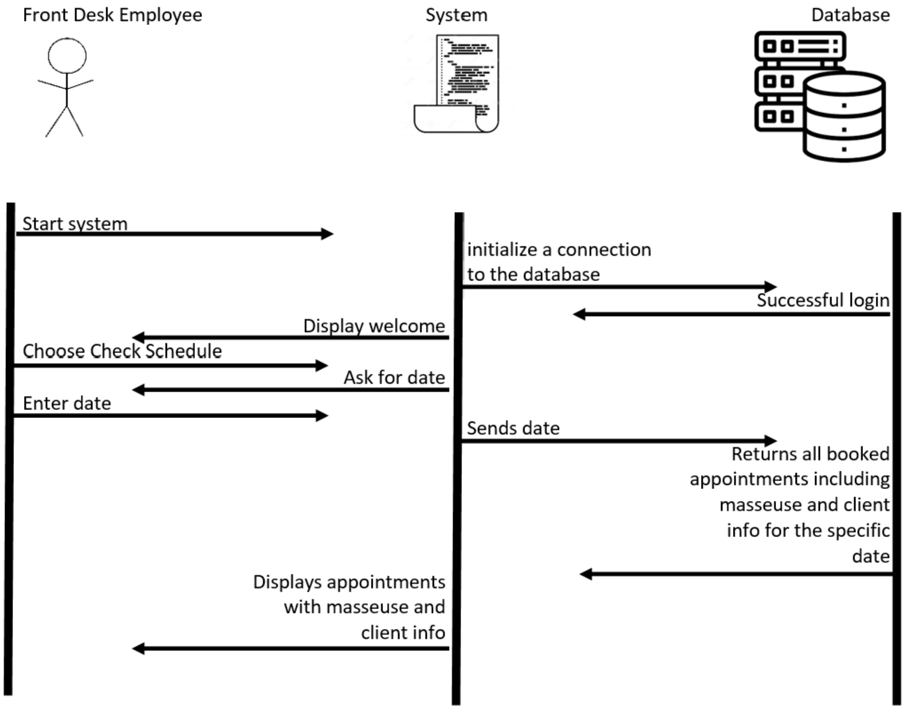
UC-# and Name	UC-12 EmailReminder
Requirements	REQ-13
Initiating Actor	User, Admin
Actor's Goals	ApptChecker
Preconditions	First valid user or Admin to logs on each day.
Postconditions	Emails clients with reminder of appointment the next day.
Flow of Events Script created at appointment creation for timed form email. When time is reached email is auto-sent via script.	

System Sequence Diagrams

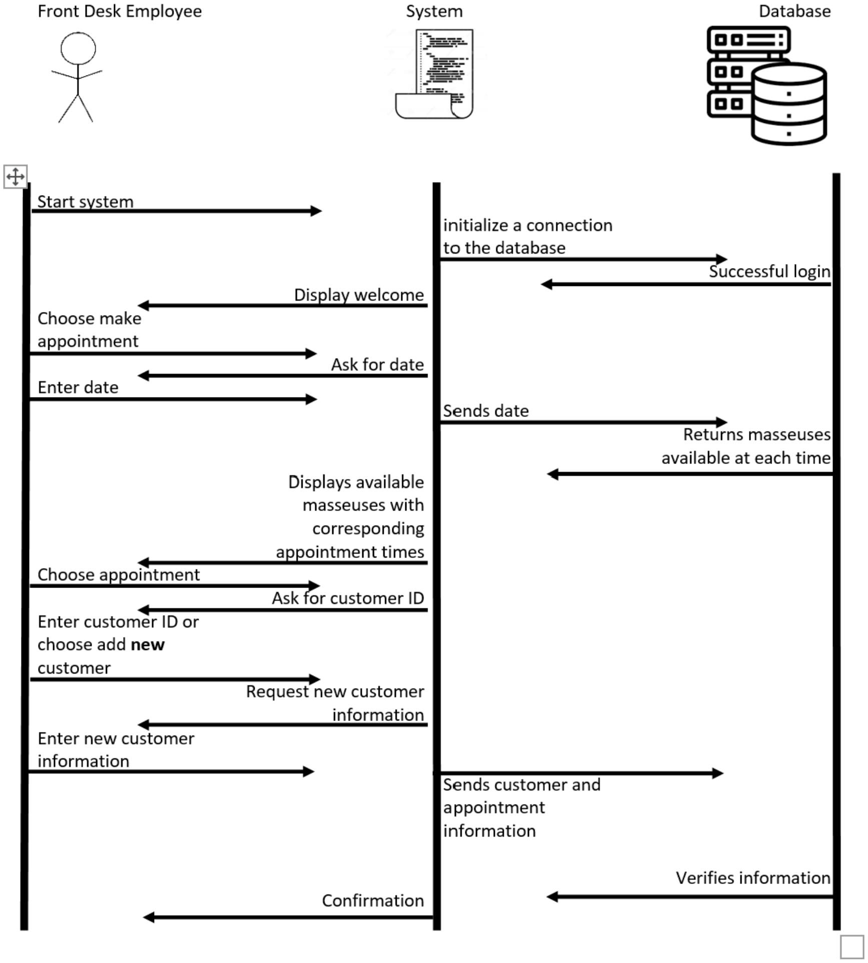
UC-03: Add Masseuse



UC-06: Check Schedule


Front Desk Employee
System
Database
Start system
initialize a connection to the database
Successful login
Display welcome
Choose Check Schedule
Ask for date
Enter date
Sends date
Returns all booked appointments including masseuse and client info for the specific date
Displays appointments with masseuse and client info

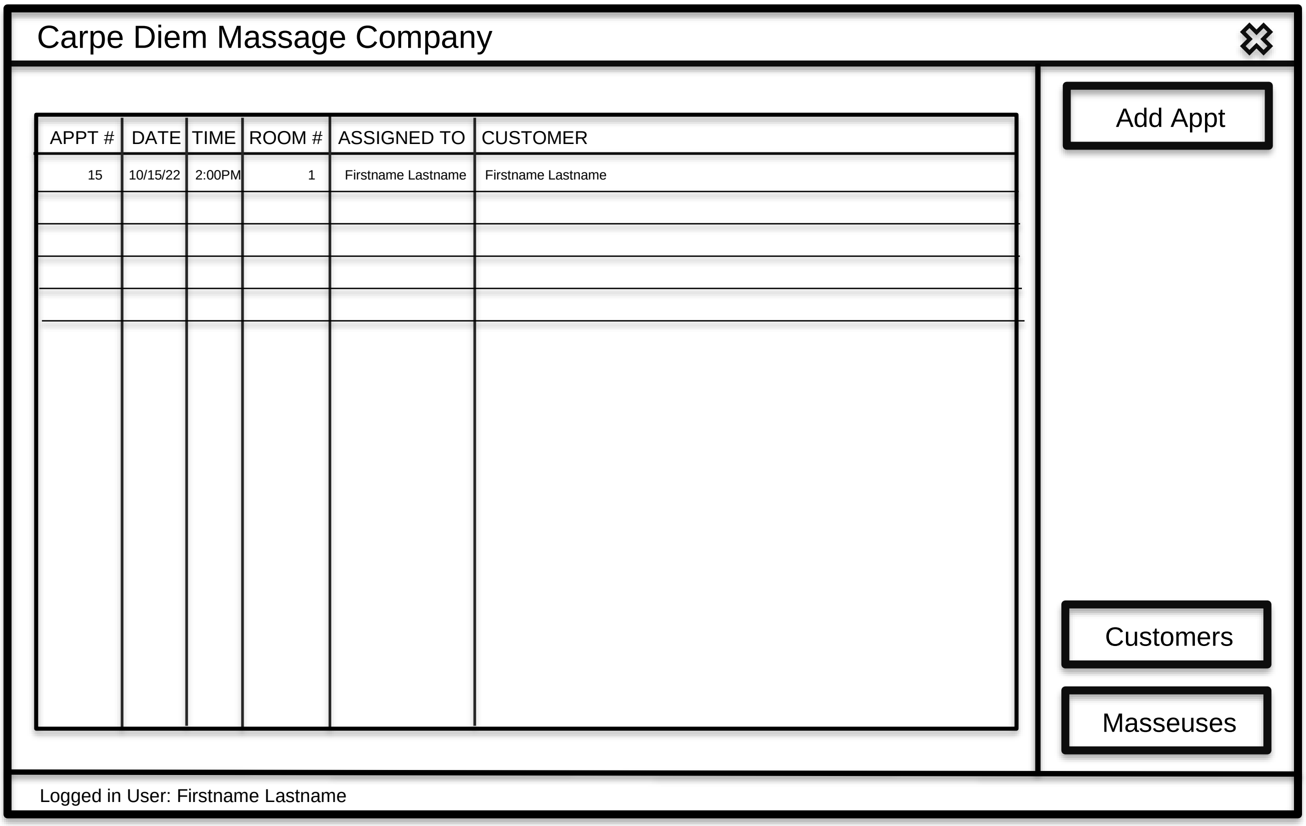
UC-08: Make Appointment


Front Desk Employee
System
Database
Start system
initialize a connection to the database
Successful login
Display welcome
Choose make appointment
Ask for date
Enter date
Sends date
Returns masseuses available at each time
Displays available masseuses with corresponding appointment times
Choose appointment
Ask for customer ID
Enter customer ID or choose add new customer
Request new customer information
Enter new customer information
Sends customer and appointment information
Verifies information
Confirmation

On-Screen Appearance Specification

Preliminary Design

Appointment Scheduler – Main Window



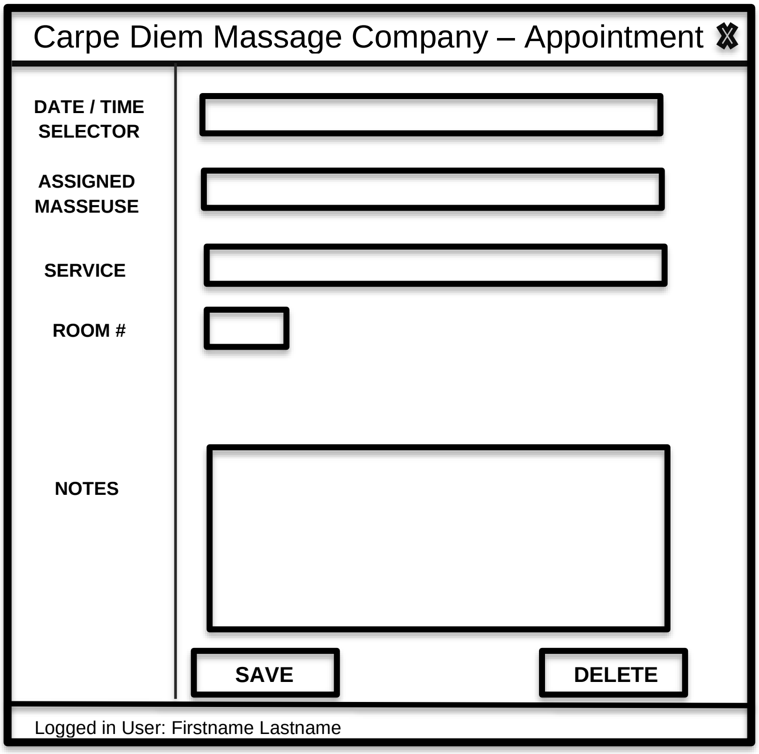
Carpe Diem Massage Company

APPT #	DATE	TIME	ROOM #	ASSIGNED TO	CUSTOMER
15	10/15/22	2:00PM	1	Firstname Lastname	Firstname Lastname

Logged in User: Firstname Lastname

The main window when program execution begins will point to the appointment scheduler which will show active and pending appointments. The appointments each can be double clicked which pulls up the appointment editor where details can be changed or the appointment could be cancelled or re-scheduled as necessary.

Appointment Editor – Popout/Floating Window



Logged in User: Firstname Lastname

Date/Time Selector – A date time picker which will only show available dates and times to choose. The selector will hide or grey out any times unavailable or times in the past.

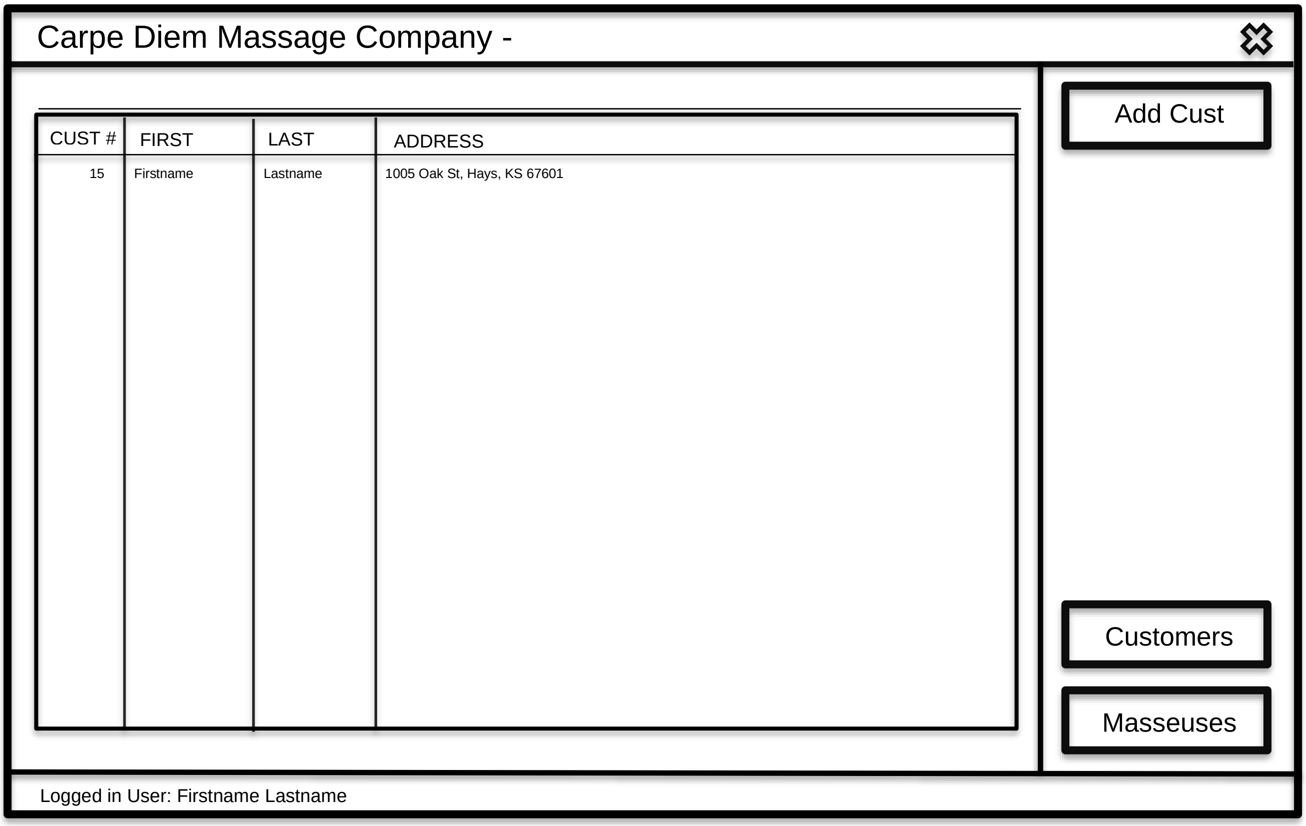
Assigned Masseuse – Based on the time picked, show which masseuses are available in a dropdown.

Service – A description of the service to be performed in a dropdown box.

Room # - A room number dropdown that will allow the user to pick a room # available at the specified time.

Notes – A text box where the user can add any additional notes that a masseuse should know when looking at the service.

Customers – Create, Read, Update, Delete Customers – Popout/Floating Window

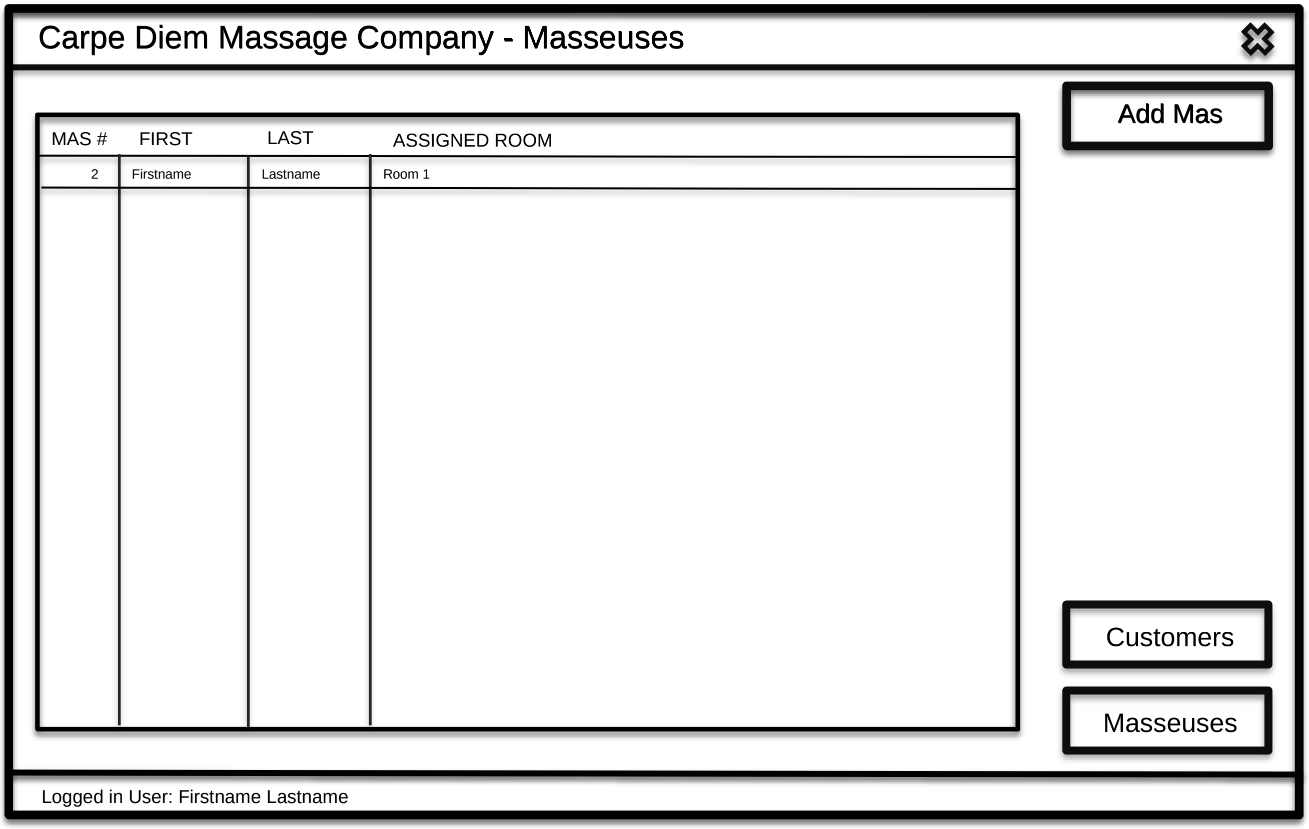

Carpe Diem Massage Company -

CUST #	FIRST	LAST	ADDRESS
15	Firstname	Lastname	1005 Oak St, Hays, KS 67601

Logged in User: Firstname Lastname

A popout/floating window that appears after clicking customers in the main window. Displays a list of customers in the database and allows the ability to add or edit (double click an existing customer), then save the information back to the database.

Masseuses – Create, Read, Update, and Delete Masseuses – Popout/Floating Window



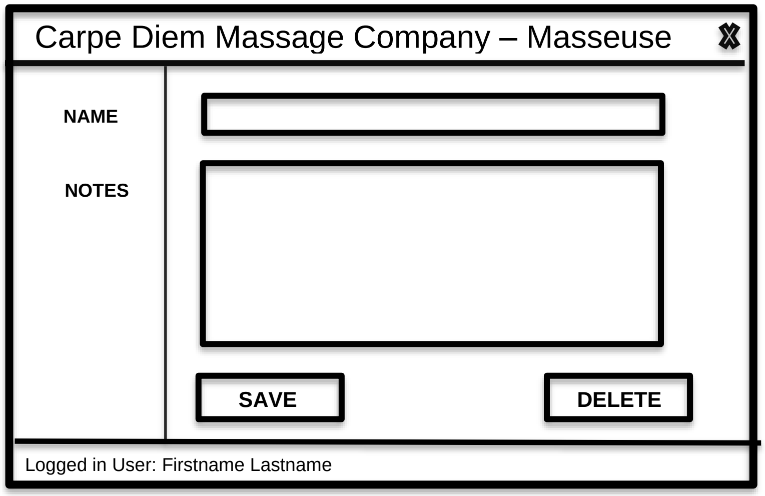
Carpe Diem Massage Company - Masseuses

MAS #	FIRST	LAST	ASSIGNED ROOM
2	Firstname	Lastname	Room 1

Logged in User: Firstname Lastname

A popout/floating window that appears after clicking masseuses in the main window. Displays a list of masseuses in the database and allows the ability to add or edit (double click an existing masseuses), then save the information back to the database.

Masseuse Editor – Popout/Floating Window



Carpe Diem Massage Company – Masseuse

Logged in User: Firstname Lastname

Name – The first and last name of the masseuse.

Notes – A text box where the user can add any additional notes about the masseuse that should be known to a manager.

User Effort Estimation

Log In – Variable Entries + 1 Button Click = 1 Total Click:

1. Enter username
2. Enter password
3. Click log in

View Appointment Scheduler – Assumes User is Logged In – 0 Total Clicks

Add Appointment to Schedule – Variable Entries + 2 Button Clicks = 2 Total Clicks

1. Click add appointment from the main screen
2. Fill in details about the appointment
3. Click Save

Add Customer – Variable Entries + 3 Button Clicks = 3 Total Clicks

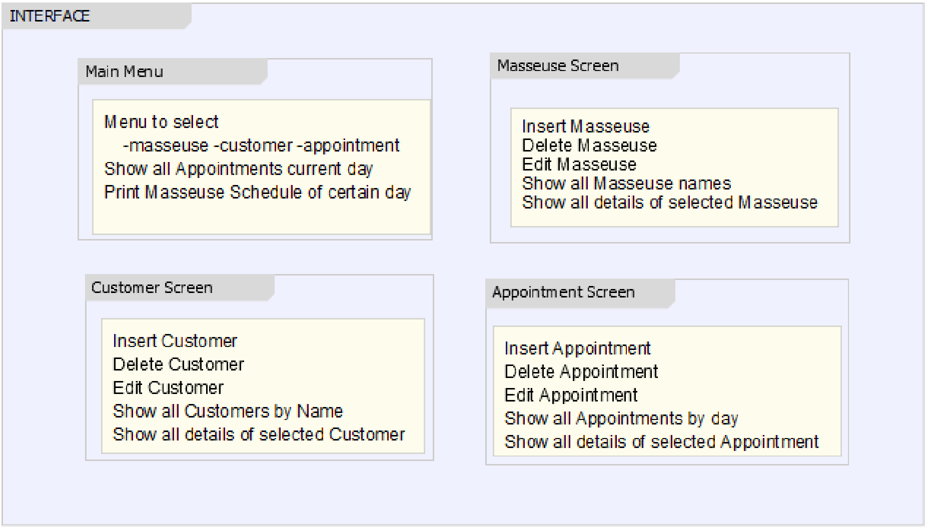
1. Click customers from the main screen
2. Click add customer from the customers screen
3. Fill in details about the customer
4. Click Save

Add Masseuse – Variable Entries + 3 Button Clicks = 3 Total Clicks

1. Click masseuses from the main screen
2. Click add masseuse from the masseuses screen
3. Fill in details about the masseuse
4. Click Save

System Architecture

Identifying Subsystems



INTERFACE

Main Menu

Menu to select
  -masseuse -customer -appointment
Show all Appointments current day
Print Masseuse Schedule of certain day

Masseuse Screen

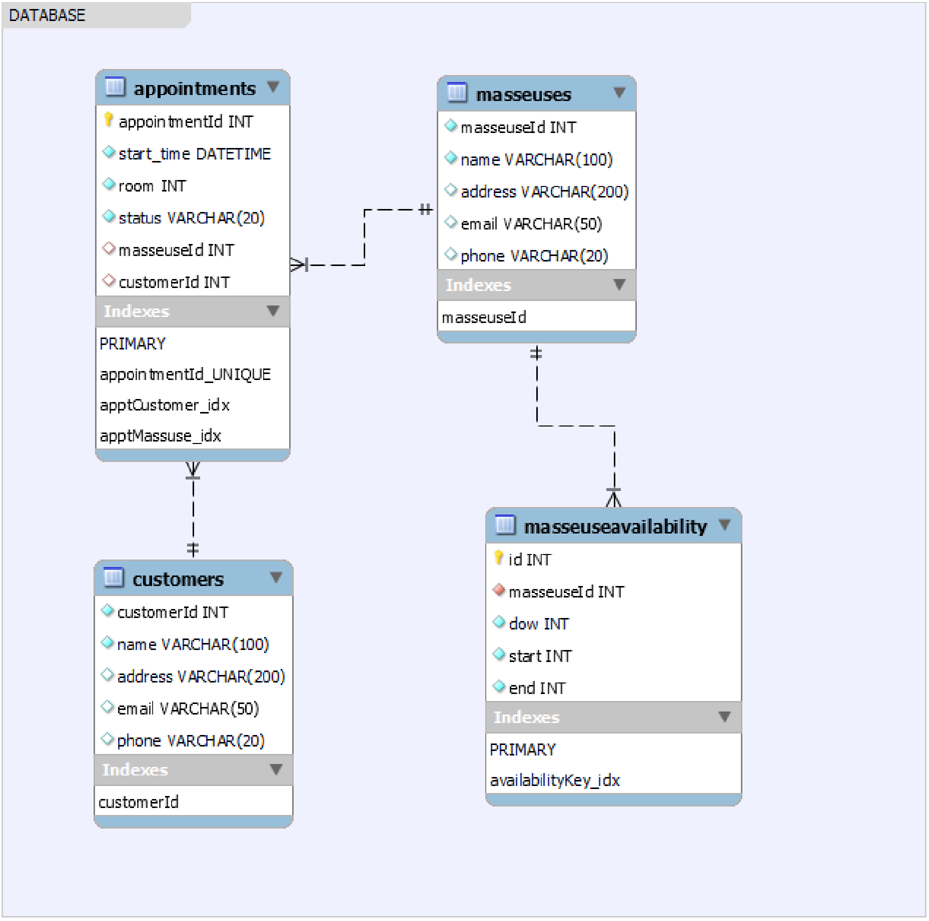
Insert Masseuse
Delete Masseuse
Edit Masseuse
Show all Masseuse names
Show all details of selected Masseuse

Customer Screen

Insert Customer
Delete Customer
Edit Customer
Show all Customers by Name
Show all details of selected Customer

Appointment Screen

Insert Appointment
Delete Appointment
Edit Appointment
Show all Appointments by day
Show all details of selected Appointment



DATABASE

appointments
appointmentId INT
start_time DATETIME
room INT
status VARCHAR(20)
masseuseId INT
customerId INT
Indexes
PRIMARY
appointmentId_UNIQUE
apptCustomer_idx
apptMassuse_idx

masseuses
masseuseId INT
name VARCHAR(100)
address VARCHAR(200)
email VARCHAR(50)
phone VARCHAR(20)
Indexes
masseuseId

customers
customerId INT
name VARCHAR(100)
address VARCHAR(200)
email VARCHAR(50)
phone VARCHAR(20)
Indexes
customerId

masseuseavailability
id INT
masseuseId INT
dow INT
start INT
end INT
Indexes
PRIMARY
availabilityKey_idx

Architecture Styles

The software will follow a client-server architecture as it will rely heavily on access to the SQL database to maintain and update all records of appointments, employees, and customers. The client side would include users and admin as they need to make appointments and update information on customers, employees, services, and the store.

The software will also be a monolithic application architecture as all the UI and data access methods will be all included in a single software application. This means that from the application the user will have full access to the database and its features, methods, etc. in a black box fashion.

Finally, there is a strong emphasis on a database-centric architecture. All the information will be stored and manipulated on the database from the software application. The database and access to it is central to the design and function of the software.

Mapping Subsystems to Hardware

- Local to the company's machine will be the UI, all accessor/mutator/initializing methods for accessing and maintaining the SQL database, and the application itself.
- The SQL database is non-local. The database will be physically managed offsite by Amazon RDS.

Connectors and Network Protocols

TCP / IP

The system connects to the Amazon RDS SQL database using TCP / IP protocol. Connection to the database is then established with a secure, randomly generated username and password. This connector was chosen for simplicity as well as ease of use to communicate data to and from the database. In addition, TCP allows for the assurance that all data sent and received from client to server is obtained in its entirety as opposed to a UDP connection where data would be sent and not verified.

Global Control Flow

Execution Ordinality

Software application is linear. The user initiates the system and in stepwise fashion can perform different requirements. There are a finite number of procedures that are possible with the user being stepped through each in the goals of managing their appointments and business.

Hardware Requirements

- Monitor
- Mouse
- Keyboard
- PC
  - HDD or SDD
  - CPU
  - RAMM
  - Motherboard
  - Network Adapter
  - USB ports
  - HDMI, VGA, DisplayPort, or DVI port
- Internet Access
  - Router
  - Modem
- Cabling

Project Size Estimation

TCF

Technical Factor	Description	Weight	Complexity Weight
T1	Distributed system (running on multiple machines)	0	1
T2	Performance objectives (are response time and throughput performance critical?)	1	1
T3	End-user efficiency	2	3
T4	Complex internal processing	2	2
T5	Reusable design or code	1	3
T6	Easy to install (are automated conversion and installation included in the system?)	1	1
T7	Easy to use (including operations such as backup, startup, and recovery)	3	1
T8	Portable	1	3
T9	Easy to change (to add new features or modify existing ones)	3	3
T10	Concurrent use (by multiple users)	1	2
T11	Special security features	1	3
T12	Provides direct access for third parties (the system will be used from multiple sites in different organizations)	0	1
T13	Special user training facilities are required	0	1

TCF = .6 + (.01*TCF)
Total TCF = 35
TCF = .95

UCP

Use Case	Name	Actor Weight	Complexity Weight	UUCP
UC-01	Login	3	1	3
UC-02	AddClient	3	1	3
UC-03	AddMasseuse	3	1	3
UC-04	EditClient	3	2	6
UC-05	EditMasseuse	3	2	6
UC-06	CheckSchedule	3	3	9
UC-07	GetMasseuseSched	3	3	9
UC-08	MakeAppt	3	3	9
UC-09	RemoveAppt	3	2	6
UC-10	SetandGetShopHours	3	1	3
UC-11	SetandGetTimeslotLen	3	1	3
UC-12	EmailReminder	1	2	2
UC-13	SetandGetServices	3	1	3
UC-14	Admin	3	3	9

UUCP Total = 74

UCP = UUCP * TCF * ECF

UCP = 74 * .95 * 1 = 70.3

Project Management

Plan of Work

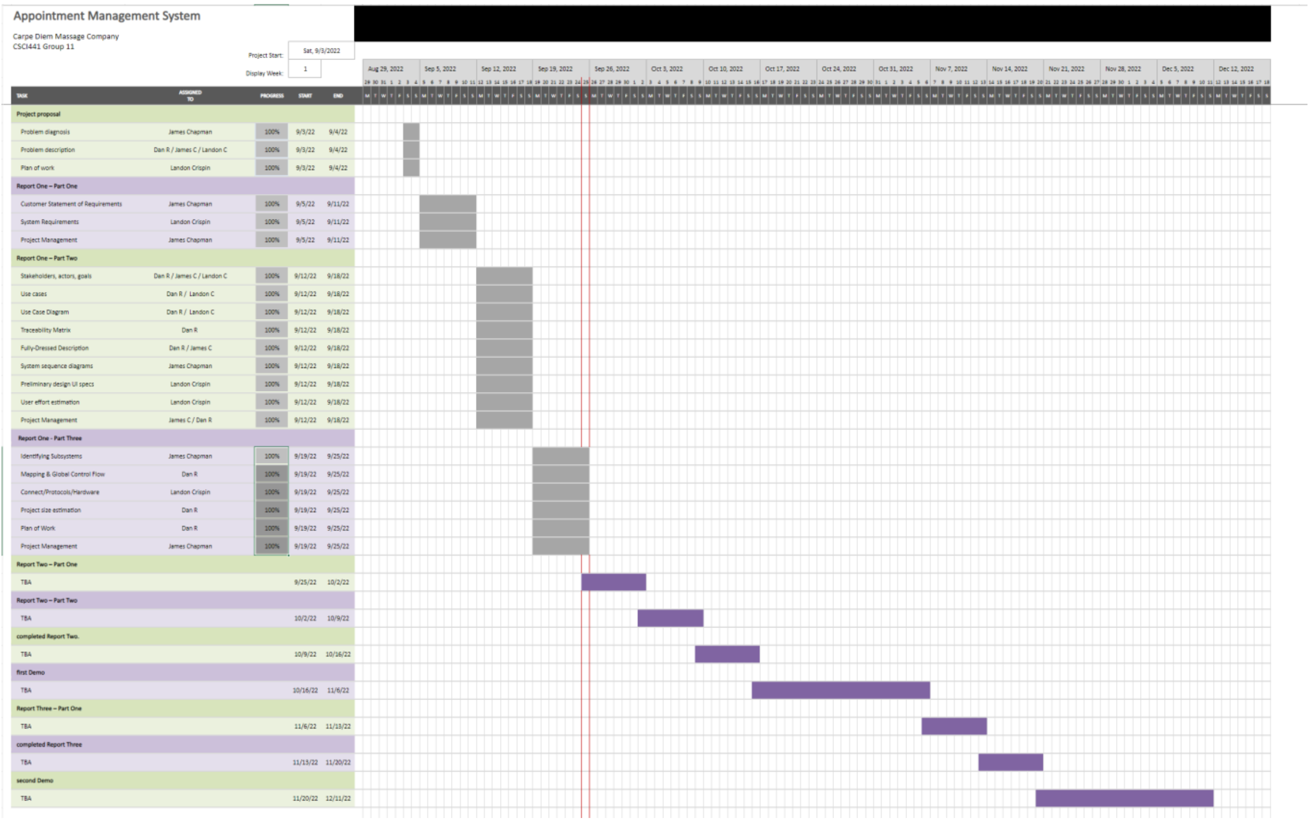
1. Start writing Report 2.
2. Begin implementation of coding within 1 week.
3. Have UC-08 (at highest priority) operational by midterm.
4. Have functional requirements implemented soon after step 3.
5. Test units as they are completed.
6. Have GUI fully functional.
7. Have software debugged and fully documented.
8. Final full Systems tests.

References

1. For descriptions of architectural styles. https://en.wikipedia.org/wiki/Software_architecture#Examples_of_Architectural_Styles_.2F_Patterns
2. …

Gantt Chart

Link to GANTT CHART for CSCI 441 Group 11



Appointment Management System

Carpe Diem Massage Company
CSCI441 Group 11

[redacted]

Project Start: Sat, 9/3/2022

Display Week: 1

TASK	ASSIGNED TO	PROGRESS	START	END
Project proposal				
Problem diagnosis	James Chapman	100%	9/3/22	9/4/22
Problem description	Dan R / James C / Landon C	100%	9/3/22	9/4/22
Plan of work	Landon Crispin	100%	9/3/22	9/4/22
Report One – Part One				
Customer Statement of Requirements	James Chapman	100%	9/5/22	9/11/22
System Requirements	Landon Crispin	100%	9/5/22	9/11/22
Project Management	James Chapman	100%	9/5/22	9/11/22
Report One – Part Two				
Stakeholders, actors, goals	Dan R / James C / Landon C	100%	9/12/22	9/18/22
Use cases	Dan R / Landon C	100%	9/12/22	9/18/22
Use Case Diagram	Dan R / Landon C	100%	9/12/22	9/18/22
Traceability Matrix	Dan R	100%	9/12/22	9/18/22
Fully-Dressed Description	Dan R / James C	100%	9/12/22	9/18/22
System sequence diagrams	James Chapman	100%	9/12/22	9/18/22
Preliminary design UI specs	Landon Crispin	100%	9/12/22	9/18/22
User effort estimation	Landon Crispin	100%	9/12/22	9/18/22
Project Management	James C / Dan R	100%	9/12/22	9/18/22
Report One - Part Three				
Identifying Subsystems	James Chapman	100%	9/19/22	9/25/22
Mapping & Global Control Flow	Dan R	100%	9/19/22	9/25/22
Connect/Protocols/Hardware	Landon Crispin	100%	9/19/22	9/25/22
Project size estimation	Dan R	100%	9/19/22	9/25/22
Plan of Work	Dan R	100%	9/19/22	9/25/22
Project Management	James Chapman	100%	9/19/22	9/25/22
Report Two – Part One				
TBA			9/25/22	10/2/22
Report Two – Part Two				
TBA			10/2/22	10/9/22
completed Report Two.				
TBA			10/9/22	10/16/22
first Demo				
TBA			10/16/22	11/6/22
Report Three – Part One				
TBA			11/6/22	11/13/22
completed Report Three				
TBA			11/13/22	11/20/22
second Demo				
TBA			11/20/22	12/11/22