Beauty Craft

REDEFINING YOUR BEAUTY

Integrated Management System for Beauty Salons

CS Group 30
Supervisor: Mr. Tharindu Wijethilake
Co-Supervisor: Miss. Sithara Fernando

Our Team

CS30

Devin Dissanayake

19000413

Sanjana Rajapaksha

19001274

Ravindu Madhubhashana

19000812

Ruwanthi Munasinghe

19001029

01

Introduction

Beauty Craft

REDEFINING YOUR BEAUTY

Beauty Craft is an integrated management system for beauty salons.

System mainly focuses on efficient and user-friendly reservation management as well as managing organizational work of the salon requiring minimum user effort and time.

The ultimate goal is to maximize the customer satisfaction and increase the revenue by that.

System Overview

Beauty Craft is an integrated salon management system that provides following subsystems to mitigate the above mentioned problems.

User Identification

Level 0

Customer

Receptionist

Service Provider

Level 1

Manager

Level 2

Owner

System Admin

02

Core Functionalities

Services Management

- A **service type** and a **customer type** based on the service nature.
- Assign to available **service providers** based on their specialization.

Improvements

- Service can have a maximum 3 slots with intervals in between.
- Each slot has a specific attributes such as,
  - Slot duration
  - Amount of resources required of each type

Resources Management

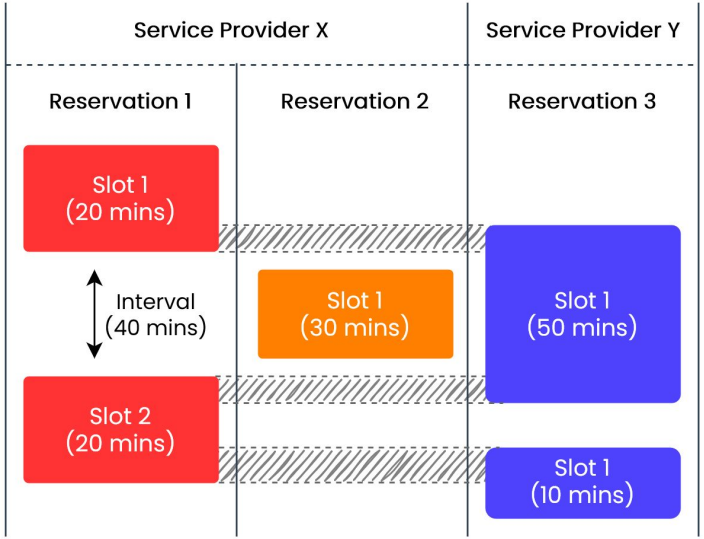
- Added considering the feedback received in proposal defence presentation.
- Initially used only to **check the availability of resources** for the **reservation management**.

Improvements

- **Purchase records** of each resource types is stored within the system with data such as manufacturer, model no, purchased date, warranty expiration date, price.
- Added **new attributes** considering the feedback received in interim presentation.

Reservation Management

- Maximize Service Providers allocation.
- Each slot has a specific **duration** and required amount of **resources**.
- Basic checks for service provider **leaves** and **closed dates**.
- **Availability of resources** is checked for all conflicting slots.
- **Custom made algorithm** to check the possibility of placing a reservation.



Leaves Management

- Main requirement is to **check service providers' availability** and to **calculate salaries**.

Improvements

- Two types of leaves named **Casual leaves** and **Medical leaves**.
- Both leave type have a limit but casual leaves can be taken exceeding the limit.
- Managers handle the requests for leaves from lower level staff members.
- These constraints are added by considering the **interim presentation feedback**.

Salary Management

Total Salary = **Basic Salary** + **Service* Commission** - **Leave Deduction**

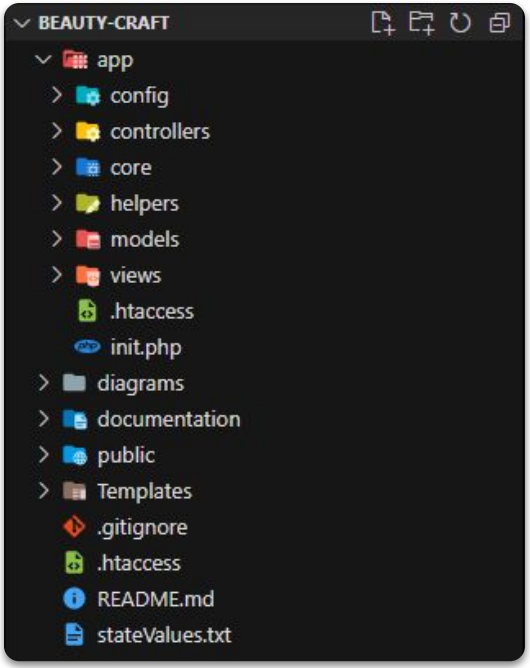
- Basic salary – Based on **staff member type**
- Service commission – A **percentage of income received** from services provided by each service provider
- Leave deduction – Deducting amount based on the **no of casual leaves exceeding the limit**

03

Highlights

MVC Architecture

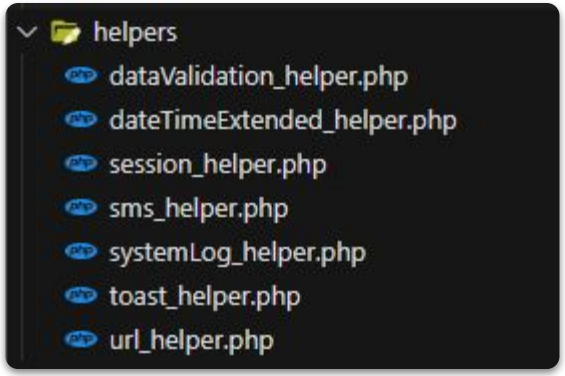
- Separation of **Models**, **Views** and **Controllers**.
- Allows efficient development and maintenance.
- Collaboration made easy, by separating focus of multiple developers.
- Easy to debug and make changes with lesser modifications.



Helper Classes

Separate classes for,

- Data validation.
- Session validation.
- SMS handling



SMS API

A SMS API is used to provide updates for both customers and staff members on,

- Used for mobile verification
- Send confirmation messages
- Inform changes

Provide two priority levels for messages.

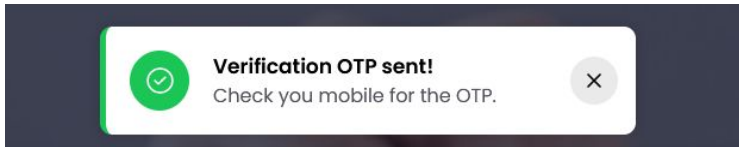
Custom SQL Query Builder

- Developed a query builder from scratch to make DB querying convenient with predefined functions.
- Builds the query based on the function called and parameters passed.
- Reduced complexity of queries and reusing most common SQL query structures.

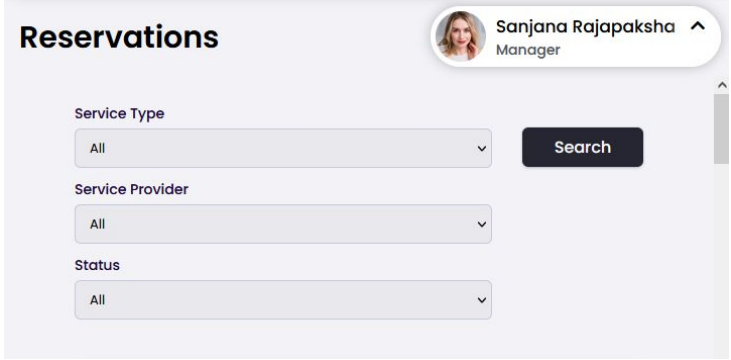
SQL Transactions

- Maintaining ACID properties when modifying multiple tables related to the same operation.

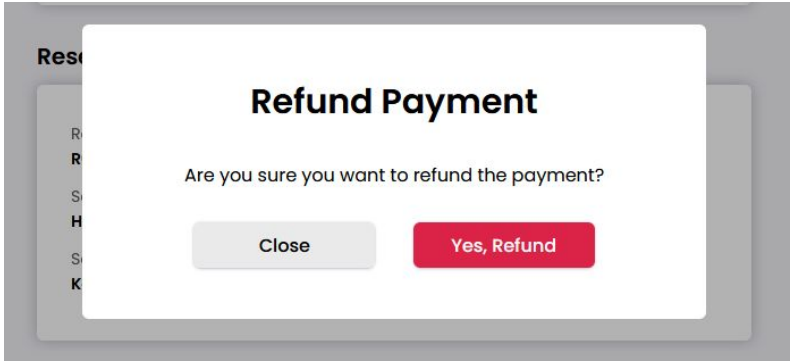
UI Highlights



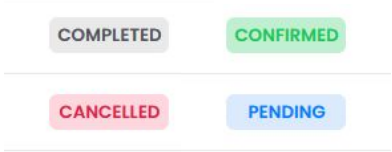
Toast Notifications



Data Filtering



Pop-up Confirmation Modals



Color Scheme

Good Coding Practices

- Organized folder structure.
- Object Oriented helpers classes.
- Code reuse by inheriting Core Classes (Controller, Model).
- Camel Case for variable naming.

04

Demo

Thank You!