VILNIUS UNIVERSITY

FACULTY OF MATHEMATICS AND INFORMATICS

SOFTWARE ENGINEERING STUDY PROGRAMME

Technical document

Centralized sales system

Kipras Kazarinas, Lukas Jasiulionis, Matas Sipavičius,
Nikita Savickis, Tomas Kazonas

Vilnius
2025

Contents

1 Introduction

1.1 Contacts

Team Leader Contacts

In case of any questions regarding documentation or help interpreting it, please reach out to the team leader via the contacts below :)

VU mail:	lukas.jasiulionis@mif.stud.vu.lt
Personal email (prefer this one):	me@lukasja.lt
Add on Discord:	lukasj

1.2 System

Centralized sales system, CPS for short, is a software system designed to provide intuitive and convenient interfaces for employees in the food catering and/or beauty service industry that deal with client management and scheduling on a daily basis. Business owners and system maintainers should also be left satisfied by having minimal overhead when it comes to system integration and management.

The readers may be flexible when it comes to realizing the solutions to problems and requirements within the document. The document serves as a roadmap that can be changed where necessary.

1.3 Requirements check-list

This section contains a check-list that helps track the mandatory system behavior on the system-end and on the employee-end.

1.3.1 Required business functionality for the system

This section concerns functionality in terms of the behavior of the system. Both the manager/super admin user group and the employee user group may not need to directly interact or have control over some of the following functionalities, but rather it is a behavior expected of the system.

- A super-admin should be present and have equal or superior rights to a business manager.
- Changes on the employee end and manager end should only be made by users with authorized credentials.
- Owner contact information should be present, alongside the business name, location and contact information of the owner.

- Business owners and managers can pick between purchasing a beauty service subscription and/or a food service subscription. There is no combined plan for both, if an establishment specializes in both they need to be purchased and managed as two separate subscriptions.
- Business owners and maintainers can modify their employee database.
- Business owners and maintainers are able to access, modify, monitor employee activity and transaction records.
- Items (services or products), employees, discounts, roles, service charges, tax charges, payment records, orders, reservations should all be manageable by business owners and super admins.
- In the case of discounts, you should be able to set time limits on discounts, have them be applicable to one item or a batch order.
- Historical records of payments should be preserved indefinitely.
- System must be capable of logging item(product and or service) variations.
- Changes to tax management should not affect historical records. Different taxes should be applicable to different items or services.
- Orders should have an employee assigned to them.
- Changes to service charge for an item should not affect historical records.
- The system should be aware of the business inventory and reflect inventory change (when something is added to the inventory or when products are sold).
- Dynamic inventory. Upon a customer purchasing a product or a product being used up the change in amount should be reflected in the system. Owners and administrators should be able to stock up on products.
- Processing payments via Stripe.
- Sending SMS messages to customer phones informing about successful/updated/cancelled bookings.
- Accepting third-party gift cards (like Google Pay gift cards, Apple Pay gift cards)

1.3.2 Required business functionality for the employee-end

At the real and daily level, employees will work day in and day out with the system. Quality of life functionality is preferred and basic functionalities must be met.

- Employees should have their own accounts with passwords that they must login to in-order to have access to the system.

- Employee accounts should be capable of managing reservations for customers alongside the reservation time of booking, appointment time, employee, customer, service.

- Employee accounts should only be capable of creating order entries, modifying them before closing, closing or canceling orders, and capable of registering cash, company issued gift cards, third-party gift cards (Apple pay, Google pay) and or credit/debit card payments separately.

- Employee should be able to split the bill among the clients into separate payments.

- Employee should be able to apply a discount to the order wherever applicable.

- The employee should be able to receive a tip.

- Employees should be capable of order refunding.

- Employees should have the ability to view and provide the customer with a detailed check of all the costs, taxes, discounts and service charges present in the transaction.

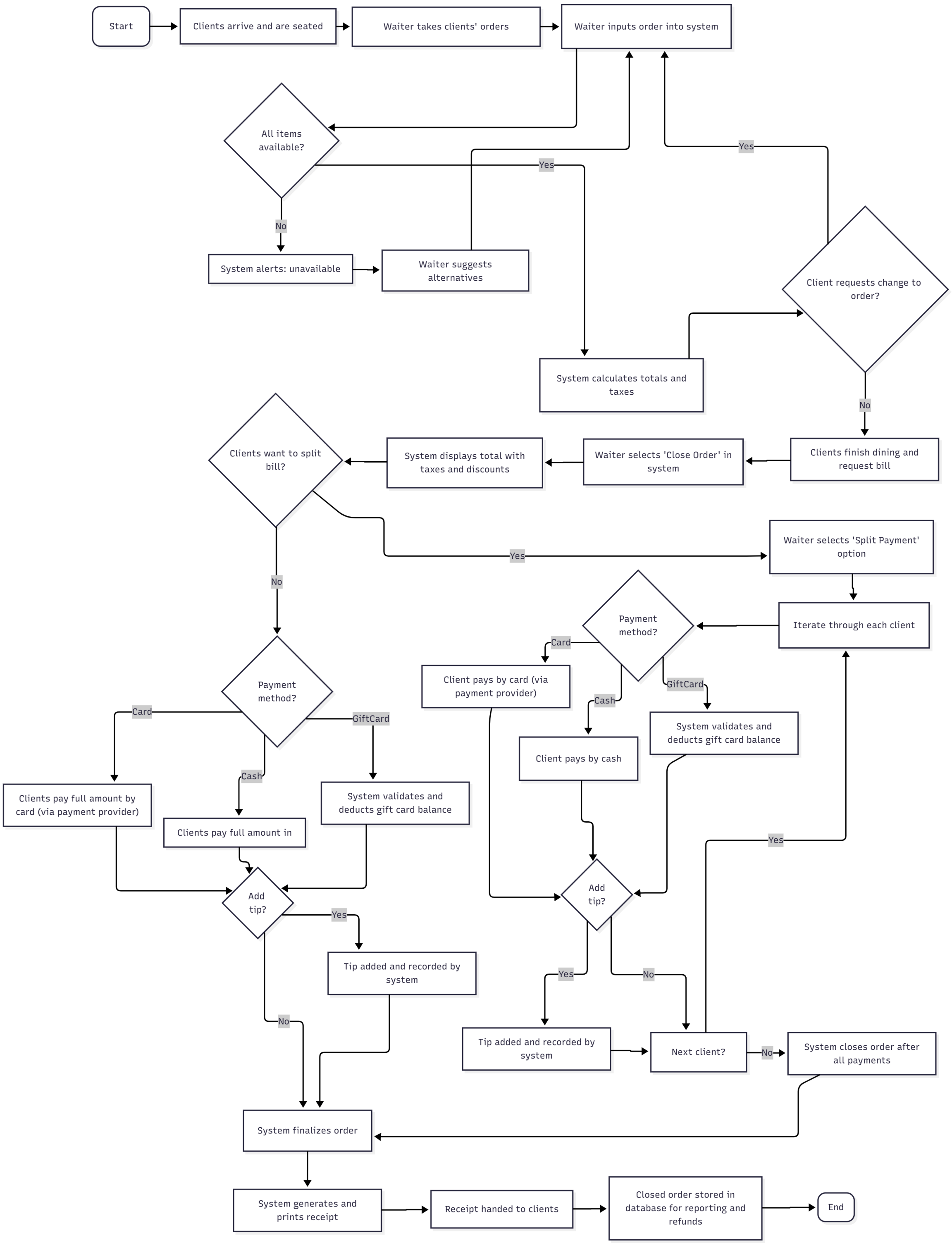
The **Data model** section will have clearer definitions for expected system behavior in the context of data and entity functionality.

2 Business flows

2.1 Restaurant business flow

The following business flow encapsulates a typical business scenario of how a restaurant waiter goes from seating the customers to servicing them and later handling transactions once the customers are ready to pay for the services with different payment methods and how the system records this transaction.

- Clients arrive at the restaurant and are seated.
- Waiter approaches the table and asks for their order.
- Clients provide the order to the waiter.
- Waiter inputs the items into the system.
- System validates availability of menu items and calculates initial totals and taxes.
- One of the clients requests a change to an item in the order.
- Waiter updates the order in the system.
- System revalidates item availability and updates totals.
- When clients finish dining, they request the bill.
- Waiter selects the "Close Order" option in the system.
- System displays total with taxes and any applicable discounts.
- Clients request to split the bill into multiple payments.
- Waiter selects the "Split Payment" option in the system.
- Each client selects their payment method (card, cash, or gift card).
- System processes payments - card payments are verified through an integrated payment provider.
- Clients optionally add a tip.
- After all payments are completed, the waiter closes the order in the system.
- System finalizes the transaction, generates the receipt, and marks the order as closed.
- Receipt is printed and handed to the table.
- Closed order record is preserved in the database for future reporting, analytics, or refunds.

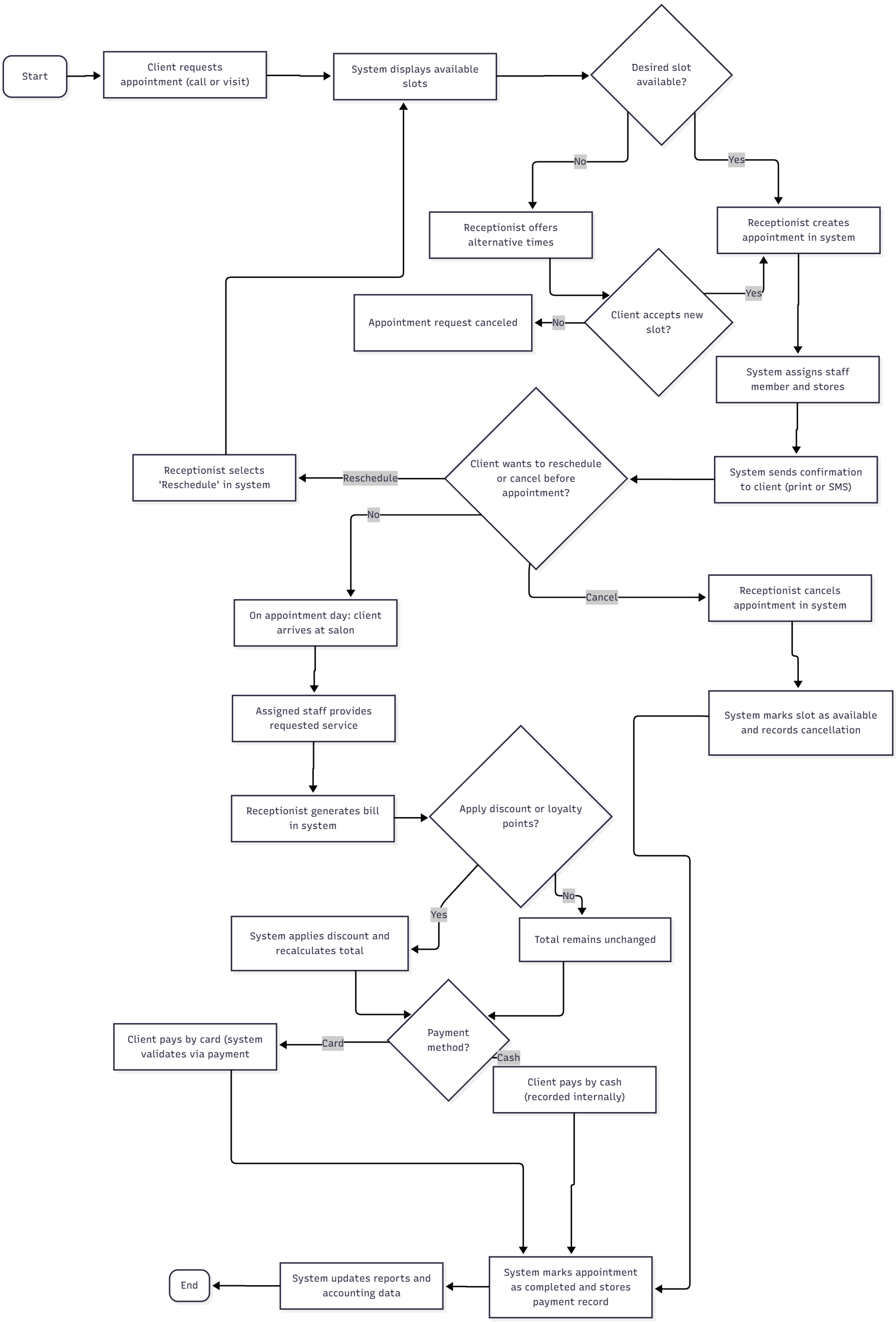

Start
Clients arrive and are seated
Waiter takes clients' orders
Waiter inputs order into system
All items available?
Yes
No
System alerts: unavailable
Waiter suggests alternatives
Yes
Client requests change to order?
System calculates totals and taxes
No
Clients finish dining and request bill
Waiter selects 'Close Order' in system
System displays total with taxes and discounts
Clients want to split bill?
Yes
Waiter selects 'Split Payment' option
No
Iterate through each client
Payment method?
Card
Client pays by card (via payment provider)
Cash
GiftCard
System validates and deducts gift card balance
Client pays by cash
Payment method?
Card
GiftCard
Cash
Clients pay full amount by card (via payment provider)
Clients pay full amount in
System validates and deducts gift card balance
Add tip?
Yes
Tip added and recorded by system
No
Add tip?
Yes
No
Yes
Tip added and recorded by system
Next client?
No
System closes order after all payments
System finalizes order
System generates and prints receipt
Receipt handed to clients
Closed order stored in database for reporting and refunds
End

1 figure. *Restaurant flow chart*

2.2 Beauty salon business flow

The following business scenario now shows how the system behaves in a different context from a restaurant. This time it's the domain of a beauty salon and certain elements differ. A client first places a reservation, either in person or telecommunication, then the employee must find an opening that fits the client's criteria and check the availability. The client is notified of the reservation and is capable of informing an employee to modify the reservation and or cancel it. The client then arrives on the agreed time and an employee is capable of seeing the appointment. Once the service is provided, payment is verified and the service record is stored.

- Client contacts the salon (via phone or in person) to request an appointment.
- Receptionist opens the reservation system and searches for available service slots based on the client's requested service, date, and time.
- System displays available times and staff for the chosen service.
- Client selects a preferred time and staff member.
- Receptionist enters client details and confirms the booking in the system.
- System creates a reservation entry, marks the selected slot as unavailable, and generates a booking confirmation.
- Before the appointment date, the client may call to reschedule or cancel the appointment.
- Receptionist searches the existing booking in the system and updates or removes it accordingly.
- System adjusts staff availability and sends updated confirmation or cancellation notice.
- On the day of the appointment, the assigned staff member views the day's bookings in the system.
- The client arrives at the salon and is checked in by the receptionist.
- The staff member performs the booked service.
- After completion, the receptionist retrieves the corresponding appointment record and generates the bill.
- Client selects a payment method (cash or card).
- System processes payment, optionally applies any discount or loyalty points, and issues a digital or printed receipt.
- System marks the appointment as completed and stores payment details for accounting and future reference.


Start
Client requests appointment (call or visit)
System displays available slots
Desired slot available?
No
Yes
Receptionist offers alternative times
Receptionist creates appointment in system
Client accepts new slot?
Yes
No
Appointment request canceled
System assigns staff member and stores
System sends confirmation to client (print or SMS)
Client wants to reschedule or cancel before appointment?
Reschedule
Receptionist selects 'Reschedule' in system
No
Cancel
Receptionist cancels appointment in system
On appointment day: client arrives at salon
System marks slot as available and records cancellation
Assigned staff provides requested service
Receptionist generates bill in system
Apply discount or loyalty points?
Yes
No
System applies discount and recalculates total
Total remains unchanged
Payment method?
Card
Cash
Client pays by card (system validates via payment
Client pays by cash (recorded internally)
System marks appointment as completed and stores payment record
System updates reports and accounting data
End

2 figure. *Beauty salon flow chart*

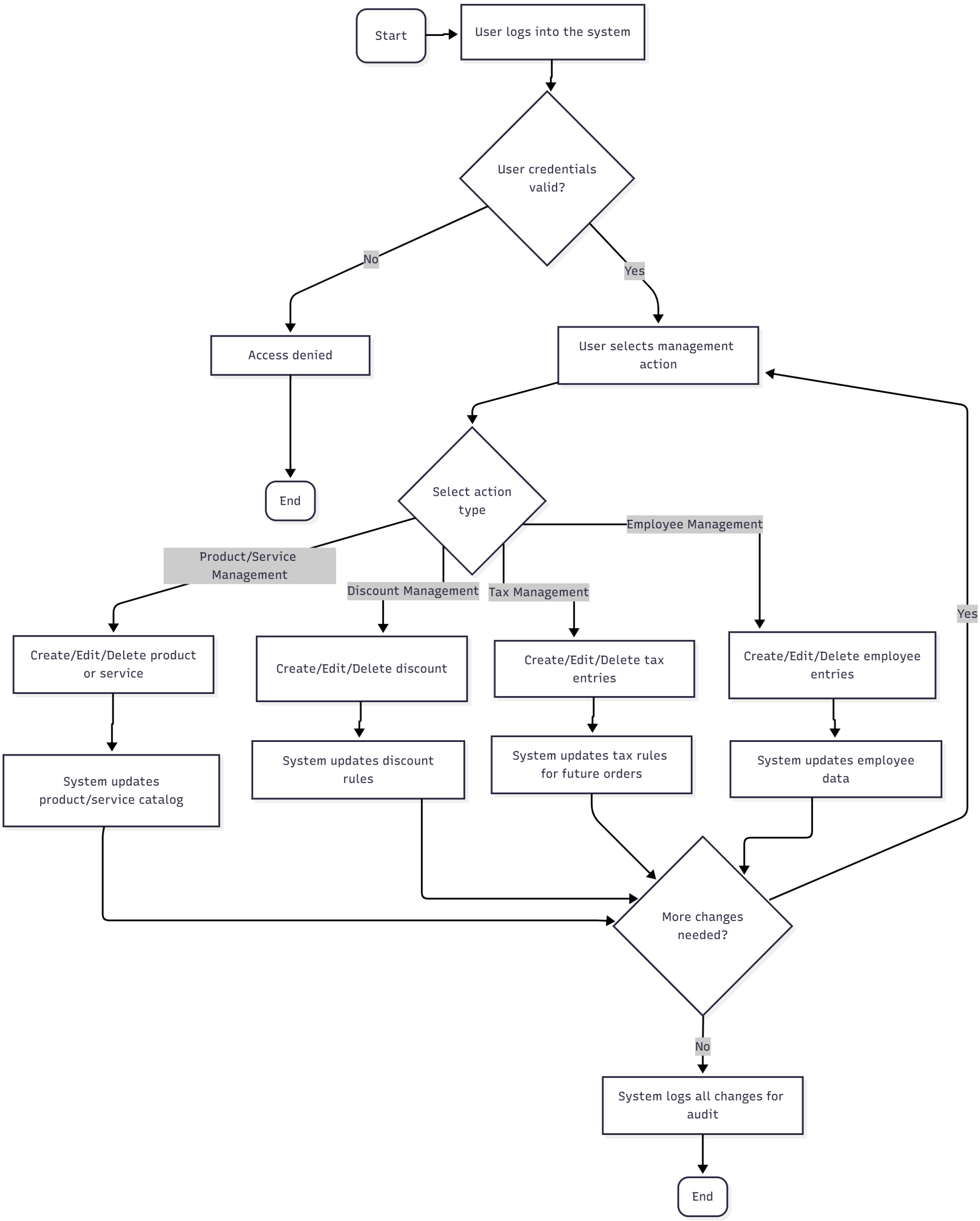
2.3 System management business flow

The system must be manageable and approachable from a maintainers / owners point of view. The business flow below describes the core and fundamental changes that such users may enforce and or view in such a system. It starts with verifying the user and their subscribed plans, once that is successful access to the business management panel is granted and changes to employee staff, tax code, discounts, products, services, inventory can all take place. Alongside the ability to view the transaction records and the ability to see who is accountable for each service record.

- Authorized staff logs into the system using their credentials.
- The system verifies the user's permissions.
- Once authenticated, the staff accesses the management panel.
- The staff selects a management area: employee accounts, taxes, products/services, or discounts.
- **Employee Management Panel:**
  - Staff creates a new employee account, edits an existing account, or deactivates an employee.
  - The system validates the input, updates the employee records, and logs the action for auditing.
- **Tax Management Panel:**
  - Staff creates a new tax configuration, edits an existing one, or deletes an outdated tax.
  - The system updates future transactions while preserving historical records.
  - The action is logged for auditing purposes.
- **Product/Service Management Panel:**
  - Staff adds new products or services, edits details of existing items, or deletes outdated products/services.
  - Staff can also define variations, like "latte with oat milk" or "spa massage 60min/90min."
  - The system validates updates and applies changes only to future orders.
  - All actions are logged automatically.
- **Discount Management Panel:**

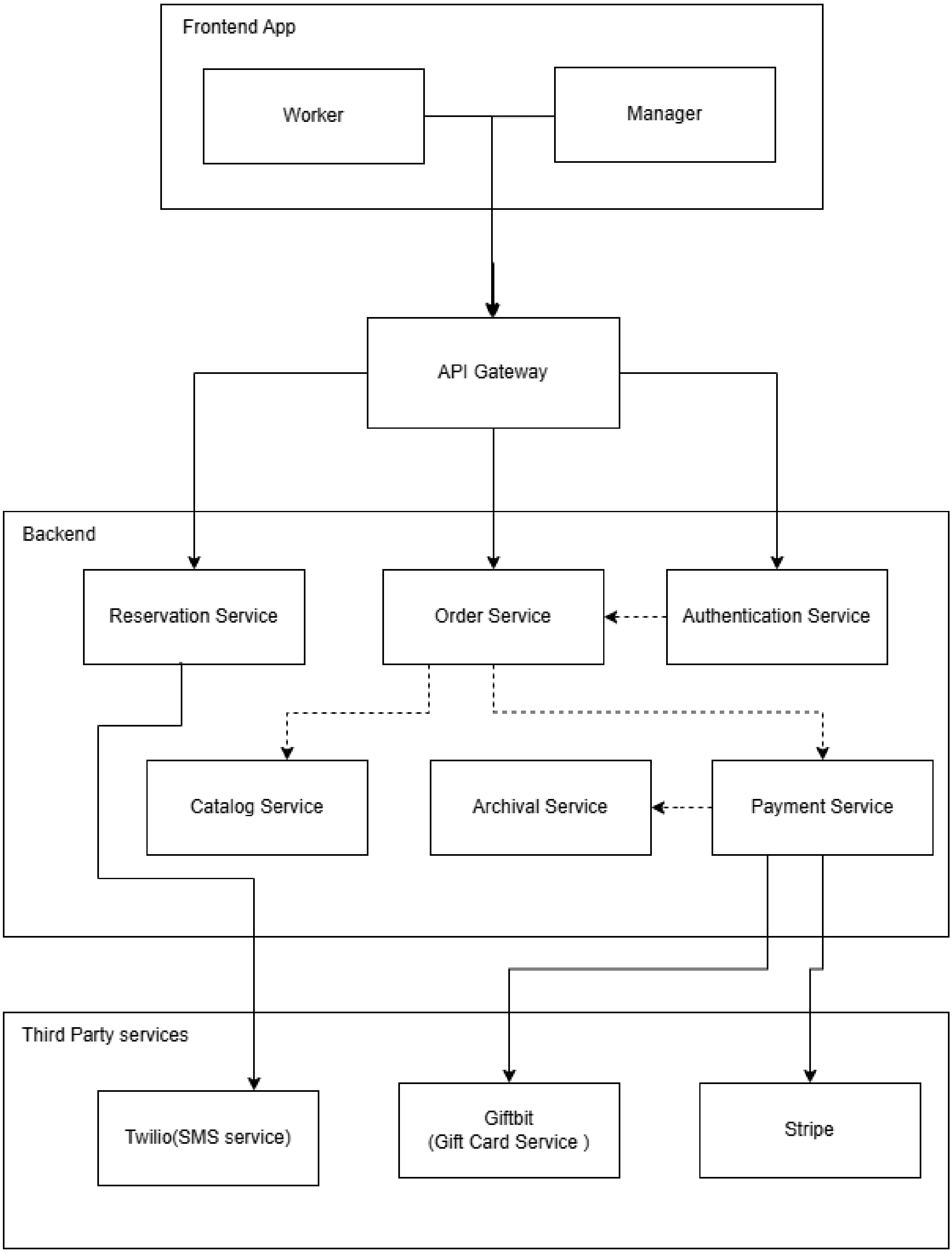
  - Staff creates, modifies, or deletes discounts, which can apply to individual products or entire orders, and can have time-based restrictions.
  - The system validates the rules, updates the records, and logs the action.

- After completing all updates, the system confirms that changes were successfully applied.


Start
User logs into the system
User credentials valid?
No
Yes
Access denied
End
User selects management action
Select action type
Product/Service Management
Discount Management
Tax Management
Employee Management
Create/Edit/Delete product or service
Create/Edit/Delete discount
Create/Edit/Delete tax entries
Create/Edit/Delete employee entries
System updates product/service catalog
System updates discount rules
System updates tax rules for future orders
System updates employee data
More changes needed?
Yes
No
System logs all changes for audit
End

3 figure. *System management flowchart*

3 High level architecture



4 figure. *System architecture*

- Solid line means external call (client to API Gateway to service or third-party service)
- Dashed line means internal service call (service to service)

This 4 figure. package diagram illustrates an architecture centered around an API Gateway, which serves as the single entry point for the Frontend App.

The Gateway is responsible for routing requests to specialized backend services like the Reservation Service, Order Service, and Authentication Service. The Order Service forms the core of the business logic, demonstrating dependencies on other specialized services, it relies on the Authentication Service for user identity and permissions checks and requires information from the Catalog Service for product details.

Furthermore, the Order Service initiates interaction with the Payment Service via a dashed line, suggesting an asynchronous or event-driven communication pattern—a common design for handling financial transactions and catalog updates in a decoupled microservice environment.

Third-party services such as Twilio, Giftbit and Stripe(Payment service) are connected to some of the Backend services. This structure ensures a highly scalable, maintainable, and loosely coupled backend system.

3.1 Frontend Responsibilities

- Login: The frontend displays login/registration forms, sends credentials to the Backend's API, and securely manages the received authorization token.
- Order Creation: It fetches item data, displays a visual menu for user selections, and sends the final order details to the server via `POST /orders` requests.
- Reservation Creation: The frontend provides an interface for users to select date, time, and party size, and then sends this data to the Backend via a `POST /reservations` request.
- Service Creation: For administrative users, it displays configuration forms to input new service details and submits this configuration data to the Backend.
- Tag Addition: It provides an interface for applying tags to products and submits these modifications to the appropriate Backend API endpoint.
- Account (Bill) Splitting: The frontend retrieves order details and presents a visual interface for users to allocate items to different payers before submitting the split payment request to the Backend (the so-called splitting of the bill).
- Time Zone Localization: It detects the user's local time zone and converts UTC date/time values into the user's local time for display purposes.

- Payment Interface: This component displays the final calculated amount, integrates payment method forms, and submits the necessary transaction data to the Backend's `/payments` API.

3.2 Third-party services

Certain services and actions within the system require third-party vendors.

Payment integration

Stripe for payment processing. This is an especially important service as we will be using it for our payment service. Most credit-card and or debit-card checkouts via online banks will be done with specialized devices used for scanning cards. For entering information by hand into the front-end on the employee side you should consider using Stripe web-hooks to insure safe online transactions.

SMS integration

Twilio for SMS communications, whenever a client has a reservation made in their name.

Third-party giftcard integration (Optional)

For integration with Google-pay and Apple-pay gift cards one of the viable third-party vendors would be Giftbit

3.3 Backend Responsibilities

- Financial Transactions: The backend executes all critical calculations, including taxes, discounts, and service charges, and ensures data integrity during all payment processing operations.

- State Management: It tracks and updates the operational status of all entities, such as changing an order's status from 'New' to 'Preparing' or a table's status to 'Occupied'.

- Authentication: This function validates user login attempts, manages password encryption, and generates secure tokens upon a successful login.

- Authorization: On every API request, the backend checks if the authenticated user has the necessary permissions and roles to perform the requested operation.

- Role Assignment: It manages the creation, deletion, and assignment of user roles and permissions within the database.

- Service Addition: The backend is responsible for storing and managing the server-side logic and configuration for new services, items, taxes, or discounts.

- Data Sorting and Filtering: It processes query parameters like `sortBy` and `filterByName` to efficiently retrieve, sort, and filter large datasets from the database before sending them to the Frontend.

- Archiving: The backend implements the logic to move important data into an archive storage for long-term retention .

4 Wire-frames

This section contains GUI mockups, wire-frames, for the system's frontend.

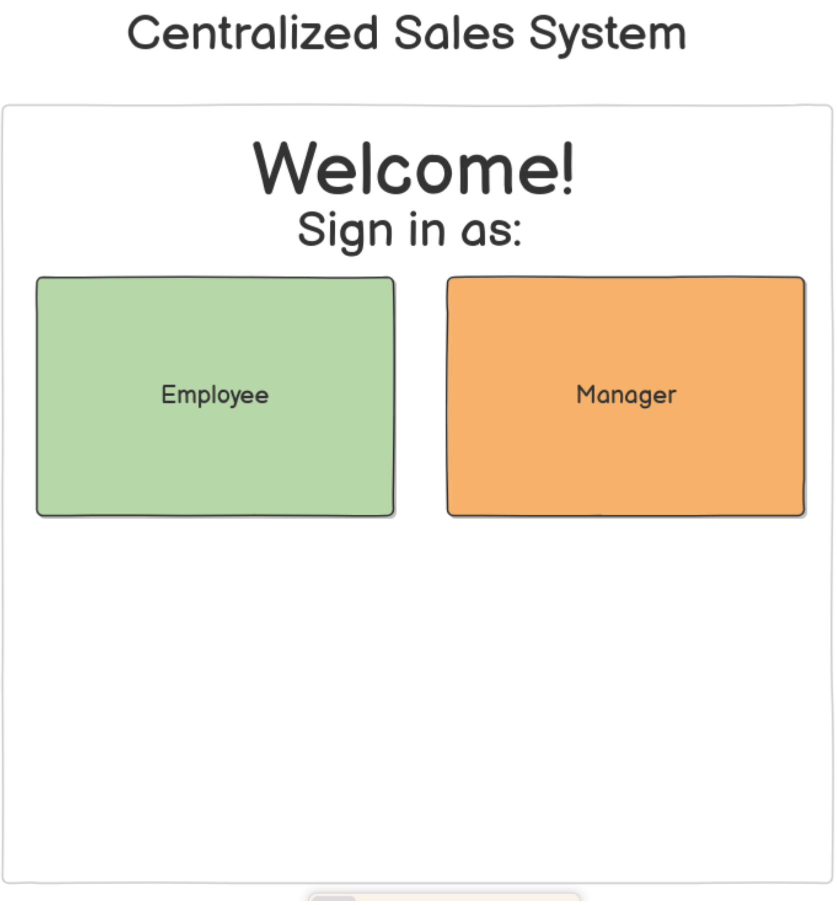
You can interact with the low-fidelity mockups by opening the "cps-wire-frames.pdf" file.

CAUTION - GENERAL

The wire-frames are meant to serve as visual aids for the GUI part of the system. If at any point the visuals contradict internal logic or are inconsistent with other more technical parts of the documentation, they may be disregarded and or appended to properly fit the business domain.

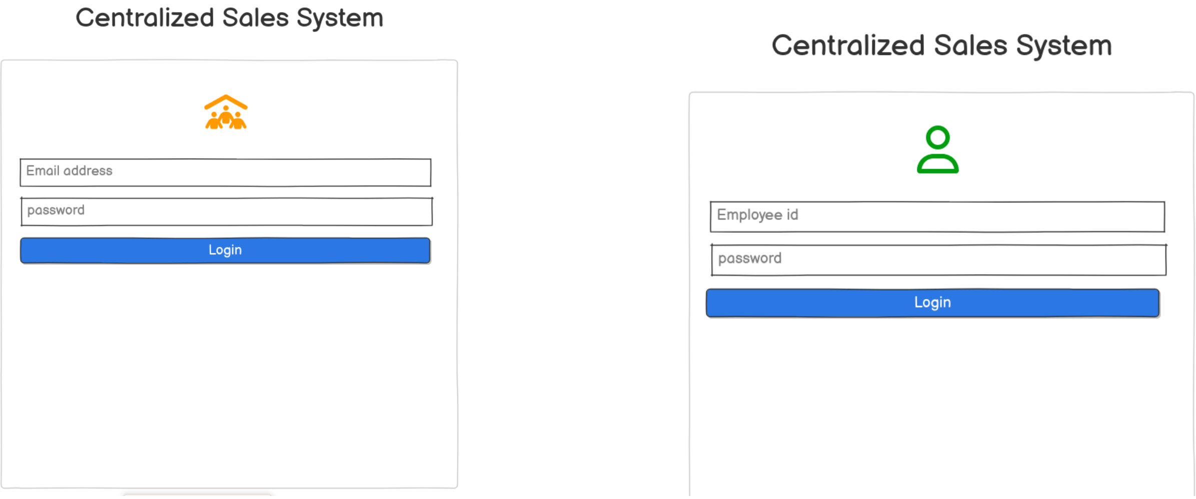
4.1 Shared interfaces

First, managers, system maintainers and establishment employees all login via the same web-portal. (**5 figure.**)



5 figure. *Login by user group portal*

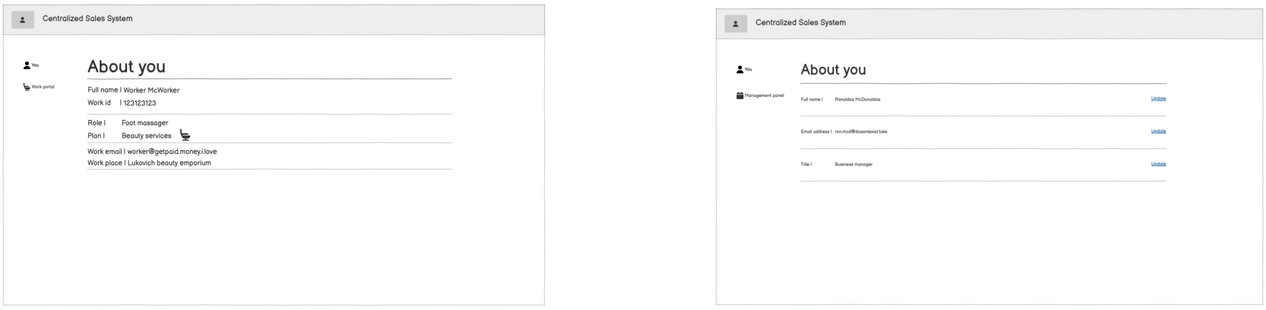
Upon the user picking which group they belong to, they are taken to the next step of the logging process - verification and user authorization. While the business / admin account needs to input an email address and a password, the employee account must provide an employee ID in-place of an email address. (**6 figure.**)



(a) *Login Manager/Super Admin(IT)* **(b)** *Login business employee*

6 figure. *Manager/Super Admin login vs. Employee login*

The final similarity that we will point out is that both types of users have an ”About” section. The content in them varies between the users - managers and system admins have their full name, email address and title (not a role mind you, having a title sounds more *pristine*). The employee account, meanwhile contains their full name, employee ID, role in the company, plan to which their employer has subscribed them to, their work place name and work place email address. (**7 figure.**)



Full name I Worker McWorker
Work id I 123123123
Role I Foot massager
Plan I Beauty services
Work email I worker@getpaid.money.i.love
Work place I Lukovich beauty emporium

Full name I Ronaldas McDonaldas
Email address I ron.mcd@doesntexist.fake
Title I Business manager

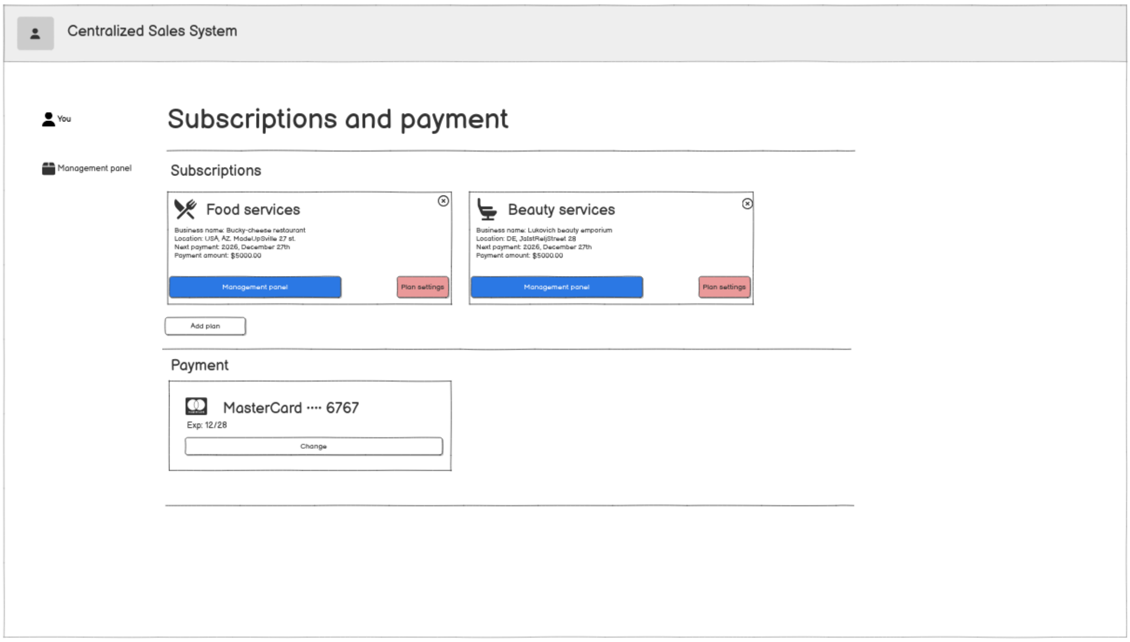
(a) *Employee ”About” section* **(b)** *Manager ”About” section*

7 figure. *Manager/Admin ”About” vs. Employee ”About”*

4.2 Manager/Super Admin management panel

The following characteristics are unique to the manager / administrator roles.

First, we have the ”Subscriptions and payment” sections that allow the administrator to add their new businesses and buy plans for them in either food catering or beauty catering, from here they will also be accessing those specific panels. (**8 figure.**)



Centralized Sales System

Subscriptions and payment

Subscriptions

Food services

Business name: Bucky-cheese restaurant
Location: USA, AZ. MadeUpSville 27 st.
Next payment: 2026, December 27th
Payment amount: $5000.00

Beauty services

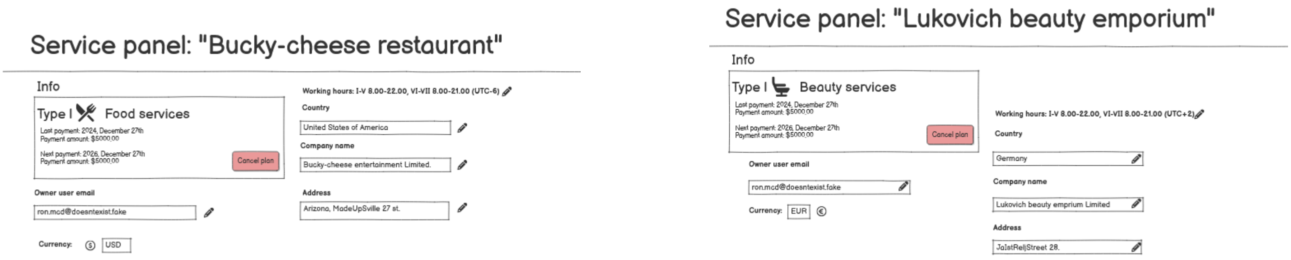
Business name: Lukovich beauty emporium
Location: DE, JalstReljStreet 28
Next payment: 2026, December 27th
Payment amount: $5000.00

Payment

MasterCard ···· 6767
Exp: 12/28

8 figure. *Services management panel*

The next wire-frames in question are the panels in the case of ”Food services” panel and ”Beauty services” panel.
The panel is primarily made up of two parts - ”Info” and ”Manage”.
”Info” has all the business related information - the name of the establishment, country, payment due dates, owner and their email, currency settings, address, that doubles as the billing address, and also working hours alongside the time-zone. (**9 figure.**)



Service panel: "Bucky-cheese restaurant"

Info

Type I Food services

Last payment: 2024, December 27th
Payment amount: $5000.00

Next payment: 2026, December 27th
Payment amount: $5000.00

Owner user email: ron.mcd@doesntexist.fake

Currency: USD

Working hours: I-V 8.00-22.00, VI-VII 8.00-21.00 (UTC-6)

Country: United States of America

Company name: Bucky-cheese entertainment Limited.

Address: Arizona, MadeUpSville 27 st.

(a) *Food service ”Info” section*

Service panel: "Lukovich beauty emporium"

Info

Type I Beauty services

Last payment: 2024, December 27th
Payment amount: $5000.00

Next payment: 2026, December 27th
Payment amount: $5000.00

Owner user email: ron.mcd@doesntexist.fake

Currency: EUR

Working hours: I-V 8.00-22.00, VI-VII 8.00-21.00 (UTC+2)

Country: Germany

Company name: Lukovich beauty emprium Limited

Address: JalstReljStreet 28.

(b) *Beauty service ”Info” section*

9 figure. *Different service panel info sections*

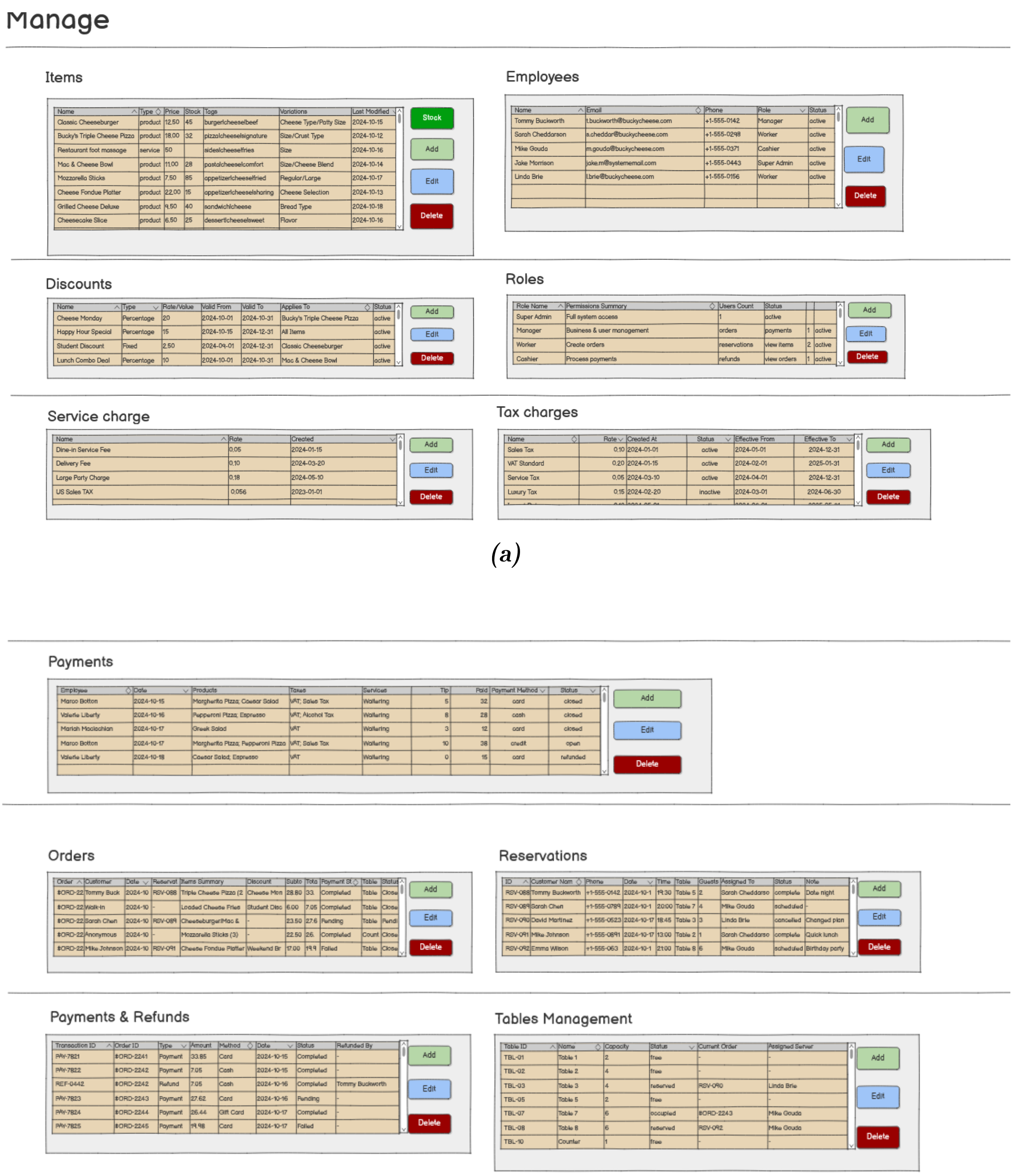
CAUTION - EXTRA!

You may need to make major changes in your implementation compared to the wire-frames below. Although the wire-frame is mostly accurate, some tables may have incorrect or missing attributes. Be vigilant and double-check when implementing this part in your user interface.

The next two wire-frames are by far the most important and will vary based off certain implementation choices so it will be kept brief. The main idea is that there are management panels that work like kinds of ”Tables” that managers or admins can directly operate with when it comes to adding new ones, updating preexisting tables and deleting entries. It is highly advised you open the PDF file containing the wire-frames. It so massive that we will only explore one of the examples (the pdf file has both).

The ”Items” table should be unique in that the amount of items changes dynamically based off orders and that restocked by the manager.
Once again, table names and fields may vary based off implementation choices. (**10 figure.**)

NEXT PAGE FOR IMAGES



Manage

Items

Name	Type	Price	Stock	Tags	Variations	Last Modified
Classic Cheeseburger	product	12.50	45	burgerlcheeselbeef	Cheese Type/Patty Size	2024-10-15
Bucky's Triple Cheese Pizza	product	18.00	32	pizzalcheeselsignature	Size/Crust Type	2024-10-12
Restaurant foot massage	service	50		sideslcheeselfries	Size	2024-10-16
Mac & Cheese Bowl	product	11.00	28	pastalcheeselcomfort	Size/Cheese Blend	2024-10-14
Mozzarella Sticks	product	7.50	85	appetizerlcheeselfried	Regular/Large	2024-10-17
Cheese Fondue Platter	product	22.00	15	appetizerlcheeselsharing	Cheese Selection	2024-10-13
Grilled Cheese Deluxe	product	9.50	40	sandwichlcheese	Bread Type	2024-10-18
Cheesecake Slice	product	6.50	25	dessertlcheeselsweet	Flavor	2024-10-16

Employees

Name	Email	Phone	Role	Status
Tommy Buckworth	t.buckworth@buckycheese.com	+1-555-0142	Manager	active
Sarah Cheddarson	s.cheddar@buckycheese.com	+1-555-0298	Worker	active
Mike Gouda	m.gouda@buckycheese.com	+1-555-0371	Cashier	active
Jake Morrison	jake.m@systememail.com	+1-555-0443	Super Admin	active
Linda Brie	l.brie@buckycheese.com	+1-555-0156	Worker	active

Discounts

Name	Type	Rate/Value	Valid From	Valid To	Applies To	Status
Cheese Monday	Percentage	20	2024-10-01	2024-10-31	Bucky's Triple Cheese Pizza	active
Happy Hour Special	Percentage	15	2024-10-15	2024-12-31	All Items	active
Student Discount	Fixed	2.50	2024-09-01	2024-12-31	Classic Cheeseburger	active
Lunch Combo Deal	Percentage	10	2024-10-01	2024-10-31	Mac & Cheese Bowl	active

Roles

Role Name	Permissions Summary	Users Count	Status		
Super Admin	Full system access	1	active		
Manager	Business & user management	orders	payments	1	active
Worker	Create orders	reservations	view items	2	active
Cashier	Process payments	refunds	view orders	1	active

Service charge

Name	Rate	Created
Dine-in Service Fee	0.05	2024-01-15
Delivery Fee	0.10	2024-03-20
Large Party Charge	0.18	2024-05-10
US Sales TAX	0.056	2023-01-01

Tax charges

Name	Rate	Created At	Status	Effective From	Effective To
Sales Tax	0.10	2024-01-01	active	2024-01-01	2024-12-31
VAT Standard	0.20	2024-01-15	active	2024-02-01	2025-01-31
Service Tax	0.05	2024-03-10	active	2024-04-01	2024-12-31
Luxury Tax	0.15	2024-02-20	inactive	2024-03-01	2024-06-30

(a)

Payments

Employee	Date	Products	Taxes	Services	Tip	Paid	Payment Method	Status
Marco Botton	2024-10-15	Margherita Pizza; Caesar Salad	VAT; Sales Tax	Waitering	5	32	card	closed
Valerie Liberty	2024-10-16	Pepperoni Pizza; Espresso	VAT; Alcohol Tax	Waitering	8	28	cash	closed
Mariah Maclachlan	2024-10-17	Greek Salad	VAT	Waitering	3	12	card	closed
Marco Botton	2024-10-17	Margherita Pizza; Pepperoni Pizza	VAT; Sales Tax	Waitering	10	38	credit	open
Valerie Liberty	2024-10-18	Caesar Salad; Espresso	VAT	Waitering	0	15	card	refunded

Orders

Order	Customer	Date	Reservat	Items Summary	Discount	Subto	Tota	Payment St	Table	Status
#ORD-22	Tommy Buck	2024-10	RSV-088	Triple Cheese Pizza (2	Cheese Mon	28.80	33.	Completed	Table	Close
#ORD-22	Walk-in	2024-10	-	Loaded Cheese Fries	Student Disc	6.00	7.05	Completed	Table	Close
#ORD-22	Sarah Chen	2024-10	RSV-089	CheeseburgerlMac &	-	23.50	27.6	Pending	Table	Pendi
#ORD-22	Anonymous	2024-10	-	Mozzarella Sticks (3)	-	22.50	26.	Completed	Count	Close
#ORD-22	Mike Johnson	2024-10	RSV-091	Cheese Fondue Platter	Weekend Br	17.00	19.9	Failed	Table	Close

Reservations

ID	Customer Nam	Phone	Date	Time	Table	Guests	Assigned To	Status	Note
RSV-088	Tommy Buckworth	+1-555-0142	2024-10-1	19:30	Table 5	2	Sarah Cheddarso	complete	Date night
RSV-089	Sarah Chen	+1-555-0789	2024-10-1	20:00	Table 7	4	Mike Gouda	scheduled	-
RSV-090	David Martinez	+1-555-0523	2024-10-17	18:45	Table 3	3	Linda Brie	cancelled	Changed plan
RSV-091	Mike Johnson	+1-555-0891	2024-10-17	13:00	Table 2	1	Sarah Cheddarso	complete	Quick lunch
RSV-092	Emma Wilson	+1-555-063	2024-10-1	21:00	Table 8	6	Mike Gouda	scheduled	Birthday party

Payments & Refunds

Transaction ID	Order ID	Type	Amount	Method	Date	Status	Refunded By
PAY-7821	#ORD-2241	Payment	33.85	Card	2024-10-15	Completed	-
PAY-7822	#ORD-2242	Payment	7.05	Cash	2024-10-15	Completed	-
REF-0442	#ORD-2242	Refund	7.05	Cash	2024-10-16	Completed	Tommy Buckworth
PAY-7823	#ORD-2243	Payment	27.62	Card	2024-10-16	Pending	-
PAY-7824	#ORD-2244	Payment	26.44	Gift Card	2024-10-17	Completed	-
PAY-7825	#ORD-2245	Payment	19.98	Card	2024-10-17	Failed	-

Tables Management

Table ID	Name	Capacity	Status	Current Order	Assigned Server
TBL-01	Table 1	2	free	-	-
TBL-02	Table 2	4	free	-	-
TBL-03	Table 3	4	reserved	RSV-090	Linda Brie
TBL-05	Table 5	2	free	-	-
TBL-07	Table 7	6	occupied	#ORD-2243	Mike Gouda
TBL-08	Table 8	6	reserved	RSV-092	Mike Gouda
TBL-10	Counter	1	free	-	-

(b)

10 figure. *Management interface - food catering scenario*

4.3 Restaurant employee interface

The following sections contains 3 figures detailing *roughly* how the employee portal interfaces should look like in the food service scenario. As always, make modifications wherever necessary.

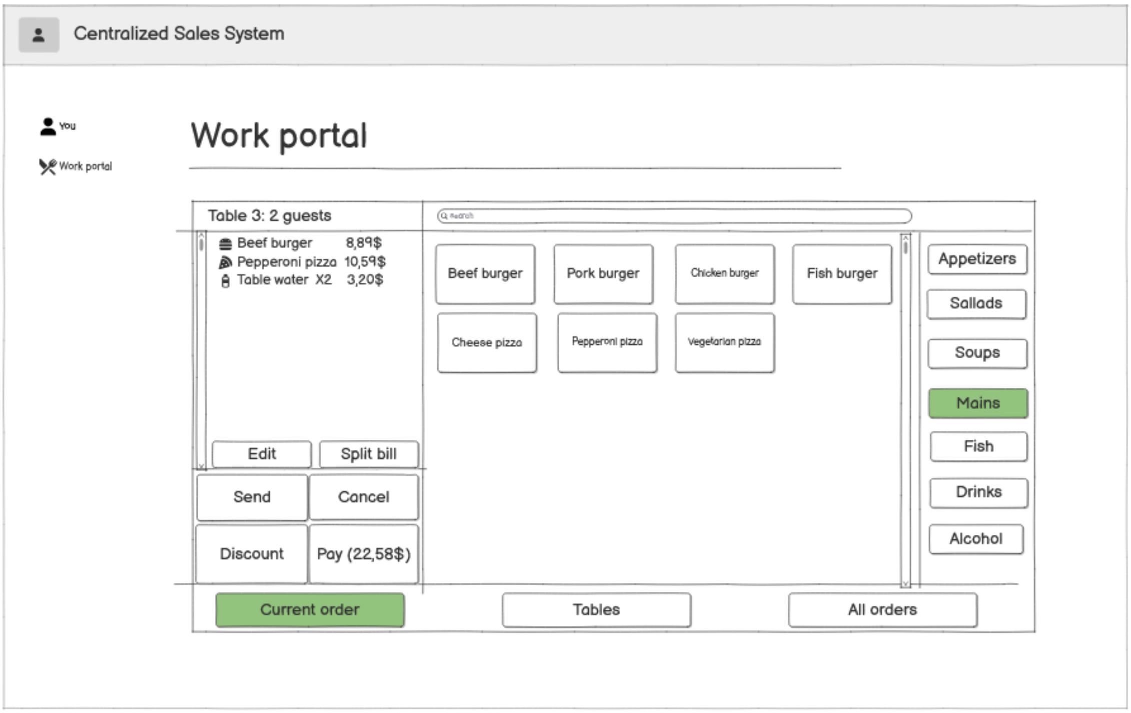
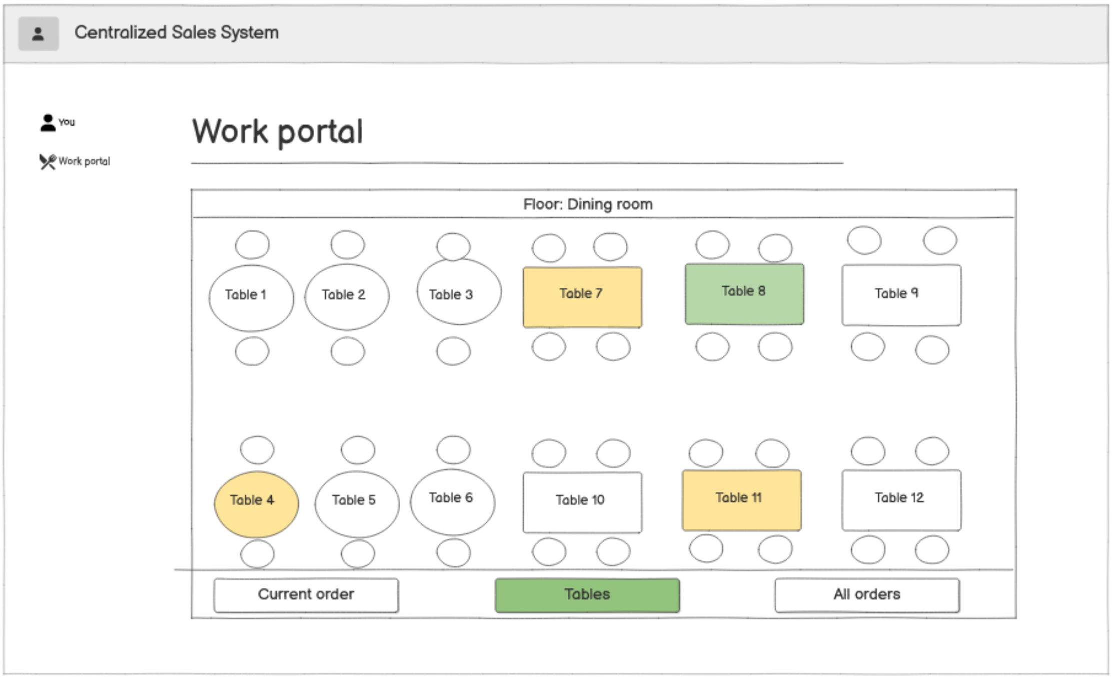


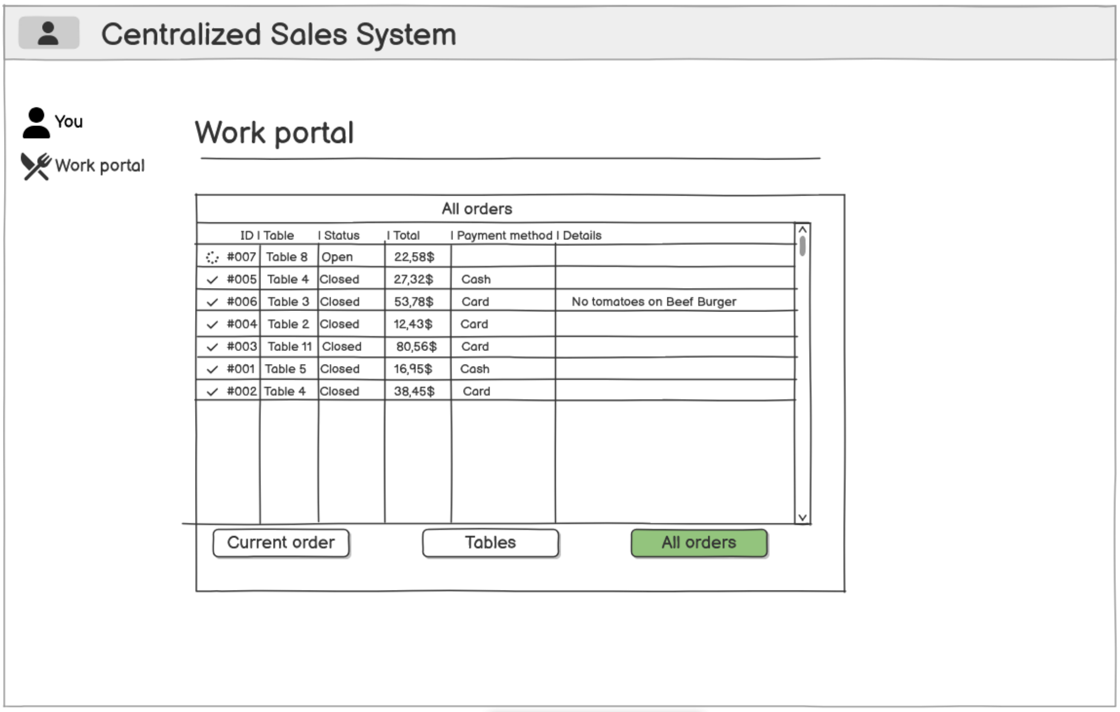
Table 3: 2 guests

Beef burger		8,89$
Pepperoni pizza		10,59$
Table water	X2	3,20$

11 figure. *Single order management*



12 figure. *Table seating management*



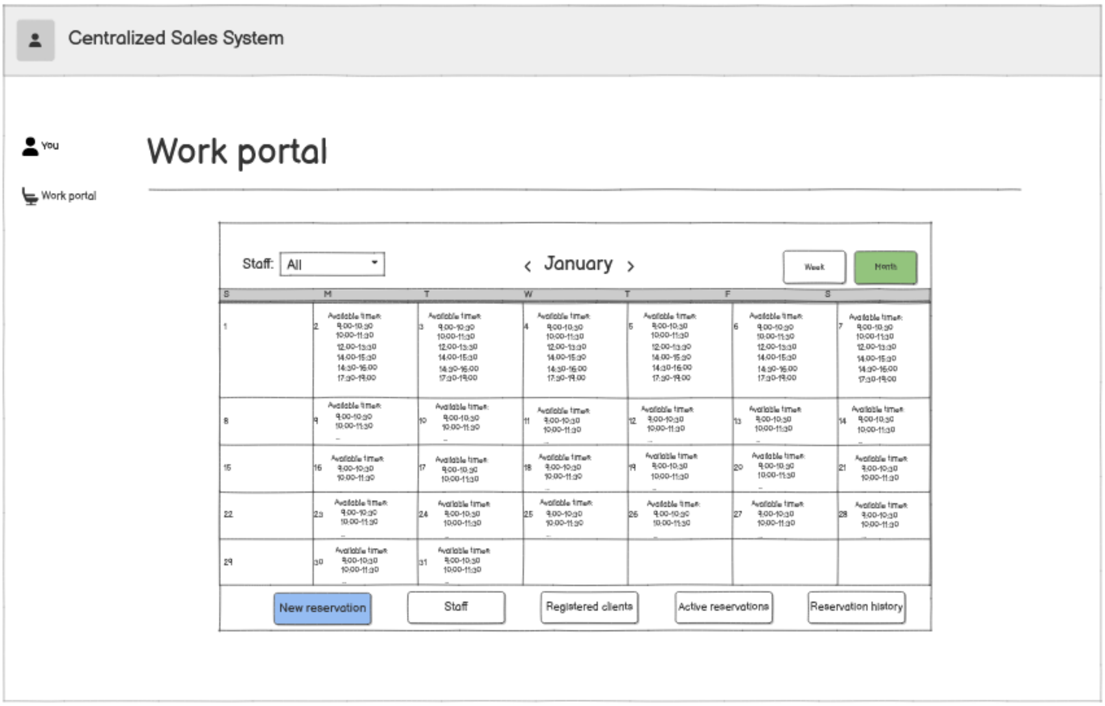
All orders

	ID	Table	Status	Total	Payment method	Details
◌	#007	Table 8	Open	22,58$		
✓	#005	Table 4	Closed	27,32$	Cash	
✓	#006	Table 3	Closed	53,78$	Card	No tomatoes on Beef Burger
✓	#004	Table 2	Closed	12,43$	Card	
✓	#003	Table 11	Closed	80,56$	Card	
✓	#001	Table 5	Closed	16,95$	Cash	
✓	#002	Table 4	Closed	38,45$	Card	

13 figure. *All order management*

4.4 Beauty salon employee interface

The following sections contain 6 figures detailing how *roughly* the employee portal interfaces should look like in the beauty service scenario. As always, make modifications wherever necessary.

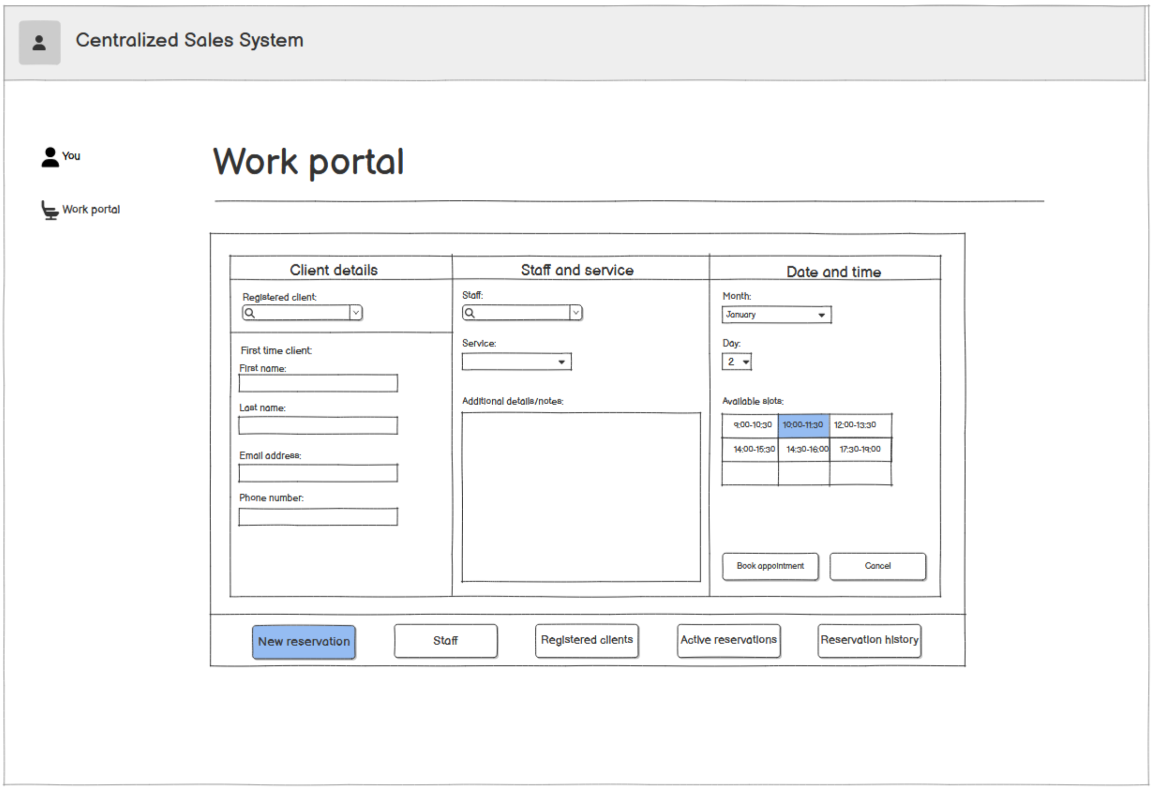


Staff: All

‹ January ›

S	M	T	W	T	F	S
1	2 Available times: 9:00-10:30 10:00-11:30 12:00-13:30 14:00-15:30 14:30-16:00 17:30-19:00	3 Available times: 9:00-10:30 10:00-11:30 12:00-13:30 14:00-15:30 14:30-16:00 17:30-19:00	4 Available times: 9:00-10:30 10:00-11:30 12:00-13:30 14:00-15:30 14:30-16:00 17:30-19:00	5 Available times: 9:00-10:30 10:00-11:30 12:00-13:30 14:00-15:30 14:30-16:00 17:30-19:00	6 Available times: 9:00-10:30 10:00-11:30 12:00-13:30 14:00-15:30 14:30-16:00 17:30-19:00	7 Available times: 9:00-10:30 10:00-11:30 12:00-13:30 14:00-15:30 14:30-16:00 17:30-19:00
8	9 Available times: 9:00-10:30 10:00-11:30 –	10 Available times: 9:00-10:30 10:00-11:30 –	11 Available times: 9:00-10:30 10:00-11:30 –	12 Available times: 9:00-10:30 10:00-11:30 –	13 Available times: 9:00-10:30 10:00-11:30 –	14 Available times: 9:00-10:30 10:00-11:30 –
15	16 Available times: 9:00-10:30 10:00-11:30	17 Available times: 9:00-10:30 10:00-11:30	18 Available times: 9:00-10:30 10:00-11:30 –	19 Available times: 9:00-10:30 10:00-11:30 –	20 Available times: 9:00-10:30 10:00-11:30 –	21 Available times: 9:00-10:30 10:00-11:30
22	23 Available times: 9:00-10:30 10:00-11:30 –	24 Available times: 9:00-10:30 10:00-11:30 –	25 Available times: 9:00-10:30 10:00-11:30 –	26 Available times: 9:00-10:30 10:00-11:30 –	27 Available times: 9:00-10:30 10:00-11:30 –	28 Available times: 9:00-10:30 10:00-11:30 –
29	30 Available times: 9:00-10:30 10:00-11:30 –	31 Available times: 9:00-10:30 10:00-11:30 –				

14 figure. *Reservation schedule management*



Work portal

Client details

Registered client:

First time client:

First name:

Last name:

Email address:

Phone number:

Staff and service

Staff:

Service:

Additional details/notes:

Date and time

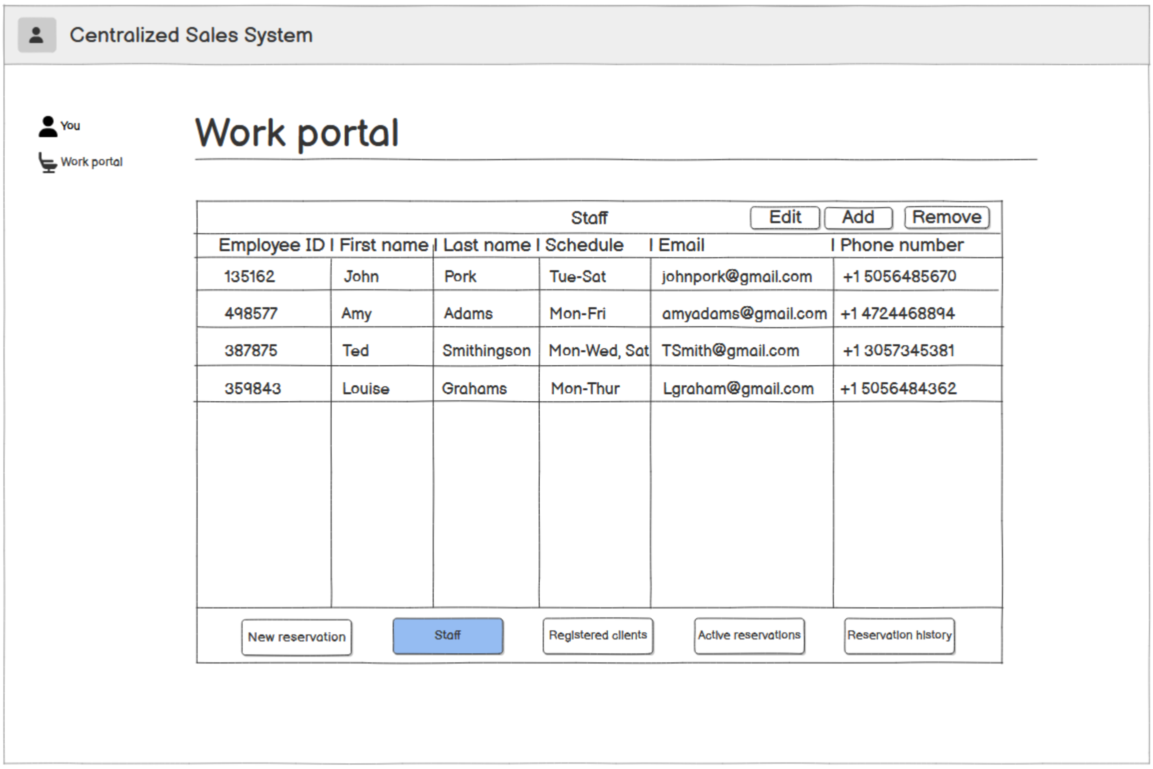
Month: January

Day: 2

Available slots:

9:00-10:30	10:00-11:30	12:00-13:30
14:00-15:30	14:30-16:00	17:30-19:00

15 figure. *Reservation creation*

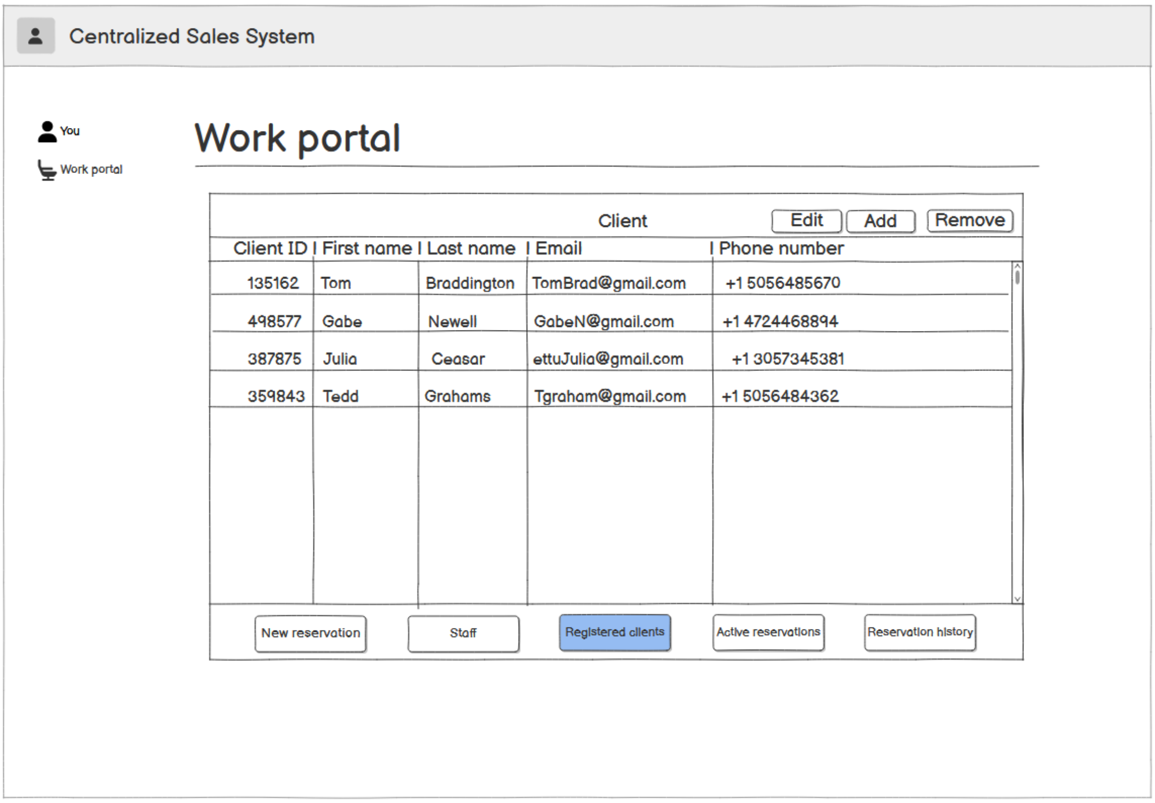


Work portal

Staff

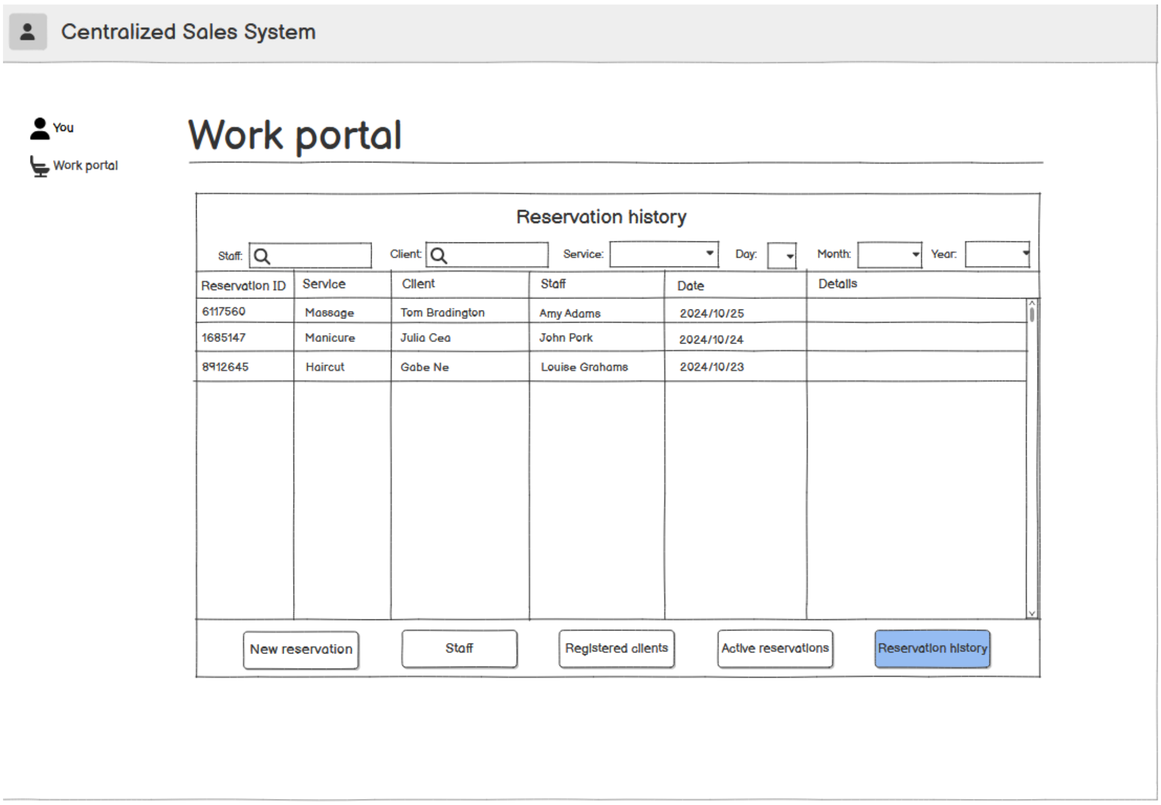
Employee ID	First name	Last name	Schedule	Email	Phone number
135162	John	Pork	Tue-Sat	johnpork@gmail.com	+1 5056485670
498577	Amy	Adams	Mon-Fri	amyadams@gmail.com	+1 4724468894
387875	Ted	Smithingson	Mon-Wed, Sat	TSmith@gmail.com	+1 3057345381
359843	Louise	Grahams	Mon-Thur	Lgraham@gmail.com	+1 5056484362

16 figure. *On duty staff listing and management*



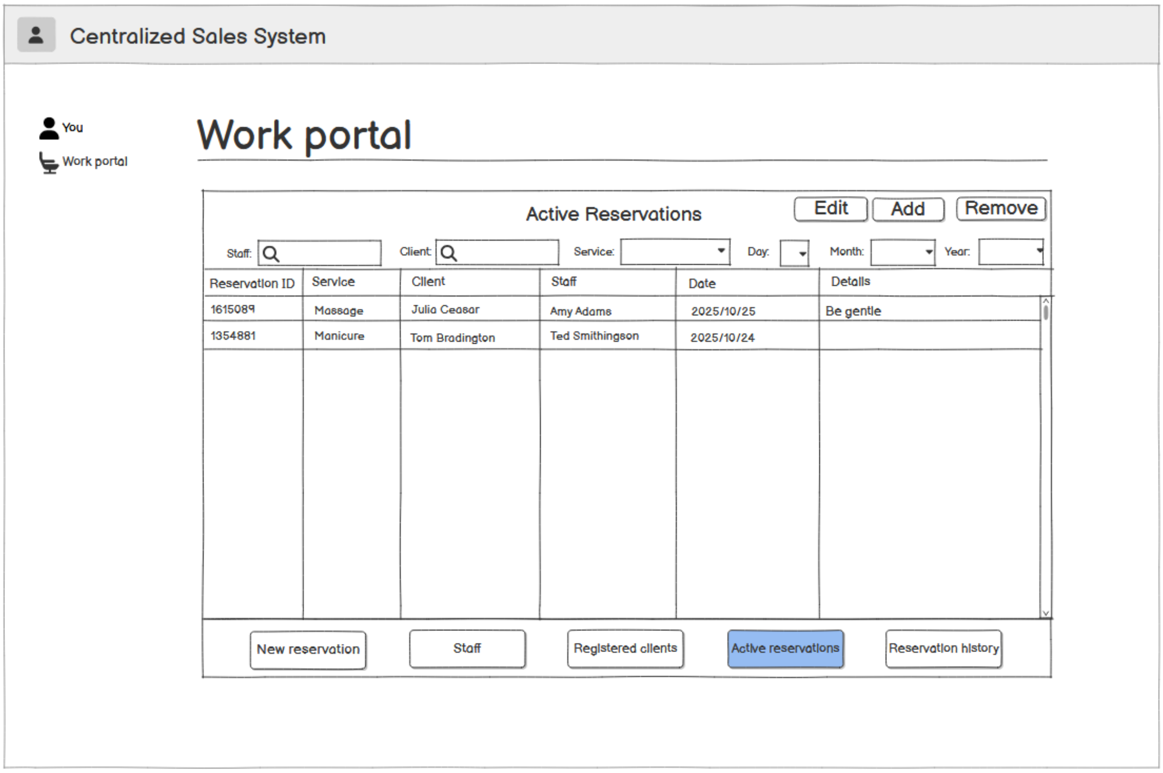
Client ID	First name	Last name	Email	Phone number
135162	Tom	Braddington	TomBrad@gmail.com	+1 5056485670
498577	Gabe	Newell	GabeN@gmail.com	+1 4724468894
387875	Julia	Ceasar	ettuJulia@gmail.com	+1 3057345381
359843	Tedd	Grahams	Tgraham@gmail.com	+1 5056484362

17 figure. *Registered clients*



Reservation ID	Service	Client	Staff	Date	Details
6117560	Massage	Tom Bradington	Amy Adams	2024/10/25	
1685147	Manicure	Julia Cea	John Pork	2024/10/24	
8912645	Haircut	Gabe Ne	Louise Grahams	2024/10/23	

18 figure. *Reservation history*



Reservation ID	Service	Client	Staff	Date	Details
1615089	Massage	Julia Ceasar	Amy Adams	2025/10/25	Be gentle
1354881	Manicure	Tom Bradington	Ted Smithingson	2025/10/24	

19 figure. *Active reservations*

5 Data model

5.1 Rules

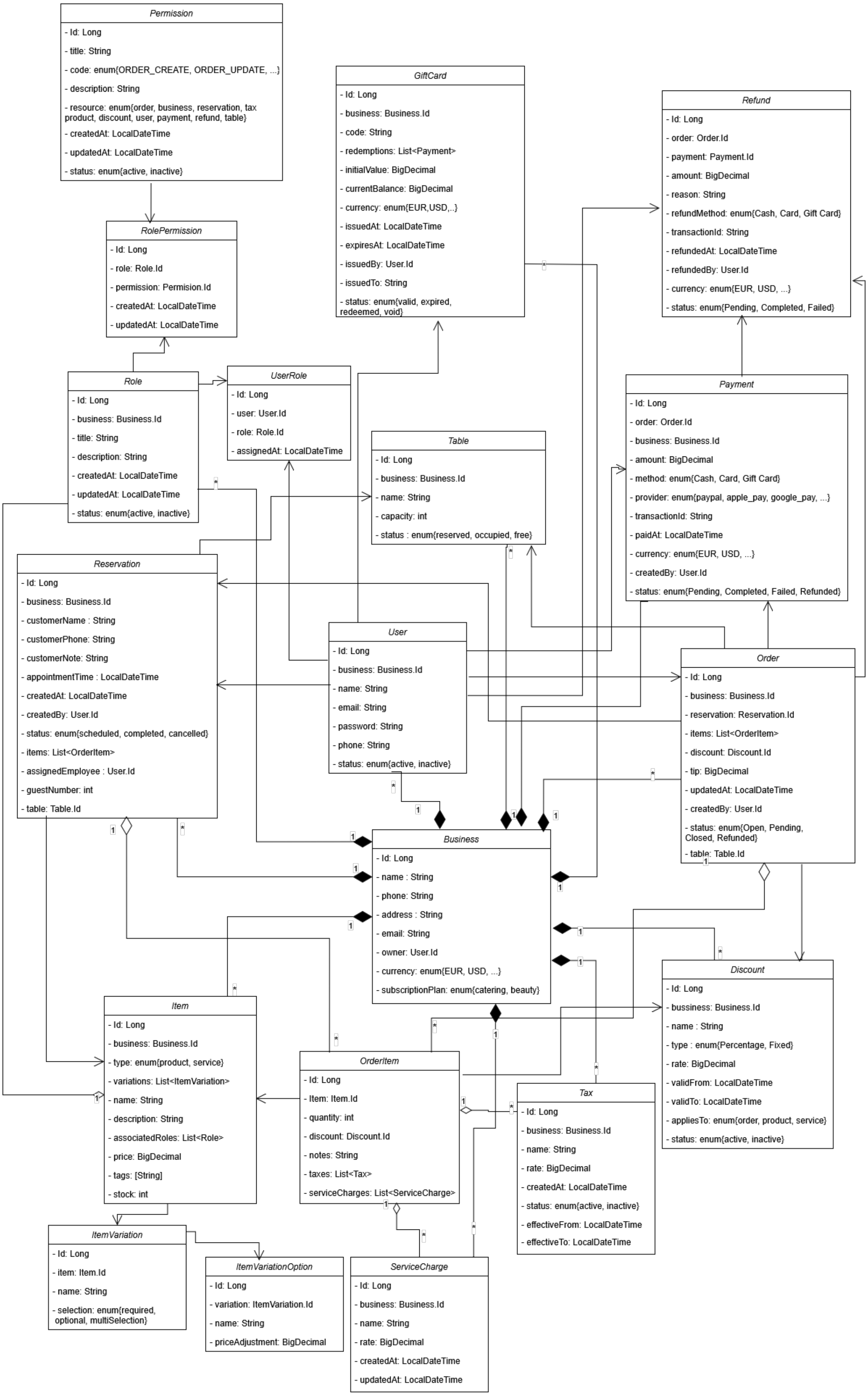
- Pending and closed orders cannot be modified or removed.
- When tax or discount rate is changed it must not change in historical records (past orders).
- Timestamp values are always stored in UTC/GMT/Epoch time.
- Decimal values should be saved in data types that have a high degree of precision. In data model in this document BigDecimal (Java special class) datatype is used. If you are not using Java, then an equivalent data type must be used.
- Calculatable fields such as the total order price are not stored in the database.

5.2 Visualization of the data model

The data model seen in 20 figure. illustrates the main data structures and their relationships within the Centralized Sales System. It defines how entities such as Users, Reservations, Orders, Items, Payments, Discounts, Taxes, Roles and other entities interact through associations, aggregations and compositions to support key business operations like booking and ordering.

Relationship explanation:

- A **White diamond (◇-)** indicates aggregation — one class "owns" a collection of the other, but the child can also exist independently.
- A **black diamond (◆-)** indicates composition — the child cannot exist without the parent.
- The **plain arrow (→)** represent dependencies between modules.
- The numbers near the associations (1 and *) represent multiplicity:
  - 1 means "exactly one."
  - * means "many (zero or more)."


Permission
- Id: Long
- title: String
- code: enum{ORDER_CREATE, ORDER_UPDATE, ...}
- description: String
- resource: enum{order, business, reservation, tax product, discount, user, payment, refund, table}
- createdAt: LocalDateTime
- updatedAt: LocalDateTime
- status: enum{active, inactive}
RolePermission
- Id: Long
- role: Role.Id
- permission: Permision.Id
- createdAt: LocalDateTime
- updatedAt: LocalDateTime
GiftCard
- Id: Long
- business: Business.Id
- code: String
- redemptions: List<Payment>
- initialValue: BigDecimal
- currentBalance: BigDecimal
- currency: enum{EUR,USD,..}
- issuedAt: LocalDateTime
- expiresAt: LocalDateTime
- issuedBy: User.Id
- issuedTo: String
- status: enum{valid, expired, redeemed, void}
Refund
- Id: Long
- order: Order.Id
- payment: Payment.Id
- amount: BigDecimal
- reason: String
- refundMethod: enum{Cash, Card, Gift Card}
- transactionId: String
- refundedAt: LocalDateTime
- refundedBy: User.Id
- currency: enum{EUR, USD, ...}
- status: enum{Pending, Completed, Failed}
Role
- Id: Long
- business: Business.Id
- title: String
- description: String
- createdAt: LocalDateTime
- updatedAt: LocalDateTime
- status: enum{active, inactive}
UserRole
- Id: Long
- user: User.Id
- role: Role.Id
- assignedAt: LocalDateTime
Table
- Id: Long
- business: Business.Id
- name: String
- capacity: int
- status : enum{reserved, occupied, free}
Payment
- Id: Long
- order: Order.Id
- business: Business.Id
- amount: BigDecimal
- method: enum{Cash, Card, Gift Card}
- provider: enum{paypal, apple_pay, google_pay, ...}
- transactionId: String
- paidAt: LocalDateTime
- currency: enum{EUR, USD, ...}
- createdBy: User.Id
- status: enum{Pending, Completed, Failed, Refunded}
Reservation
- Id: Long
- business: Business.Id
- customerName : String
- customerPhone: String
- customerNote: String
- appointmentTime : LocalDateTime
- createdAt: LocalDateTime
- createdBy: User.Id
- status: enum{scheduled, completed, cancelled}
- items: List<OrderItem>
- assignedEmployee : User.Id
- guestNumber: int
- table: Table.Id
User
- Id: Long
- business: Business.Id
- name: String
- email: String
- password: String
- phone: String
- status: enum{active, inactive}
Order
- Id: Long
- business: Business.Id
- reservation: Reservation.Id
- items: List<OrderItem>
- discount: Discount.Id
- tip: BigDecimal
- updatedAt: LocalDateTime
- createdBy: User.Id
- status: enum{Open, Pending, Closed, Refunded}
- table: Table.Id
Business
- Id: Long
- name : String
- phone: String
- address : String
- email: String
- owner: User.Id
- currency: enum{EUR, USD, ...}
- subscriptionPlan: enum{catering, beauty}
Discount
- Id: Long
- bussiness: Business.Id
- name : String
- type : enum{Percentage, Fixed}
- rate: BigDecimal
- validFrom: LocalDateTime
- validTo: LocalDateTime
- appliesTo: enum{order, product, service}
- status: enum{active, inactive}
Item
- Id: Long
- business: Business.Id
- type: enum{product, service}
- variations: List<ItemVariation>
- name: String
- description: String
- associatedRoles: List<Role>
- price: BigDecimal
- tags: [String]
- stock: int
OrderItem
- Id: Long
- Item: Item.Id
- quantity: int
- discount: Discount.Id
- notes: String
- taxes: List<Tax>
- serviceCharges: List<ServiceCharge>
Tax
- Id: Long
- business: Business.Id
- name: String
- rate: BigDecimal
- createdAt: LocalDateTime
- status: enum{active, inactive}
- effectiveFrom: LocalDateTime
- effectiveTo: LocalDateTime
ItemVariation
- Id: Long
- item: Item.Id
- name: String
- selection: enum{required, optional, multiSelection}
ItemVariationOption
- Id: Long
- variation: ItemVariation.Id
- name: String
- priceAdjustment: BigDecimal
ServiceCharge
- Id: Long
- business: Business.Id
- name: String
- rate: BigDecimal
- createdAt: LocalDateTime
- updatedAt: LocalDateTime

20 figure. *Systems data model*

5.3 Main data pieces

Data pieces are according to the data model seen in 20 figure.

- **Business** - a company that uses our application. It has a name, subscription plan (catering or beauty), contact information (phone, email, address), currency on which the business operates and business owner information. Business entity is linked to orders, reservations, users, discounts, taxes, tables, items, service charges, giftcards and roles. These vary by businesses.
  Required fields: id, name, phone, email, address, owner, currency, subscription plan.
  Optional fields: none.

- **User** - A person who uses our system. It contains user credential information (email, password, phone) and status (active, inactive).
  Status explanation:

  - **Active** - The user account exists and currently has access to the system. This status is for employees and owners who are currently working there.
  - **Inactive** - The user exists, but has been deactivated and cannot access the system. These could be the status for employees who worked at some business at some time, but not anymore.

  Required fields: id, business, name, email, password, phone, status.
  Optional fields: none.

- **Item** - it is something that a business sells. It can be food items or services. It has type (product or service), name, description, price tags and variations. Food items can have stock, while services have associated roles. AssociatedRoles should be used only if item is a service, as it shows the roles that are able to do the service.
  Required fields: id, business, type, name, price, variations.
  Optional fields: description, associatedRoles, tags, stock.

- **ItemVariation** - An item variation is a configurable attribute of a product or service like size, flavour or duration. It is linked with an item, has a name and a selection that tells whether variation is required, optional or it has to be multiple selection.
  Required fields: id, item, name, selection.
  Optional fields: none.

- **ItemVariationOption** - an item variation option tells what are the options for the item variation. For example, if item variation is ”size”, then the options could be ”small”, ”medium” or ”large”. It has name, is linked with item variation and has price adjustment which tells, how much this variation of the item differ from the vanilla version of it.
  Required fields: id, variation, name, priceAdjustment.
  Optional fields: none.

- **Order** - it is the main sale record. It has a list of order items, status (open, pending, closed, refunded) and timestamp when order was last updated and the user who did it. In addition, an order can have discounts and tips. It is linked to order items, payments and refunds. It is also linked to Table entity that is required for catering businesses. An order can be linked with a reservation since when after a completed reservation a customer has to pay for services or products.
  Status explanation:
  - **Open** - The order was placed by the customer and created by the employee. It can be modified.
  - **Pending** - The order is waiting for the payment. The order **cannot** be modified or removed.
  - **Closed** - The payment was completed. It **cannot** be modified or removed.
  - **Refunded** - A closed order was refunded.

  Required fields: id, business, items, updatedAt, createdBy, status.
  Optional fields: reservation, tip, discount, table.
- **OrderItem** - it is a part of an order. Shows what products were bought, their quantity, applicable discount, service charges and taxes and additional notes if needed.
  Required fields: id, product, quantity, tax.
  Optional fields: notes, discount, serviceCharges.
- **Reservation** - it is the main booking service where you can either book a beauty service or reserve a table in the restaurant. It has customer contact information (name and phone number), additional notes, a list of selected services or products, appointment time, status (scheduled, completed, cancelled) and timestamp when the reservation was created and the user who created it. For catering businesses a table reservation is needed (guestNumber and table). When reserving a table it is unnecessary to select products. For beauty services there has to be an assigned employee that does the service for the customer and selected services.
  Status explanation:
  - **Scheduled** - it means that the reservation is upcoming.
  - **Completed** - it means that the customer arrived and reservation has taken place.
  - **Cancelled** - the customer cancelled a scheduled reservation and it has been called off.

  Required fields: id, business, customer name, customer phone, appointmentTime, createdAt, createdBy, status.
  Optional fields: customerNote, items, assignedEmployee, guestNumber, Table.

- **Tax** - contains a tax rate. It has name, rate, status (active or inactive), and when it is effective (from-to).
  Required fields: id, business, name, rate, createdAt, status, effectiveFrom.
  Optional fields: effectiveTo.

- **ServiceCharge** - contains charge rate for service. It has name, rate and when it was created.
  Required fields: id, business, name, rate, createdAt, updatedAt.
  Optional fields: none.

- **Discount** - a price reduction for order items or whole orders. It has a name, rate that can be a percentage or a fixed amount, status, it can be time-limited and it varies by businesses.
  Required fields: id, business, name, type, rate, validFrom, appliesTo, status.
  Optional fields: validTo.

- **Payment** - contains payment information. It is linked to order, business, has amount, payment method (cash, card, gift card), provider, transaction id, timestamp when the payment has occured, currency, an employee who opened the payment process and a status of the payment (pending, completed, failed, refunded).
  Status explanation:

  - **Pending** - The payment is in progress and awaiting confirmation or completion.
  - **Completed** - The payment was processed and received successfully.
  - **Failed** - The payment was attempted but did not go through successfully.
  - **Refunded** - The payment was returned to the customer after being completed.

  Required fields: id, order, business, amount, method, provider, transactionId, paidAt, currency, createdBy, status.
  Optional fields: none.

- **Table** - represents a table, a place where customers can eat or reserve a place for eating. It is required only for catering businesses. It has name, capacity and status (reserved, occupied, free).
  Status explanation:

  - **Reserved** - The table has been booked in advance for a specific time or customer.
  - **Occupied** - The table is currently in use by guests.
  - **Free** - The table is available and not currently reserved or occupied.

  Required fields: id, business, name, capacity, status.
  Optional fields: none.

- **GiftCard** - it represents a prepaid stored-value card that customers can use to pay for services and goods. It is linked to business, it has a unique code, redemptions where payments are tracked, it stores initial value, current balance, currency, it has timestamps when it was issued and when it expires, it can have customer's name (issuedTo) and status (valid, expired, redeemed, void).
  Status explanation:
  - **Valid** - The gift card was issued and can be redeemed.
  - **Expired** - The gift card was issued, but not redeemed until the expiration date. It cannot be used anymore.
  - **Redeemed** - The gift card was issued and redeemed before the expiration date.
  - **Void** - A gift card can be void if it was issued in error, canceled by the issuer, or found to be fraudulent.

  Required fields: id, business, code, redemptions, initialValue, currentBalance, currency, issuedAt, expiresAt, issuedBy, status.
  Optional fields: issuedTo.

- **Refund** - contains refund information. It has order and payment information, an amount that is going to be refunded, reason for a refund, refund method (cash, card, gift card), transaction id, currency, status (pending, completed, failed), a user who did the refund and a timestamp when it was done.
  Required fields: id, order, payment, amount, refundMethod, transactionId, refundedAt, refundedBy, currency, status.
  Optional fields: reason.

- **Role** - represents user roles in the system. It has business, title, description status (active, inactive) and timestamps when it was created and last updated. Roles can vary by business.
  Required fields: id, business, title, createdAt, updatedAt, status.
  Optional fields: description.

- **Permission** - contains all permissions that can be assigned to roles. It has title, a permission identifier called code, description, resource that shows to which group it belongs (order, business, reservation, tax, product, discount, user, payment, refund, table, gift-card, service_charge), status (active, inactive) and timestamps when it was created and last updated.
  Possible codes for permissions:
  - **Business management**:
    - **BUSINESS_VIEW** - View business profile and settings.
    - **BUSINESS_UPDATE** - Edit business details (name, address, email, currency, etc.).

    - **BUSINESS_DELETE** - Delete or deactivate business.
    - **BUSINESS_SUBSCRIPTION_MANAGE** - Change subscription plan or billing settings.

- **User and staff management**:
    - **USER_VIEW** - View users and staff profiles.
    - **USER_CREATE** - Add new staff or users.
    - **USER_UPDATE** - Edit user details.
    - **USER_DELETE** - Deactivate or remove a user.
    - **ROLE_VIEW** - View roles and assigned permissions.
    - **ROLE_CREATE** - Create a new role.
    - **ROLE_UPDATE** - Edit an existing role.
    - **ROLE_DELETE** - Delete or deactivate a role.
    - **PERMISSION_VIEW** - View permissions list.
    - **PERMISSION_ASSIGN** - Assign permissions to roles.
    - **USER_ROLE_ASSIGN** - Assign roles to users.

- **Product and service management**:
    - **ITEM_VIEW** - View products or services.
    - **ITEM_CREATE** - Create a new product or service.
    - **ITEM_UPDATE** - Edit product or service details.
    - **ITEM_DELETE** - Deactivate or remove product or service.
    - **TAX_VIEW** - View taxes.
    - **TAX_MANAGE** - Create, update, or deactivate tax rules.
    - **SERVICE_CHARGE_VIEW** - View service charges.
    - **SERVICE_CHARGE_MANAGE** - Create, update, or deactivate service charges.
    - **DISCOUNT_VIEW** - View available discounts.
    - **DISCOUNT_CREATE** - Create new discount.
    - **DISCOUNT_UPDATE** - Modify discount.
    - **DISCOUNT_DELETE** - Deactivate or remove discount.
    - **DISCOUNT_APPLY** - Apply discount to an order item or an order.

- **Orders and order item management**:
    - **ORDER_VIEW** - View order list and details.
    - **ORDER_CREATE** - Create new order.
    - **ORDER_UPDATE** - Edit existing order.
    - **ORDER_DELETE** - Cancel or delete order.
    - **ORDER_ITEM_ADD** - Add item to order.

        - **ORDER_ITEM_UPDATE** - Modify order item.
        - **ORDER_ITEM_REMOVE** - Remove order item.
        - **ORDER_CLOSE** - Mark order as closed or paid.
    - **Payment, refund and gift card management**:
        - **PAYMENT_VIEW** - View payments and their status.
        - **PAYMENT_CREATE** - Record or process a new payment.
        - **PAYMENT_UPDATE** - Modify payment record (e.g., add transaction ID).
        - **PAYMENT_DELETE** - Void or cancel payment.
        - **PAYMENT_REFUND** - Issue refund for a payment.
        - **REFUND_VIEW** - View refund history.
        - **REFUND_CREATE** - Create refund record.
        - **REFUND_DELETE** - Cancel or delete refund record.
        - **GIFTCARD_ISSUE** - Issue a gift card.
        - **GIFTCARD_REDEEM** - Mark gift card as redeemed.
        - **GIFTCARD_VOID** - void or cancel a gift card.
    - **Reservation and table management**:
        - **RESERVATION_VIEW** - View reservations or appointments.
        - **RESERVATION_CREATE** - Add new reservation or appointment.
        - **RESERVATION_UPDATE** - Modify existing reservation.
        - **RESERVATION_CANCEL** - Cancel reservation.
        - **TABLE_VIEW** - View tables or seating arrangements.
        - **TABLE_MANAGE** - Create, update, or remove table configurations.

    Required fields: id, title, code, resource, createdAt, updatedAt, status.
    Optional fields: description.

- **RolePermission** - represents what roles have what permissions. It has role, permission and and timestamps when it was created and last updated.
  Required fields: id, role, permission, createdAt, updatedAt.
  Optional fields: none.

- **UserRole** - represents which users have which roles. It has role, user and a timestamp when the role was assigned.
  Required fields: id, user, role, assignedAt.
  Optional fields: none.

Relationships:

- One **business** can have many **users, reservations, orders, items, discounts, taxes, service charges, giftcards, tables** and **roles**.

- One **Order** and **reservation** can have many **order items**. An **order item** cannot exist independently.

- An **Order** depends on **discount** for pricing rules, on **table** for table information, on **user** for user information, **reservation** for reservation details.

- An **order item** depends on **discount**, **taxes** and **service charges** for pricing rules and taxing rates as well as on **item** for catalog details.

- A **Reservation** depends on **order items** for ordered services, on **table** for table information and on **User** for user information.

- **Payment** depends on **order** for order information.

- **Item** depends on **ItemVariation** for item variation details. **ItemVariation** depends on **ItemVariationOption** for item variation options information.

- **Refund** depends on **order** and **payment** for order and payment information.

- **RolePermission** depends on **role** and **permission** for role and permission information.

- **UserRole** depends on **users** and **roles** for user and role information.