

# ChillBill Lab 2

## Team

**ChillBill**

[\[github repo\]](#)

## Assigned design from

**Super App**

[\[design document\]](#)

## Evaluation

### What we did not implement

What	Reasoning
Authentication via email & password	The provided database schema lacked authentication fields, such as encrypted password
Email/SMS Notifications	While the design mentions inviting users and issuing receipts, we did not implement an SMTP service to send actual emails. <b>(It was marked as optional)</b>
Service Charge Configuration	The package diagram mentioned a Service Charge entity, but the business flows described it alongside Tips. We implemented Tips (variable amounts) and Taxes (fixed percentages), which cover the majority of use cases, without a separate "Service Charge" auto-calculation logic.
Employee Availability/Schedule management, communication between bar and kitchen.	The document mentions (in flow descriptions) employees checking staff availability in walk-in customer flow, managers checking employee schedules when making a reservation, bar and kitchen Order communication. We did not find this reasonable to implement, because this does not seem as an essential asset that would be part of a POS.

## What we did differently

What	Reasoning
Added <code>googleId</code> field to <code>Employee</code> database table	The design lacked it, but mentioned authentication via Google OAuth
Added <code>Seat</code> table for reservations	The design document had <code>tableOrArea</code> string as a field of <code>Reservation</code> , which we found to make less sense, as a hurried employee can enter the wrong seat, which would lead to confusion. In our implementation Managers and Owners can add/remove seats and employees can select from the available
Unified Admin & Employee Tables	The design suggested a separate <code>AdminAccount</code> table for Super Admins. We merged this into the <code>Employee</code> table (where <code>businessId</code> is null for Super Admins). This simplifies the authentication logic and database schema while maintaining role-based access control.
Order Confirmation and Freeze of an Order	<p>The document mentioned that the Employee would go over the details of the Order with the Customer and confirm/finalize the Order - making it ready for payment and freezing Order details.</p> <p>We thought that Order finalization/freezing is not necessary as most often in real life situations (like in bars), Employees just confirm the Order details by looking through them and then go straight to payment.</p>

## What was missing

What	Reasoning
Business database table is missing type of business	The UI (in document) differed based on type of business (hair dresser / cafe). We implemented an all-rounder instead, which leads to the user having to always choose if they're adding a service or a product.
Permission altering system	While the design document mentioned (most of the time) who was able to do what, it would've probably been nice for the Owner to be able alter permissions.
No UI for customers	In a real POS you would want the customer to have some display (e.g. to enter card information or to create an order). The document mentions customers do not directly interact with the system though.

## Final rating

**9/10**, all around a good document, but was missing some minor details.