# Concierge

## TEAM 6

MYKOLA, JOSH, NATHAN, RAINCLOUDED, LEEROY

# THE PROJECT

# What is Concierge?

- Concierge is a simple, elegant, and easy to use hospitality management and operational facilitation platform. We provide your esteemed guests with an opulent point of contact with hotel staff.

- We provide a bespoke service which permits staff to offer prestigious hospitality without ever interacting with a guest face-to-face.

# What is Concierge?

## Problem statement

- Hotels still rely on outdated systems to serve guests and manage business operations.

# What is Concierge?

## Where we come in

- Provide an intuitive platform that allows guests to conveniently access hotel benefits and empowers staff to streamline operations.

# What Benefits Do I Have The Privilege to Receive?

| Guest | Staff |
|---|---|
| • View hotel amenities<br>• Request hotel services<br>     • Food service<br>     • Spa & Massage<br>     • Wakeup call | • Update hotel information<br>• Oversee cases of hotel incidents<br>• Fulfill guest submitted requests<br>• Create and manage guest and staff accounts |

# And Now, The Demo

# THE VISUAL DESIGN

# Design Kickoff

### Visual Design Team
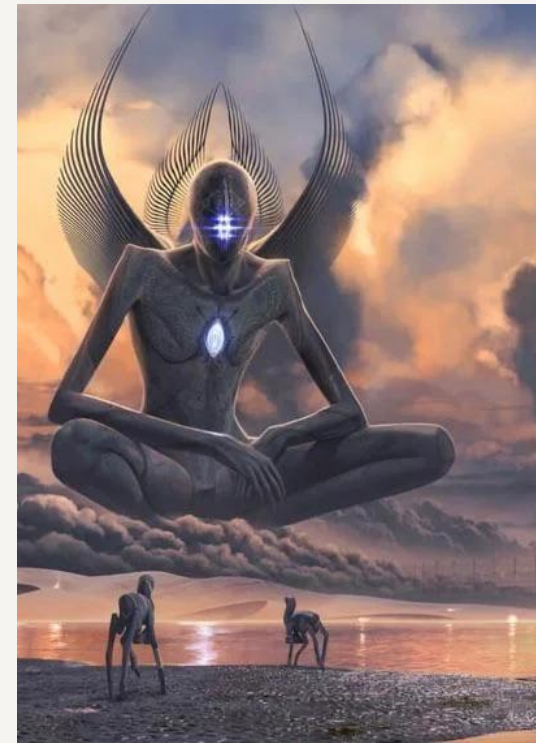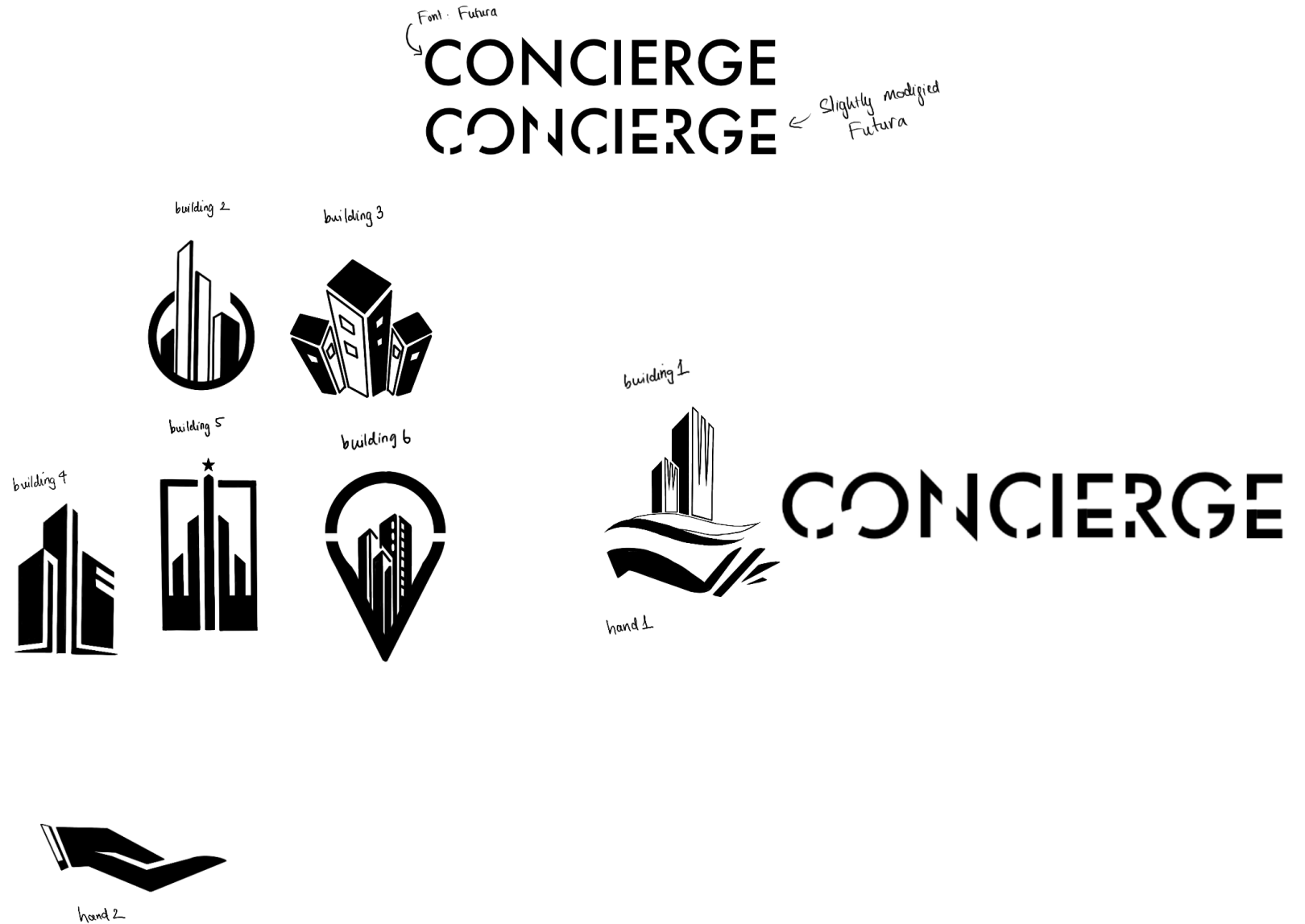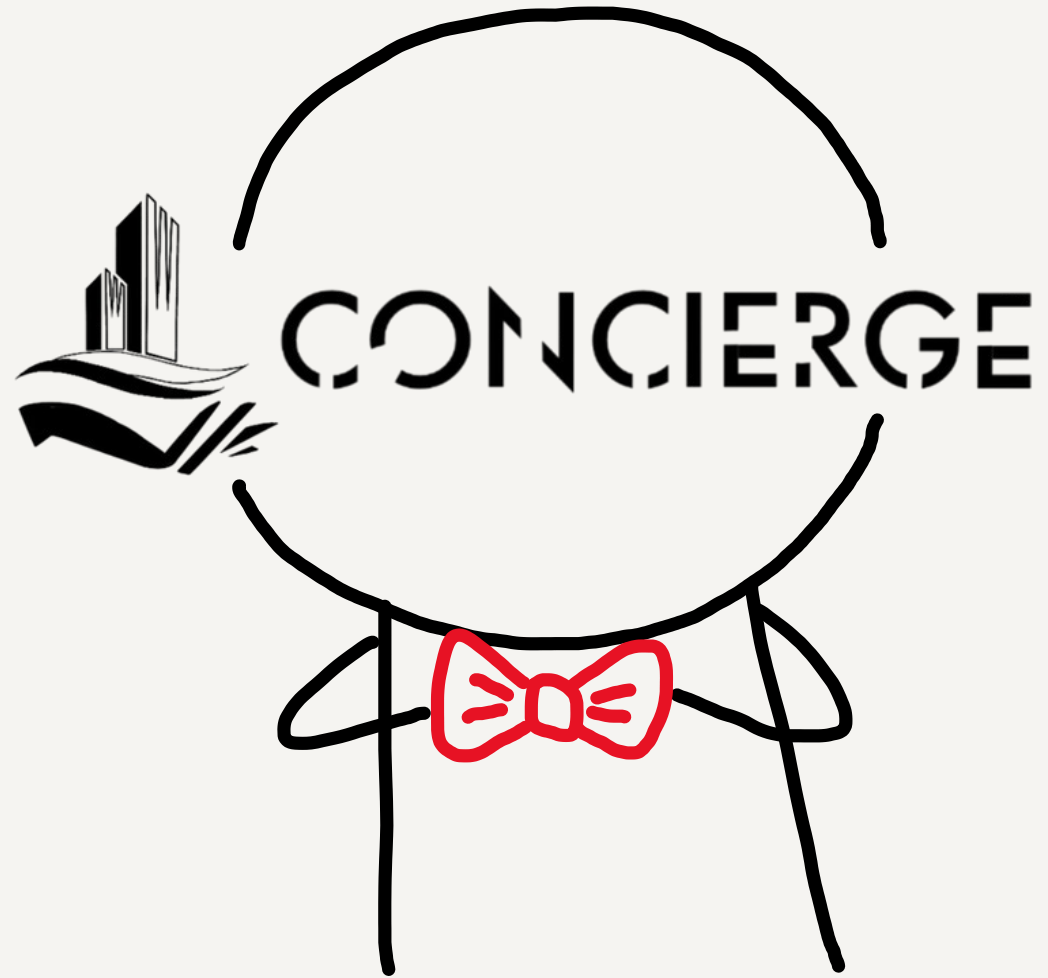


Image from Wiki

### System Design Team



Image from Imgflip
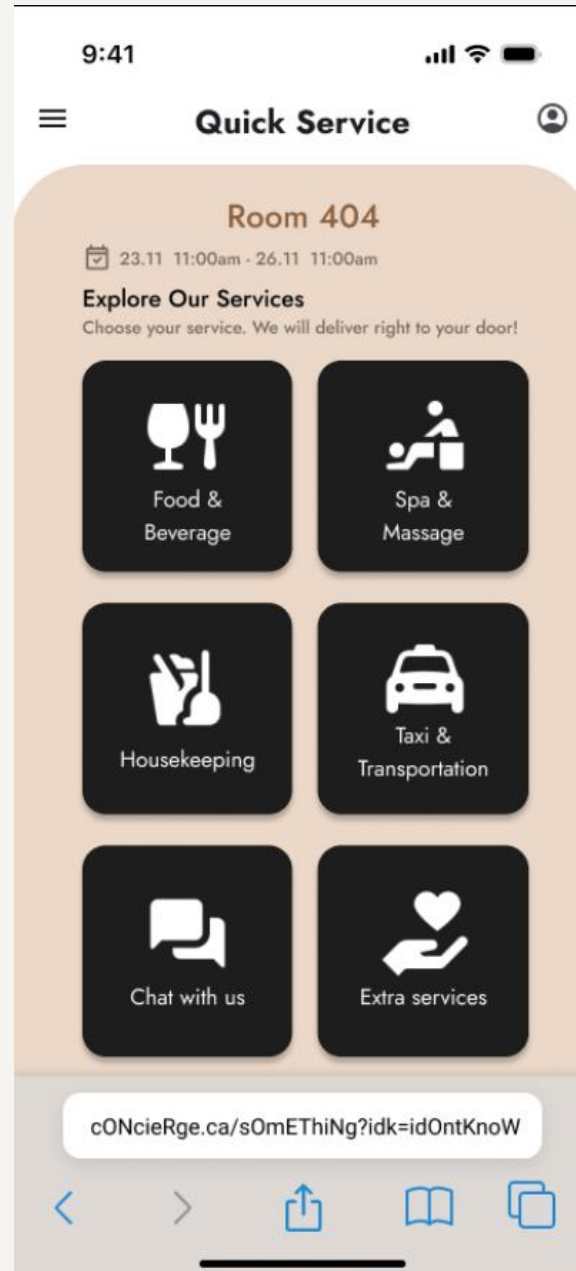
# Logo Design

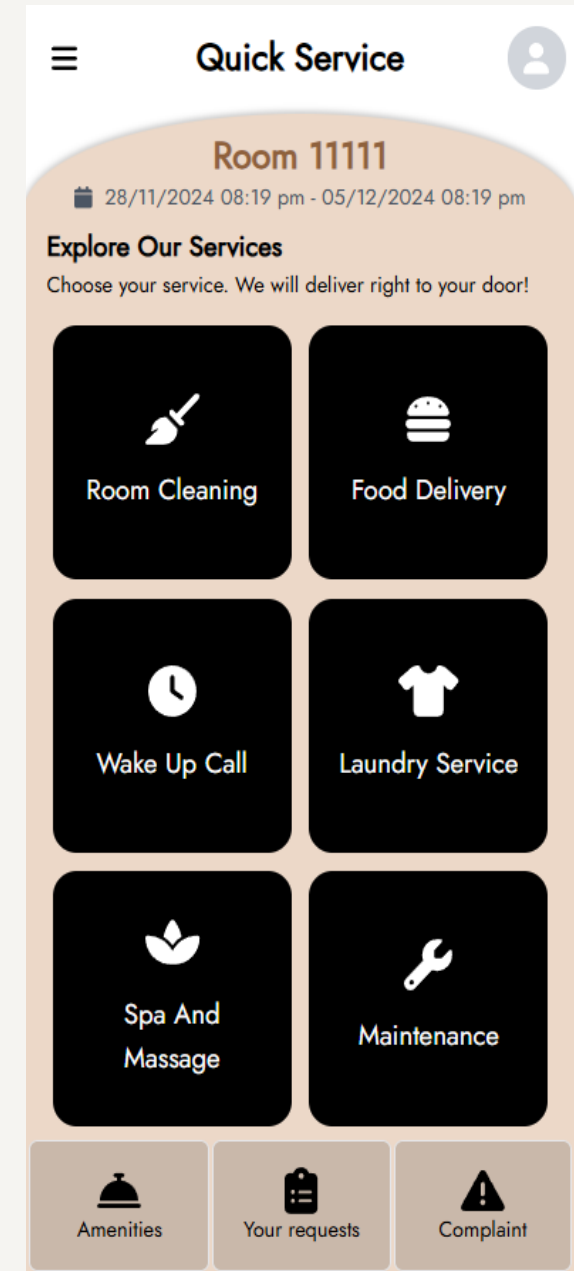"A caring hand for the hotel"

# Logo Design



Very cutesy

# Guest Front-end: Figma Prototype vs. Production
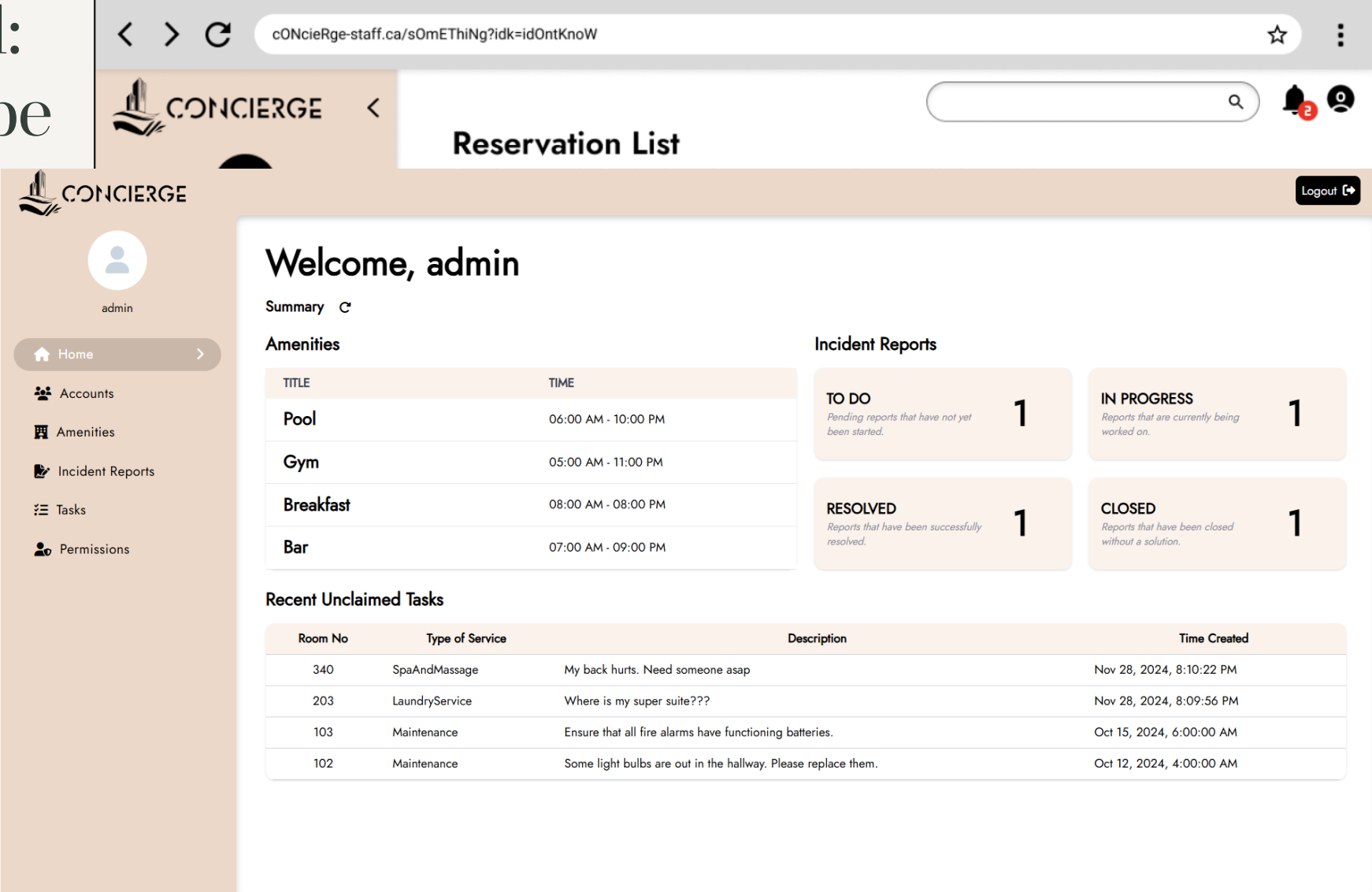
## Mobile-friendly Interface



Prototype



Production

# Staff Front-end: Figma Prototype vs. Production

## Desktop-first
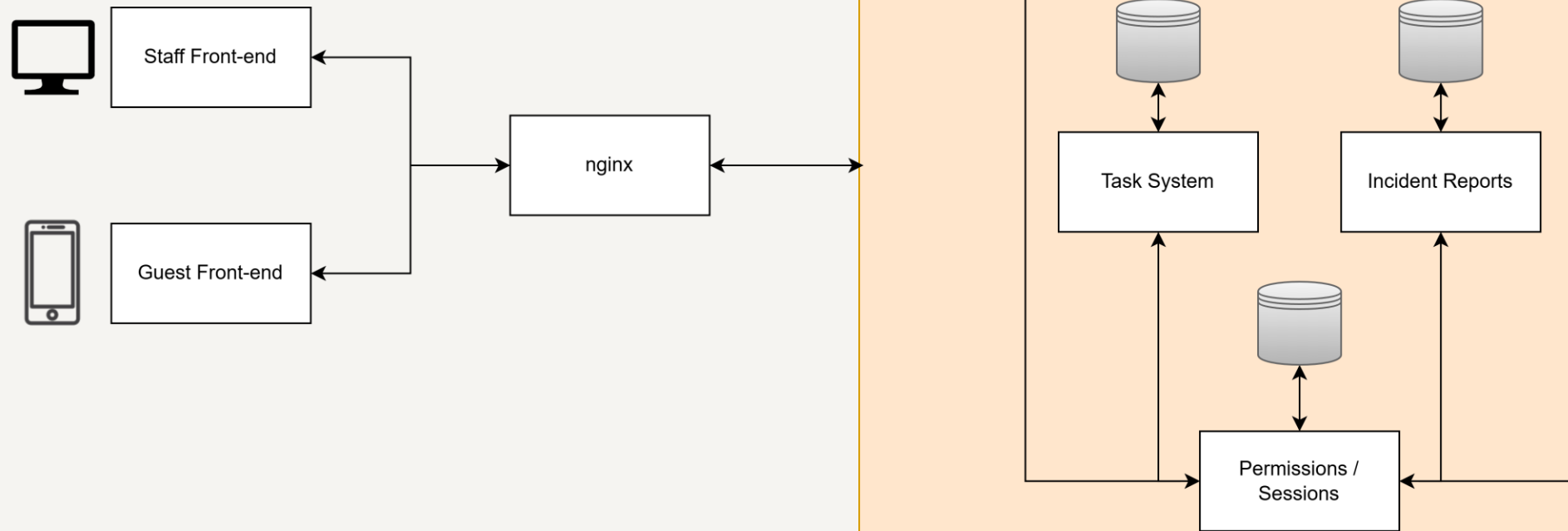
# Why invest time in visual designing/Figma?

- Clear reference

- Reducing confusion & conflicts

- Consistence



*Pro tip: Happy designers = Happy Developers
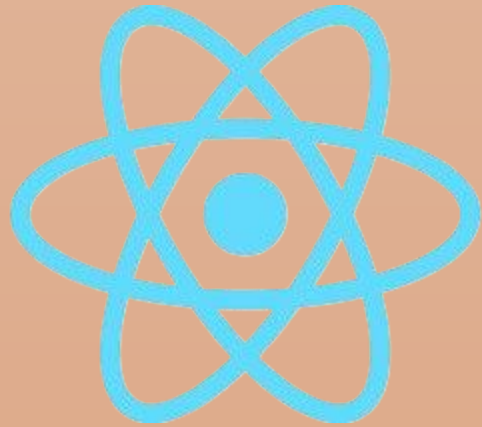
# THE SYSTEM DESIGN

# Architecture Diagram

# What is our Tech Stack?

yes

# The Front-end Tech Stack



Guest Front-end



Staff Front-end

# The Back-end Tech Stack

| Permissions | | Incident Reports | |
|---|---|---|---|
| Accounts | | Task System | |
| Amenities | | | |

# The Other Tools

Docker

Nginx

Cypress

Redoc

Github

# The Good

- We used no 3rd party services (for the app itself)

- Every piece was freshly baked in-house and made from scratch

- Using microservices, we were able to utilize a variety of tools and languages

- We were all able to learn a lot of new things in a short period of time

LET US DIVE IN

# The Writing it from Scratch

- 61313 LOC in total

| Front-end | | Back-end | | Other Supporting Files | |
|---|---|---|---|---|---|
| Total LOC: | 4,653 | Total LOC: | 14,291 | Total LOC: | 42,369 |
| Staff Front-end: | 3,548 | Accounts: | 2,715 | API Docs | |
| Guest Front-end: | 1,105 | Amenities: | 1,500 | | |
| | | Incident Reports: | 858 | | |
| | | Permissions: | 3,631 | | |
| | | Task System: | 2,039 | | |

# The Bad

- By not using 3rd party services, we did not gain the experience of such integration

- It took a tremendous amount of time to complete some of our assigned tasks

# The Learn For Next Time

- Integrate with services for user accounts and payment

- Increase focus on front end functionality

# THE DEV-OPS

# The Pipeline

All tests, deployment, and integration done in Github
Workflows
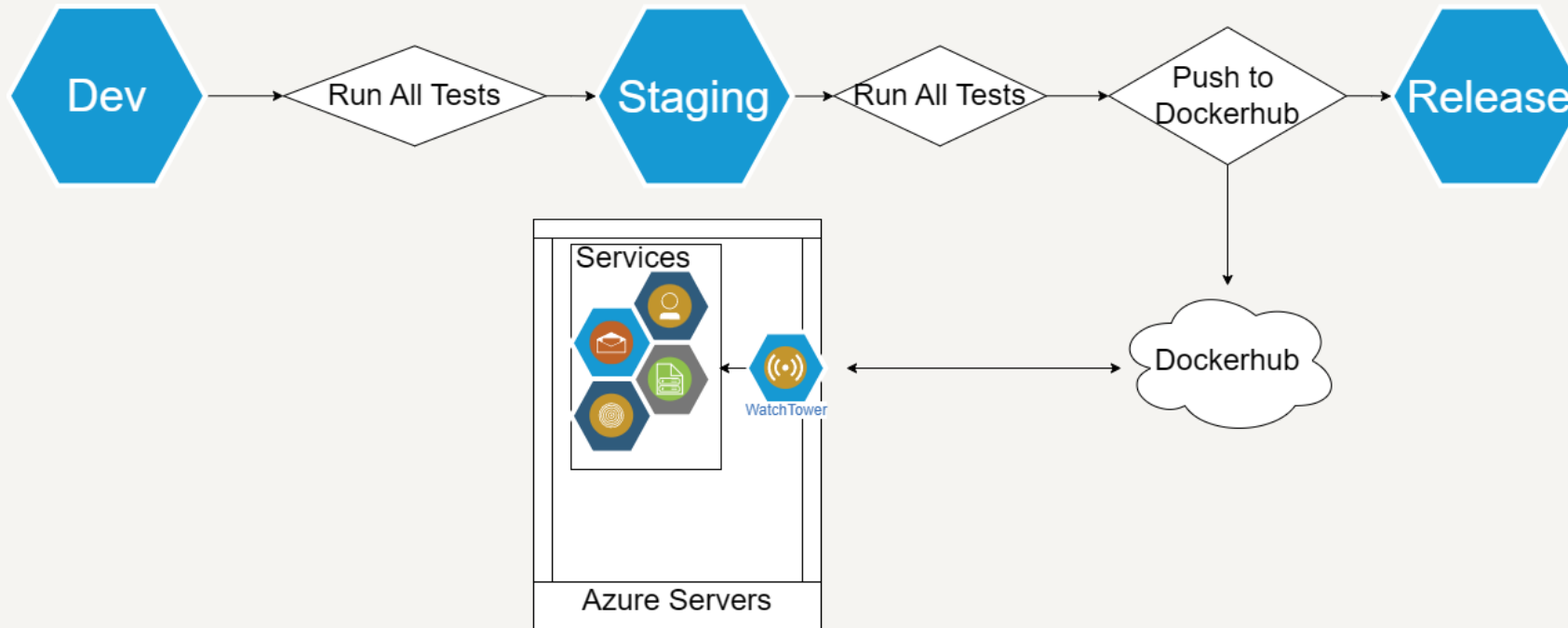
# The Good

- Cohesive CI and CD Configuration

- Free usage of machines for running pipelines

- Caching which sped up requirement downloads

- Experience with Azure

# The Bad

- Pipeline performance controlled by GitHub

- Difficult to manage all the different services in a way what was clean while not being wasteful of cpu time

- Overly verbose workflows sometimes to support microservices

- Free Azure server resource limits

# The Interesting
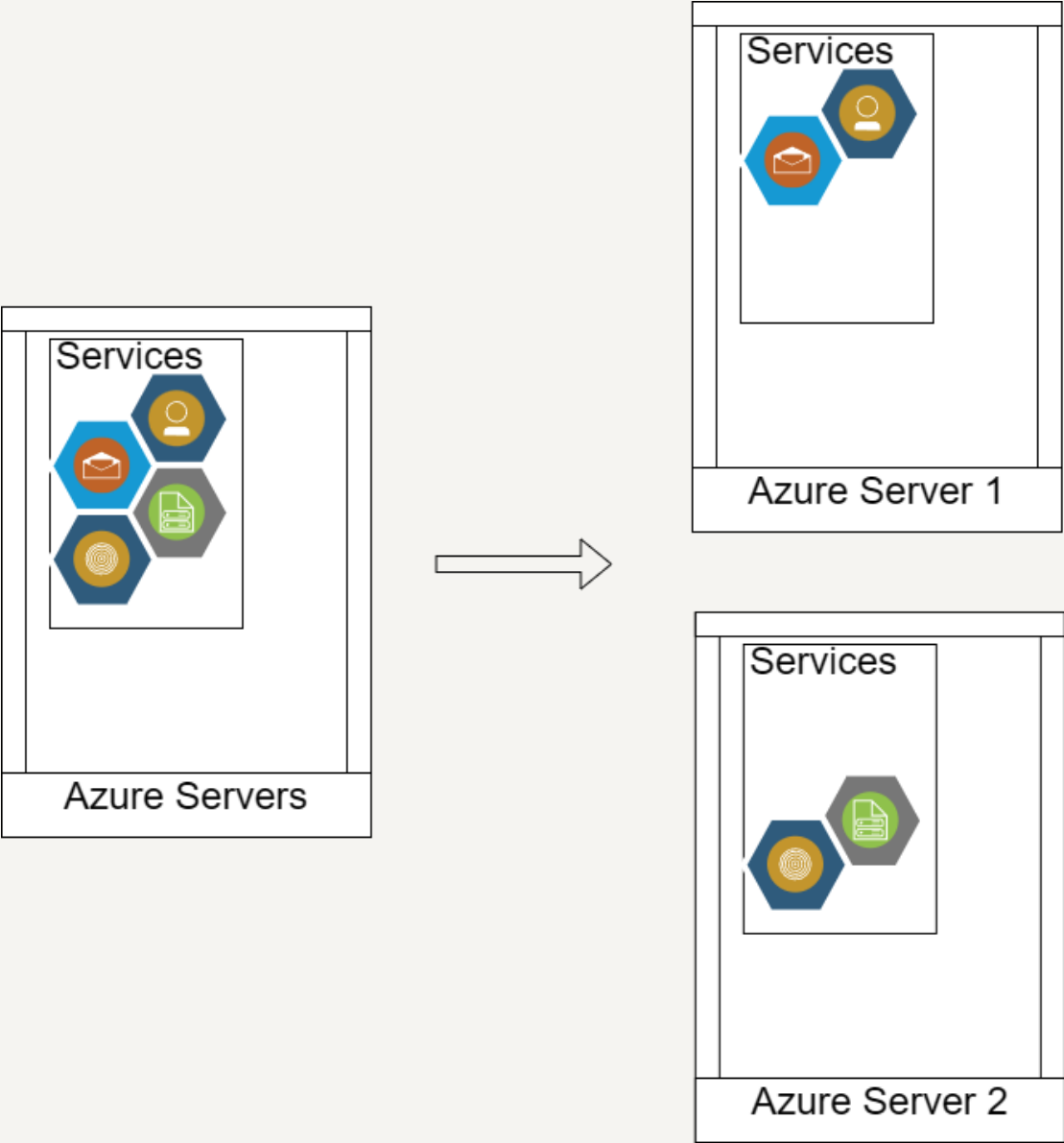
# THE "GROUP" IN GROUP PROJECT

# Did We Get Along?

- Teamwork and communication plays a big part in COMP 4350

- Even as a team of skilled programmers, we still had our own strengths and weaknesses to consider.

- Despite this, how did we overcome these challenges and collaborate to bring a shared vision to life?

# Member 1

*Area for improvement:*

Lack of experience/knowledge in working with modern developing tools

*How I grew:*

Put in a lot of effort into building a grand knowledge base

*Strength:*

Open to feedback/critique

# Member 2

*Area for improvement:*

Extra motivated to take on too much work

*How I grew:*

Reassessed task distribution

(Specifically, after sprint 1)

*Strength:*

Tremendous knowledge base able to support

All aspects of the project

# Member 3

*Area for improvement:*

Big yapper

*How I grew:*

Reigned it in, locking in, and removing distractions

*Strength:*

Boosted overall group morale

# Member 4

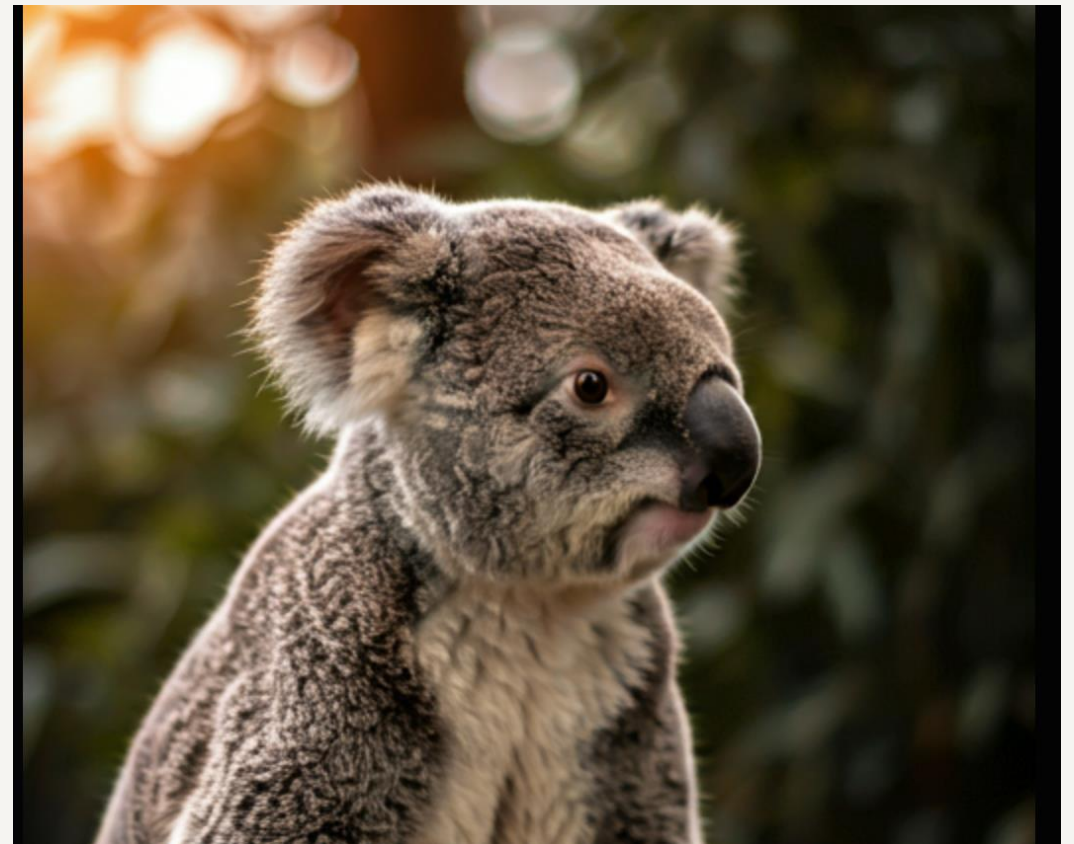*Area for improvement:*

Delayed Communication

*How I grew:*

Addressed the issue with the group and became more comfortable with my contributions

*Strength:*

Very passionate about front end development

And finding gaps in the project

# Member 5

*Area for improvement:*

Spent too much time on little details

*How I grew:*

Revaluation and task prioritisation

*Strength:*

Produces exemplary polishes

# Member 6

*Area for improvement:*

Not committed

*How I grew:*

Dropped the class

*Strength:*

Nice human

# The Good

- We had a good team with diverse but complimentary skills

- Everyone had a good attitude (even with the incident 🤐 )

- Everyone contributes and was able to gain experience in the areas they wanted

# The Not-So-Bad

- Lots big ideas that don't always fit in the allocated time

- Eagerness to learn new tools – that sometimes had a large learning curve

# Most important

Focusing on the hotel we worked hard

To build the application concierge

Lots of work and tears, painful to discard

But things went wrong, bringing time to absterge

Learning a lot of development tools

Through tough times, tears, and some mental bruises

Totally neglecting practical rules

Working hard we were off to the cruises

Thinking, coding, falling into a trance

With some good teamwork and strong leadership

We overcame issues and circumstance

Learning the most important part

Friendship :)

Now concierge bids you adieu

oh revoir 2 lee mond

Questions perchance? 😄 👍