

Software Requirements Specification Document

Cosy Koala IT

Vincent, Malzinskas
6474322

Aashim Lal, Memanaparambil Asokalal
103794571

Jordan, Zaz
6386601

Julian, Lai
103594920

March 2024

Contents

1	Introduction	3
1.1	Purpose	3
1.2	Scope	3
2	Project Overview	3
2.1	Domain Vocabulary	3
2.2	Pain Points	4
2.3	Domain Entities	4
2.4	Actors	4
2.5	Identifying Actors	4
2.6	Task - Brief	5
2.7	Stakeholders	5
2.8	Project Goals Author: Vincent	7
2.8.1	Primary Goals	7
2.8.2	Secondary Goals	7
2.8.3	Tertiary Goals	7
2.9	Assumptions Author: Vincent	7
3	Problem Domain	8
3.1	Data model Author: Vincent	8
3.1.1	Domain model Author: Vincent	8
3.1.2	Entity Descriptions - Detailed	8
3.2	Actors Descriptions - Detailed	9
3.3	Workflows Author: Vincent	10
3.4	Tasks	13
3.4.1	Work Area 1: Front of House	13
3.4.2	Work area 2: Back of House	13
3.4.3	Work area 3: Business, Finance and Administration	13
3.5	Complete Payment transaction onsite.	13
3.6	Take a booking	14
3.7	Take a tables order	15
3.8	Cook Food	16
3.9	Create Analytics	17
3.10	Update Website	18
3.11	Drops food to table	19
3.12	Share information Admin-Front of house, Admin-Back of house.	20
3.13	21
3.14	Functional Requirements and Task Descriptions	21
3.15	CRUD	21
4	Quality Attributes of System	21
4.1	Usability	21
4.2	Security	22
4.3	Reliability (Availability)	22
4.4	Performance	22
5	Other Requirements	22
5.1	Acquiring Knowledge using a Tacit Questionnaire	22
6	Validation of Requirements	22
6.1	Development Approach: Working with the Client	22
7	Possible Solutions	23

1 Introduction

This document contains the specifications for a operational information system (OIS) for the Cosy Koala restaurant in Hawthorn Vic. This document will outline the specifications, requirement and quality attributes of the operational information system

1.1 Purpose

The purpose of this document is to detail the functionalities and features of the OIS, aiming to streamline various restaurant operations as the business grows. From customer management and interactions to internal services and task management, the goal of this system is to allow the Cosy Koala to operate at a targeted customer patronage of 150 guests.

1.2 Scope

The OIS will facilitate the following high level tasks:

1. Customer Management.
2. Customer Interaction.
3. Internal Services.

2 Project Overview

The Cosy Koala currently sits 50 customers. It is making a physical expansion on premises to sit 150 customers. The current technical integration with operations is low. Taking orders from guest and interacting with the kitchen, and accounting is handled by hand. The Cosy Koala has identified multiple ideas they would like to include in the OIS:

1. The new system shall support reservations.
2. The new system shall support taking orders from customers.
3. The new system shall support information sharing with the Kitchen.
4. The new system shall support creating invoices.
5. The new system shall support creating receipts for customers.
6. The new system shall support handling payments.
7. The new system shall support basic statistics about ordered menu items.
8. The new system shall support online availability of the menu.
9. The new system shall support ordering from online take-away menus.
10. The new system shall possibly support arranging delivery.

2.1 Domain Vocabulary

1. OIS : Operation Information System.
2. FOH : Front of house.
3. BOH : Back of house.

2.2 Pain Points

1. Customer information:
 - (a) Managing 50+ customer orders.
 - (b) Taking onsite orders.
 - (c) Taking offsite orders.
 - (d) Collecting data on sales.
2. Front to back of house interactions:
 - (a) Organize orders.
 - (b) Make sure food is sent to correct customer.
3. Administrative functions:
 - (a) Gather customer data.
 - (b) Update website with marketing data.
 - (c) Analyse customer data.
 - (d) Provide offsite order and payment.
 - (e) Provide delivery.

2.3 Domain Entities

1. Meal
2. Payments
3. Customer
4. Front of House Staff
5. Seat
6. Table
7. Dining Room
8. Receipts
9. Sales Data
10. Order
11. Marketing Data, i.e. Menu lists, JPGs etc.
12. Point of Sales
13. Kitchen

2.4 Actors

2.5 Identifying Actors

As Software engineers we have a deep understanding of what is needed to create a successfully, effective and functional application; Versus what the client thinks the software needs in order to have those quality's. This statement implies we have the knowledge and skills to design and execute the development of some software, but a client or a group of stakeholders may have some tacit understanding of what needs to be done, which may have some Merritt, but is ultimately uniformed and may be inaccurate. By that same token, we software engineers may also have a tacit understanding of the roles of the client and stake holders. We attempt to bridge the gap by presenting some ideas and questions of what the client may need, what we expect of them and what their roles entail.

The first approach is coming up with a list of Actors; Defined as broad group of entities interacting within our business domain, like stakeholders, authorities and land-lord. We also list some functionality we assume they need from the program.

In this Scenario the team has identified , the stake holder in Koala Cafe to be:

1. Onsite Customer
2. Offsite Customer
3. Front of House Staff Member
4. Back of House Staff Member
5. Admin
6. Website
7. Delivery System
8. EFTPOS or Banking System

2.6 Task - Brief

1. Take onsite customer order .
2. Take offsite customer order.
3. Take onsite customer payment.
4. Take offsite customer payment.
5. Create offsite customer receipt.
6. Create onsite customer receipt.
7. Communicate order with Kitchen.
8. Seat customer.
9. Drop food to customer.
10. Deliver food to customer.

2.7 Stakeholders

- Business Operations
 - Finances
 - Roster
 - Web Portal
 - Admin Privileges
 - Marketing
 - Statistics/Analytics
 - Pay bills
 - Pay Salary
- Management
 - Set Roster
 - Alter Menu
 - Alter table data
 - * Arrange seating positions
 - * Alter table numbers
 - Approve Refunds
 - Order Stock
 - Have access to wait staff functionality
 - View feedback
- Barista

- Request Stock
 - view and print coffee orders
 - alert stock running low
 - Have access to wait staff functionality
- Wait Staff
 - Take Order
 - Process Payment
 - Reserve Table
 - Special meal Requests
 - See roster
 - Record feedback
 - Receive Tips
- Chef
 - Alter Menu
 - Alter Menu Items
 - Receive Order and print ticket
 - Confirm Order
 - Order Up
 - Order Stock
 - Alter kitchen roster
 - View Feedback
- Kitchen Staff
 - Request stock
 - Request Supply's
 - See Roster
 - Alert stock running low
- Maintenance
 - Order Supply's
 - track tasks
 - alter to-do list
- Customer
 - make booking online or via phone
 - order online via phone
 - order or book via App
 - order via QR code
 - order via Digital Kiosk
 -

Other Actors:

- Health and Safety
- Land Lord
- Suppliers

2.8 Project Goals Author: Vincent

2.8.1 Primary Goals

The primary Goal of Cosy Koala and the reason for releasing a tender for an OIS software provider is to be able to support a physical increase of guest from 50 to 150 in their restaurant.

2.8.2 Secondary Goals

1. Support increased business through take-away.
2. Support informed customers through their website.
3. Support informed management about the customers purchases.
4. Capture potential customers through their website.

2.8.3 Tertiary Goals

Cosy Koala is interested in the possibility of supporting the arrangement of delivery.

2.9 Assumptions Author: Vincent

A number of assumptions are made in relation to the implementation of this OIS is that manual scaling of operations are impossible. Whether that means no more staff will be available. Or whether staff will be expected to have other duties.

The system should model closely as possible to the current practices.

3 Problem Domain

3.1 Data model Author: Vincent

3.1.1 Domain model Author: Vincent

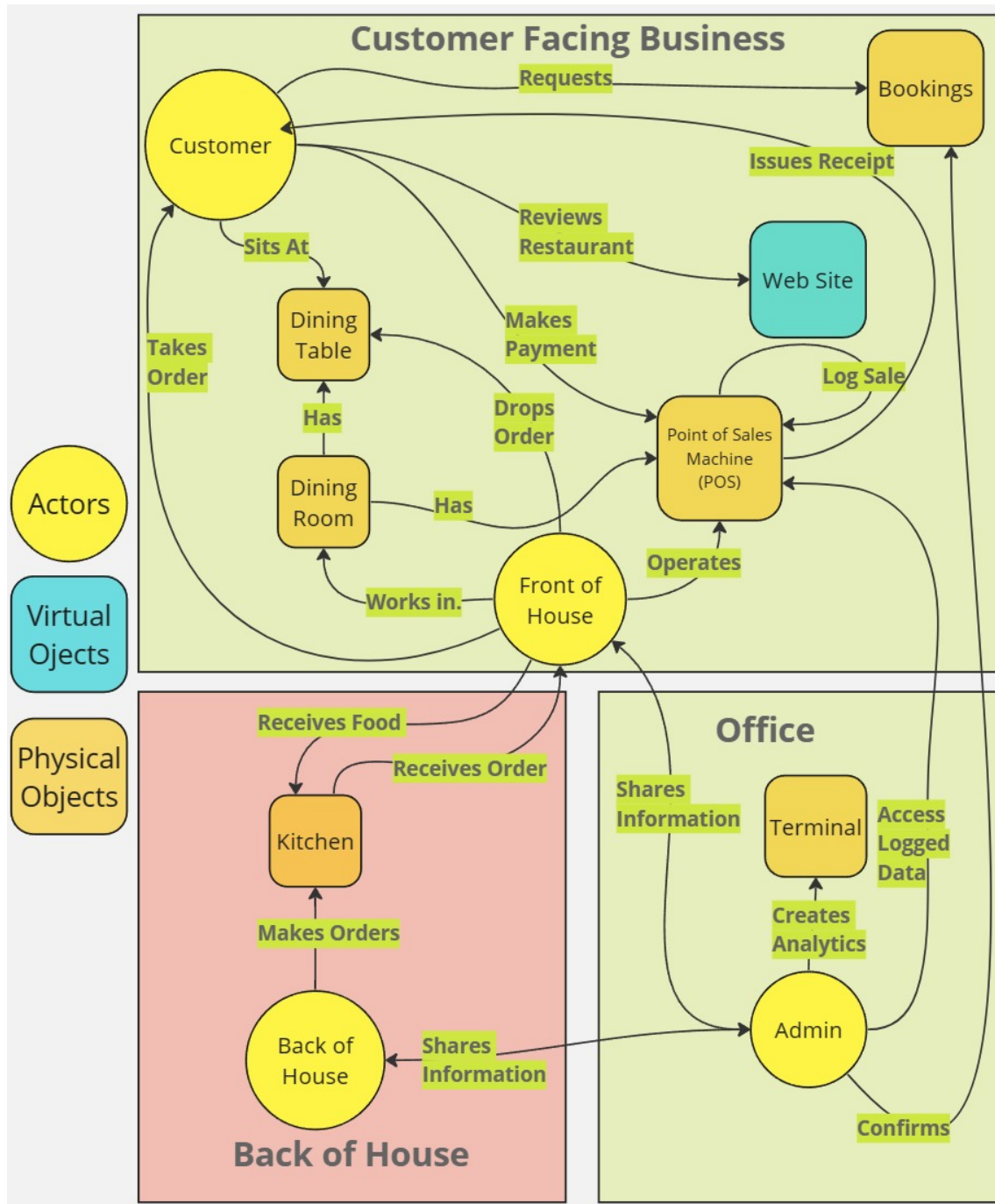


Figure 1: Cosy Koala Domain Model

3.1.2 Entity Descriptions - Detailed

Meal: To group together items for payment, delivery, or dropping at a table it would be helpful to place them into a unit of a meal. This way a meal can be paid for as one, or moved to a new table as one etc.

Payments: This will model the owed amounts for a meal, capable of dividing payments based on a number of divisions, i.e. by item, by table, by room for functions. It will interact with receipts and the Payment System (Actor).

Customer: This will model both the onsite and offsite customers, it will interact with bookings, tables, dining rooms, sales Data. It will contain customer specific data like meal, sales data.

Front of House: This will be a model for anyone interacting with customers capable of creating orders, bookings, taking payment and issuing receipts etc.

Seat: A customer can be assigned a seat to locate a meal or part of a meal to the correct customer.

Table: Contains seats and can be used for grouping meals into a larger single unit for booking or payment.

Dining Room: Containerizes and organizes customers orders i.e. If the restaurant has a multiple rooms this can be used to better locate customers for orders. It will contain tables which contain seats and customers.

Receipts: Will be a record for Sales data and Customers of a meal including its expense, items, time, location. Will be issues by the payment entity.

Sales Data: This is a aggregation of the Receipts data.

Order: (Onsite) Contains the meal order items, seat, table, dining room, that is ordered by a customer.
(Offsite) Contains the meal order items, customer that is ordered by a customer.

Marketing Data: This will be the data that can be sent to the Website (Actor) in order to update the website.

Point of Sales Machine: Facilitates payment transactions. Issues receipts. Logs sales data. Is operated be a front of house staff member. Interacts with the payment entity and the payment system (Actor)

Kitchen: Containerizes and organizes customers food order for pick up by front of house.

3.2 Actors Descriptions - Detailed

Onsite Customer: Is a person dining on food or beverages prepared by Cosy Koala. This is done on site in a dining room at a table. .

Offsite Customer: Is a person dining on food or beverages prepared by Cosy Koala. This is done off site at any location other than Cosy Koala dining rooms.

Front of House: This is a staff member who has direct contact with the customer. They take orders from customers. Take orders to the Kitchen. Bring customers their food. They are responsible for maintaining the orders integrity (Making sure the customer gets what they pay for). The take customer transactions. Communicate with Admin.

Back of House: Prepare the customers food order. Responsible for making sure front of house receive the food order with correct information i.e. order number etc. Communicate with Admin.

Website: Displays the restaurant information for customers.

Admin: Get the logged data from the Point of Sales Machine. Generate analytics from the Sales data.
Web Site: Display general information about the restaurant.

EFTPOS or Banking System: This is the outside system that will have to interact with the POS entity in order to facilitate payments.

Delivery System: This the outside system that facilitates delivery, like Uber, or menulog etc.

3.3 Workflows Author: Vincent

Task: Administrate the restaurant Actor: Admin Purpose: To manage the technical administration tasks so that everything from bookings to invoices are completed in a timely correct manner.	
Sub task	Example Solution
1. Review past data, including sales, costs, etc, particular attention is paid to the past day of service. Problem: Records are not accurate.	Communicate directly with staff from the previous day about the data generated. Automate data entry.
2. Update restaurant bookings, and communicate with customers. Problem: Bookings are unavailable for guest.	Communicate with the guest about an alternative date to book. Make the available dates and services on a given date accessible to customers.
3. Communicate and inform the FOH and BOH staff about upcoming customer demands, bookings and general workload. Problem: The bookings are changing rapidly or give short notice	Auto update FOH and BOH staff about bookings etc.

Table 1: Workflow: 1

Task: Prepare the kitchen Actor: Back of House Purpose: To prepare enough food or drink for the upcoming guest, planning at the service, day and week level is appropriate.	
Sub task	Example Solution
1. Review bookings for today and tomorrow in close detail as well as a week ahead in brief detail. Problem: Records are not accurate. Bookings have change but kitchen doesn't know.	Automate directly live data to the kitchen.
2. Receive orders and cook food. Problem: Orders are not accurate to items and customers	Deliver orders to kitchen directly from order entry at POS machine, - no hand written dockets.
3. Order needed food for upcoming bookings. Problem: Bookings have not been updated.	Automate directly live data to the kitchen.

Table 2: Workflow: 2

Task: Receive and dine guests Actor: Front of House Purpose: To prepare the room/s for upcoming guest, planning at the service and day is appropriate.	
Sub task	Example Solution
1. Review bookings for today in close detail. Problem: Records are not accurate. Bookings have change but front of house staff don't know.	Automate directly live data to the front of house staff.
2. Set tables and places for bookings and expected walk ins. Problem: Bookings have change but front of house staff don't know.	Automate directly live data to the front of house staff.
3. Receive guest/s. Take to booked seating. Problem: Guest/s has not booked	Find the available tables and seats.
4. Take orders from guests. Problem: The guest changes meal order, guest changes table or seat, guests leave, more guest come.	Have flexible table and meal placement that can be altered.
5. Take payment. Problem: Payment is to be split by meal, item, guest, table or some other way.	Make a flexible payment option.

Table 3: Workflow: 3

3.4 Tasks

3.4.1 Work Area 1: Front of House

Areas: Dinning Hall, Coffee station, Point of sales Greet guests, seat them, bring water to table, prepare drinks. Standing, assume medium-high frequency of traffic. Have responsibility over several tables occasionally helping when larger party's enter, Users: IT Novice, high school level education

3.4.2 Work area 2: Back of House

Area: Kitchen, Dry stock, Cool room, cleaning area Prepare Orders, Meal prep, Receive Delivery's, Specials Standing, fast past - dynamic environment, working in a team of 3-6 people, long shifts(8-12 hours) Have a variety of rolls in the kitchen, from cleaning to cooking Users: IT Novice, high school level education

3.4.3 Work area 3: Business, Finance and Administration

Area: Office Revenue, Spending, Ordering, Approving Changes, Web, Marketing, Hiring, Roster, Stats Small team, ad-hoc support from external services. Usually outsourcing work or working alone or in pair post breakfast/lunch rush Users: IT intermediate:, college level education

3.5 Complete Payment transaction onsite.

Task: Complete Payment transaction onsite. Purpose: Take the money owed to the restaurant after a sale of a food or drink and issue receipt. Frequency: Up to 150 times a breakfast, lunch or dinner service.	
Sub task	Example Solution
Locate the meal in system.	Enter the table number, and dining room name in order to find the meal.
Inform customer of details of meal, price etc.	The GUI can display relevant meal information.
Initiate payment system.	FoH Staff member can send the price of the meal to the EFTPOS machine.
Display whether the payment was successful.	The GUI can display to the customer and the staff member whether payment was successful.
Offer to print receipt.	The GUI could display an option for the FoH or Customer to select to print a receipt.
Print receipt	System send information to printer, possibly by network.
Move transaction to new state - paid	The system can validate the transaction was successful and move the meal to a paid/cleared data storage.
Variant:	
Customer only wants to pay for their meal and a table with multiple people	System should allow the FoH to select items from a meal for individual payment.

Table 4: Taking Payment

3.6 Take a booking

Task: Take a customers booking. Purpose: To pre-allocate a customer to a table in order to make sure the customer can dine when arriving. Frequency: Up to 150 time a service, 3 services a day.	
Sub task	Example Solution
Receive customers correspondence - email, phone call	Each day the admin will check the restaurant email and collect any bookings that have been sent.
For each booking review availability.	If some one wants to book a table for 4 at 8pm on Friday, the admin will first check if there is space for a table of 4 at that date and time.
Enter booking into record.	The admin will make a recording of the booking so that the resources cannot be double booked.
Inform the Front of house staff	The admin will transfer the bookings for today to a sheet for the front of house to organise the dining room to host.
Return confirmation correspondence to the customer	The admin will send a return email with the confirmation details.
Variant:	
The Admin receives a phone call.	This booking will need to be reviewed on the spot in order to make confirmation.
The booking is made on short notice	After the bookings and confirmation have already been made for the day. A new booking comes in. The admin will need to directly confirm and inform front of house to accommodate the booking.
The booking cannot be made.	The admin must reject the booking and offer a possible alternative.

Table 5: Take a booking

3.7 Take a tables order

Task:	
Purpose:	
Frequency:	
Sub task 1	Example Solution 1
?	?
?	?
?	?
Variant:	
?	?

Table 6: Take a tables order

3.8 Cook Food

Task:	
Purpose:	
Frequency:	
Sub task 1	Example Solution 1
?	?
?	?
?	?
Variant:	
?	?

Table 7: Cook Food

3.9 Create Analytics

Task:	
Purpose:	
Frequency:	
Sub task 1	Example Solution 1
?	?
?	?
?	?
Variant:	
?	?

Table 8: Create Analytics

3.10 Update Website

Task:	
Purpose:	
Frequency:	
Sub task 1	Example Solution 1
?	?
?	?
?	?
Variant:	
?	?

Table 9: Update Website

3.11 Drops food to table

Task:	
Purpose:	
Frequency:	
Sub task 1	Example Solution 1
?	?
?	?
?	?
Variant:	
?	?

Table 10: Drops food to table

3.12 Share information Admin-Front of house, Admin-Back of house.

Task:	
Purpose:	
Frequency:	
Sub task 1	Example Solution 1
?	?
?	?
?	?
Variant:	
?	?

Table 11: Admin communication

3.13

Task:	
Purpose:	
Frequency:	
Sub task 1	Example Solution 1
?	?
?	?
?	?
Variant:	
?	?

Table 12: Example Table

3.14 Functional Requirements and Task Descriptions

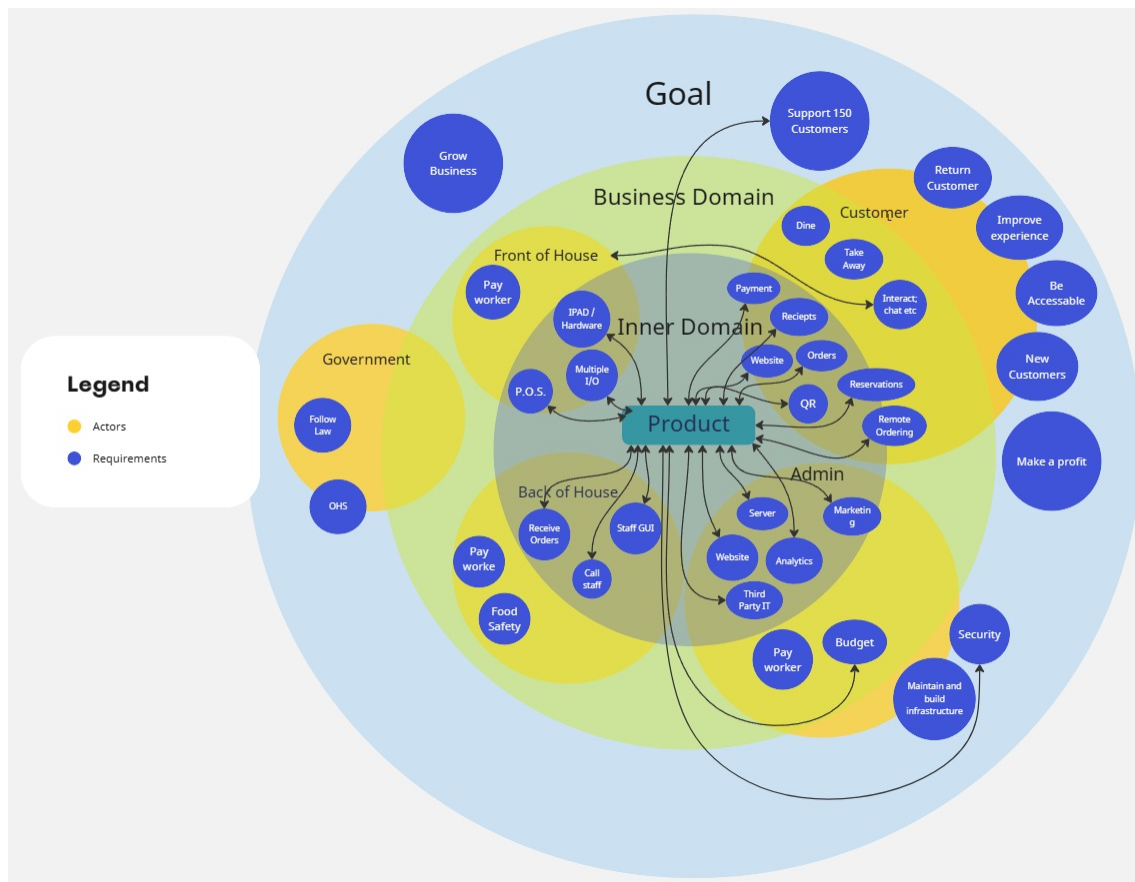


Figure 2: Cosy Koala Domain Requirements

3.15 CRUD

create, read, update, Delete

4 Quality Attributes of System

4.1 Usability

Goal: Ensure that the system is intuitive and easy to use for all users, including staff, customers, and management.
Detailed Description: The interface should be clear and straightforward, allowing users with minimal technical skills, such as children or elderly individuals, to navigate and use the system with ease. For example, a child

with basic phone usage experience should be able to place an order through a digital menu without difficulty. This can be achieved through, simple menus, clear labels, and helpful icons. The system should also provide accessible help resources and tool ps. Real-Life Reference:

4.2 Security

Goal: Protect sensitive data and ensure that transactions are secure. Detailed Description: Implement industry-standard security measures such as SSL/TLS for data transmission, encrypted storage for sensitive information, and regular security audits. The system should have role-based access controls to ensure that only authorized personnel can access certain functionalities. For example, only managers should be able to access financial reports or employee records and on the other hand it should also make sure that the customers payment details are encrypted and not accessible even by the managers. Real-Life Reference:

4.3 Correctness

Goal: Ensure the system processes information accurately and performs its functions correctly. Detailed Description: The system should validate input data to prevent errors and ensure the accuracy of orders, billing, and inventory management. Automated testing and quality assurance processes should be in place to detect and correct errors before deployment. For example, when a customer places an order, the system should accurately reflect the chosen items, quantities, and prices in the final bill. Real-Life Reference:

4.4 Reliability (Availability)

Goal: Ensure the system is operational and available during business hours without unexpected down times. Detailed Description: The system should be designed for high availability with failover mechanisms and redundancy to handle peak times and unexpected loads. For example, during a busy dinner service, the system should remain functional and responsive, managing multiple orders and reservations simultaneously without crashing or slowing down. Real-Life Reference:

4.5 Performance

Goal: Ensure the system responds quickly to user interactions and processes data efficiently. Detailed Description: The system should be optimized for fast response times, even under heavy load. This includes efficient database queries, optimized code, and the ability to handle concurrent users smoothly. For instance, when multiple customers place orders simultaneously, the system should process these quickly without delays or bottlenecks. Real-Life Reference:

5 Other Requirements

5.1 Acquiring Knowledge using a Tacit Questionnaire

(Quality Requirements) Careful construction of a tacit questionnaire allows us to extract data from stakeholders without requiring them to understand deep technical details. Instead, we are more focused on discovering what they expect or would like from the system, drawing from their experience and implicit knowledge of their role within their domain.

Here we are looking to extract some data about the and functional requirements

6 Validation of Requirements

6.1 Development Approach: Working with the Client

Ensuring that we closely align with a client's requirements at the early stages of the software development life-cycle is crucial in creating a robust and thoughtful software architecture. We are mindful that any problems faced now are much easier to tackle earlier rather than later on when they are "baked-in." Any changes that are made to later in the software's development can be costly and time consuming since some systems are likely to need re-programming or re-designing which can have a cascading effect on other system, which can lead to a larger restructuring.

The back and forth between the two parties; the client and producer (us) is a proven methodology (Validation) for creating a software system's architecture as we invest more time early to iron out any projected issues in partnership with the client. We consider all levels of domains: societal, environmental, global,business,customer

and methodically come up with strategies that we come up tacitly from a high level and systematically decompose the ideas from the abstract all the way down to the software's design. Source: IEEE's standards for software architecture documentation and the Software Engineering Institute's (SEI) Architecture Trade-off Analysis Method (ATAM).

The key is EARLY Validation.

7 Possible Solutions