UNIVERSITY of RWANDA

Research and Postgraduate Studies RPGS) Unit

Optimization and Disruption in Physical Retail Shopping Environment

Submitted By:

NGONGA DIPOKO BERYL JOCERAN

Registration Number: 221027761

College of Science and Technology

Master of Science (Option: Software Engineering)

October, 2024

OPTIMIZATION AND DISRUPTION IN PHYSICAL RETAIL SHOPPING ENVIRONMENT

By

Name: NGONGA DIPOKO BERYL JOCERAN

Registration Number: 221027761

A dissertation submitted in partial fulfillment of the requirement for the degree of

MASTER OF SCIENCE

(Option: Software Engineering)

In the College of Science and Technology

Supervisors:

Dr. Ghislain Maurice ISABWE

July 2024

DECLARATION

I declare that this project entitled: « Optimization and Disruption in Physical Retail Shopping Environment » is original and has been submitted only to the University of Agder (Universitetet i Agder) in Norway where the Master Thesis was conducted.
It is my research whereby other scholar's writing was cited and referenced. I thus declare this work is mine and was completed successfully under Dr. Ghislain Maurice ISABWE.

Reg No: 221027761

Signature: …….....

Date : 23 / 10 / 2024

UNIVERSITY of RWANDA

Research and Postgraduate Studies (RPGS) Unit

CERTIFICATE

This is to certify that the project work entitled "Optimization and Disruption in Physical Retail Shopping Environment" is a record of original work done by Beryl Joceran Ngonga Dipoko with **Reg No**: 221027761 in partial fulfillment of the requirement for the award Master of Science in Software Engineering of College of Science and Technology, University of Rwanda during the academic year 2020-2022.

Signature ……. Date: 23 / 10 / 2024

Student name: Beryl Ngonga

Supervisor

Dr. Maurice ISABWE

Signature ……. 23.10.2025

Head of Department

Mr Theoneste MURANGIRA

Signature : ……

Acknowledgements

In preliminary to any research, this is to address our gratitude to all those who contributed or not to ensure the smooth running of this research either morally, materially, financially and with ideas support.
I wish to thank the EQIP Project together with the NOPART for allowing me to be part of this student exchange program with the Support of the University of Rwanda (UR). My thanks go to Prof. Richard Musabe as my Main Supervisor in Rwanda, and to Dr. Christian Robere Simonsen and Dr. Ghislain Maurice N.Isabwe respectively Main supervisor and Co-supervisor in Norway, who gave up their preoccupations to help and facilitate during this research.
We also thank Dr. Frederic Nzanywayingoma for proposing us to this NOPART Project and checking upon us on the advancement of the Research during our stay in Norway.
To my family members (Mother & Sister) and friends for encouraging me and proposing ideas on the realization of this research.
And to all those whose names have not been mention.

In this research we decided to have an introduction where we explain WHY and HOW this research was carried and what brought us to do a research on this. The state of art is related to similar research project as the one we are currently carrying out. The Theory chapter is mostly focus on the theory we applied during our research and the methods chapter is how this theory was applied to our research.
And lastly not the least we presented result and research discussion on how to provide the solution to our research. Then we had our conclusion and appendix on how we collected data.

Abstract

Every year, queues cost brick-and-mortar retailers billions in lost revenues, and consumers are growing more impatient about standing in line. To survive the competition from e-commerce, stores need new innovations that can help kill queues Worldline, 2020.
If you have ever waited in a long line at the grocery store, for luggage at an airport etc. then you have an image of it and you know the pain it brings when you have to wait for someone.
Hence from these we understand that it is important to optimize checkout at retail environment(store or complex) and for this to be 100% efficient, the system to be proposed needs to be efficient and effective.
Bringing us to this research that we decide to conducted, which we shall solve it by applying our main theory which is the Queuing Theory. From these theory we decided to bring out two methods in other to solve this, firstly we will need to synchronise the result of our research (i.e. The System) with the retail environment for this optimization to be optimal, but from our research we also discovered that there are cases where the retail environment will not collaborate with us, hence looking at this situation, we decided to implement a digital disruptive system for it to be used at retail environment.
The slight difference here is that, the retail environment(store or complex) will not be in collaboration with us and hence we needed our system to be scalable and adaptable for it to provide the ability to the system to work without the needs of collaborating with the retail environment(store or complex).
Doing both of these cases, it will help us to increase the optimization of shopping in retail environment (store or complex).
The main aims of our research was to Optimize Queue in Retail Environment both in Rwanda and Norway since many people are still doing their shopping physically hence we did research on 4 theory which help us through out our research. Namely:

1. Queuing Theory
2. Microservices Theory
3. Software Design Theory
4. Qualitative Research Theory

Contents

List of Figures

List of Tables

Chapter 1

Introduction

As stated by (Myrset & Eriksen, 2019) Norwegian shopping centers experienced a sales growth of 17.6 percent in the period from 2013-2018, marginally stronger than overall growth in physical Norwegian retail trade. Based on (Bizclik, 2020) Rwanda is in the top five places of the inaugural A T Kearney African Retail Development Index (ARDI), The ARDI is a useful framework for retailers because it not only identifies the markets in Africa most attractive for retail expansion today, but those that offer the most potential for the future.

A report from (Research & Markets, 2022) shows that, The global food & grocery retail market size is expected to reach USD 14.78 trillion by 2030, registering a CAGR of 3.0% over the forecast period.
From those references we observe that Retail Environment(Stores or Market) has grown substantially over the years both in Rwanda and Norway and particularly around the world.
In 2018 and early 2019, Institute of Policy Analysis and Research (IPAR) in Rwanda conducted a survey, from this survey it was seen that Rwandan retailers account for over 1,476 tenants and 456 commercial buildings owners within 9 Commercial centres in the city of Kigali (Malunda et al., 2020) and in Norway, The Norwegian food & grocery retail market had total revenues of $31.4bn in 2020, representing a compound annual growth rate (CAGR) of 5.5% between 2016 and 2020, report from (Research & Markets, 2021).
This research investigated optimization and disruption in Physical Retail Shopping Environment, its impact on Rwanda and Norway, and the available strategies that was employed to solve this. A qualitative analysis was employed in carrying out the research.

1.1 Goal of Research

The goal for the research is focus on optimizing check-out process at Retail Environment (Store or Complex) which are using barcode reader for scanning their products before allowing customers to checkout. Five of the Retail Markets (i.e. T2000, Simba, Kigali Heights, UTC and Java House) on which we did research in Rwanda has not yet implemented a self checkout in their grocery stores, the approach to automated shopping and billing until now has been more checkout with someone scanning products. But here in Norway most have already been implemented which makes it easier to optimize the queues in retail markets, on the other side in case of where the retail markets are not willing to collaborate with the system, a disruptive digital system will be implemented. By helping customers to scan products barcode using their phones just once and then all those code are been saved by the system for it to be read after at the self-checkout without the removal of products from the trolley to scan back.

My approach for Rwanda is to provide a better shopping experience for the customers by saving their time, minimizing the man-power required at retail markets, while also significantly bringing out the cost of operation by eradicating the need for someone (person) to be there at the checkout post and allowing a customers to shop using just their smart phone, while allowing the retail markets to track customer purchases.

We propose to do this by using a smart phone application that allows the user to scan the products he or she wishes to purchase, generate the bill for all the products selected attach to the server directly (in the case where the system collaborate with us), and make the payment using his client ID. The entire process of bill generation is automatically carried out, and is displayed on the interface as the user continues shopping.

Once all the items are scanned with the barcode reader of the self-checkout, the customer confirms and place the order, the final bill is generated and the customer is redirected to payment options. The customer has the option to pay using the third parties payment services the system shall provide or he has the option to go with the payment offered by the store. And For Norway we shall provide the same facility as in Rwanda where the system works with the Retail Markets and hence ability to use the Client ID generated by the system.

However in case if in Rwanda or Norway the system doesn't work with the Retail markets we shall optimize check-out process by giving the ability to the customer to scan bar-code of each product at the store just once and then at check-out using the barcode reader to scan back those code but in this situation the barcode will not be on the products anymore but on the phone, Hence from phone permitting the faster scanning of the products.

1.2 Problem Statement

Every year, queues cost brick-and-mortar retailers billions in lost revenues, and consumers are growing more impatient about standing in line. To survive the competition from e-commerce, stores need new innovations that can help kill queues (Worldline, 2020).
From (Eiendom, 2019), The retail industry continues to face new challenges. According to the latest retail report from Virke, physical retail is still the main channel for the Norwegian consumer, but e-commerce, from both domestic and international platforms, grows with a faster pace and continues to gain market share. The physical retail industry is not collapsing, but changing, and going forward, the retail market will create some new winners and losers.
From (Bizimungu, 2015), According to Gael Murara, the GroceWheels chief executive he said GroceWheels have revolutionized the shopping experience of Rwandans that the shopping solution was inspired by the need to ensure Rwandan shoppers get deliveries anytime, anywhere. But still most people physically go to stores or markets, or send domestic workers to the market to get supplies for their households.
From this research conducted by the Firm above we immediately understand that most of the people still go physically to stores or markets, bringing directly a high probability of queues in retail markets and hence having an optimize checkout system in retail store or complex is will important and of a big advantage.
Hence our research problem turns around how to Optimize Queue in Retail Markets related to **checkout** and **self-checkout process**.

1.3 Research Objectives

The objective of this research can be formulated as Optimizing Queuing In-store and Self-Service in Retail Shopping Environment(Store or Markets) which reduce the queue attendance of the customer facilitating check-out process with a one-time scanning of barcode found on the products.

1.3.1 General Objectives

The General objective is first to Guide the customer during his entire shopping in the Retail store or complex and also to facilitated the checkout by making the customer to scan products himself which he will buy so that at the end of his shopping, from the checkout the bill is automatically calculated if the retail store or complex works with the system his unique client ID will be connected to the products he took that will be easy to bill all his product, then all he will be left to do is to provide his own Client ID (at checkout) to the system to automatically identify him and process his items for checkout.

1.3.2 Specific Objectives

The Specific objective of this research is to specifically focalize on:

- **Rwanda** Actually now in Kigali Rwanda majority of the Shopping Centers do not have self-service checkout, hence in this case the research will focus on two things.

Case 0: When the Retail store or complex is working with the system
Case 1: When they are not working with the system.

- **Norway** Actually now in Agder Norway majority of the Shopping Centers have both self-service checkout and the in-person checkout, hence in this case the research will focus on:
- Case 0: Facilitating the self-check-out process using a one-time scan product bar-code with the customers' phone.

1.4 Research Questions

The above works provides the justification for our work. It was summarized in section Research Goals. Following from those is a set of research questions that can be aggregated into one main research question (MQ):

MQ: *Could it be that a general system supporting shopping experience for every retail store or complex would be beneficial for solving queue and social distancing in retail stores or complex?*

Based on the main research question we can state the following sub research questions:
SQ1: How will the system solve the queuing problem at retail store or complex?
SQ2: How will the system communicate with all the different types of retail markets(Shopping centers or stores)

1.5 Thesis Structure

The remainder of this work is structured as follows:

1. **Introduction:** Presents the WHY and HOW the research will be done. It synthesizes the answers to the theoretical research questions.
2. **Theory:** This outlines the theory used in conducting this research.
3. **Methods:** This outlines the methods used on our research.
4. **Finding & Discussion:** The findings we got from these research and the discussion on how we shall apply it to the system.
5. **Conclusion:** It concludes this research and provide with recommendation to be taken.

Chapter 2

State of the Art

2.1 Background

2.1.1 Amazon Go and Fresh

Amazon Go is a chain of convenience stores in the United States and the United Kingdom, operated by the online retailer Amazon. The stores are cashierless, thus partially automated, with customers able to purchase products without being checked out by a cashier or using a self-checkout station (Wingfield, 2016) (Garun, 2016)
The flagship store sells products such as prepared foods, meal kits, limited groceries and liquor (Day, 2018). A larger variant, Amazon Go Grocery, opened in Seattle's Capitol Hill neighborhood on February 25, 2020. The following month Amazon began to offer its technology to other retailers so that their customers could make purchases without the involvement of cashiers or Amazon accounts (Kelion, 2020).
The app is required to enter the store, which has turnstiles that scan a QR code generated on the app. The app allows users to add others to their Amazon account, so a family's purchases can be charged to the same bill (Angus, 2019). The ceiling of the store has multiple cameras and store shelves have weight sensors, to detect which item(s) a customer took (Bosa, 2018). If a customer takes an item off the shelf, it will be added to the customer's virtual cart. Similarly, if a customer places an item back on the shelf, it is removed from the customer's virtual cart (Thompson, 2018).

As stated by (WikipediaFresh, 2022) Amazon Fresh opened their first grocery store in the Washington, D.C. area on May 27, 2021, in the suburb of Franconia, Virginia. On June 17, 2021, Amazon Fresh opened a location in the Seattle suburb of Bellevue that features the "Just Walk Out" technology which allows customers to skip the checkout line when purchasing their groceries. Amazon Fresh opened their first location in Washington, D.C. in the Logan Circle neighborhood on July 22, 2021; this location features the "Just Walk Out" technology. On August 5, 2021, the first Amazon Fresh grocery store in Pennsylvania opened in the Philadelphia suburb of Warrington.
Due to criticism regarding Amazon Go stores not accepting cash, Amazon Fresh stores also have checkout lanes for customers who want to pay with cash or don't have an Amazon account (Brown, 2018).

2.1.2 Alibaba Hema

As stated (Gomes, 2018) Alibaba also has its own cashierless grocery store, Hema. Launched in 2015, it's expanded to 46 stores in 13 cities in China. At Hema, self-checkout kiosks use facial recognition to connect with Alipay, the company's payment app, while digital screens display product details and dynamic prices that update automatically via Wifi-connected, e-ink price tags.

Based on (Doctoroff, 2018), There are 25 Hema supermarkets in China, with plans to open dozens more this year, and they're central to Alibaba's push for "new retail," or blending online and offline experiences. They have in-store restaurants so shoppers can select live seafood and eat on the spot. (The Chinese are so obsessed with freshness that there are two words for the concept, one for "crisp" fruits and vegetables and another for fresh-from-the-slaughterhouse meats.) Online shoppers receive free delivery within 30 minutes.

The combination of a user-friend app and barcodes on all items provides merchandise information, including sourcing, brand heritage, price points and nutritional value. In a country that's becoming a wallet-free society, you can purchase with Alipay, the Alibaba-affiliated digital payment system that has over 520 million users (Doctoroff, 2018).

2.1.3 Albert Heijn

Albert Heijn is the largest supermarket chain in the Netherlands with a market share of 34.8% in 2020. It was founded in 1887, and is part of Ahold Delhaize since 2016 (Heijn, 2006).

The internal concept idea is "The walk-in meal box". The Albert Heijn convenience supermarket is a relaxed and friendly neighborhood store focusing on fresh food and ready-made meals. The store is designed to fit the different customer journeys of the modern customers, whether it is a quick bite or a larger grocery trip, and it offers many omnichannel design solutions to improve the shopping experience (Heijn, 2006).

Based on (Gomes, 2018), Albert Heijn has implemented technology to let customers scan and bag items as they shop more easily than ever. They have two checkout-free stores where customers can tap their phone or credit card on a shelf tag for the items they want. 10 minutes later, the customers' bank accounts are debited for the amount they spent. If a shopper wants to put back an item, they reverse the chargers by tapping on the tag again.

Chapter 3

Theory on

For our research result to be efficient and effective we will be based it on 4 aspects in order to achieve it, Which are:

1. Queuing Theory
2. Microservices Theory
3. Software Design Theory
4. Qualitative Research Theory

3.1 Concept of Queuing Theory

As stated by (Kumar & Sharma, 2013), Queueing theory was introduces by A.K. Erlang a Danish mathematician who studied telephone traffic congestion problems in the first decade of the 20th century.
Queuing theory was very useful in many practical applications in areas such as, e.g., telephone exchange, traffic control, manufacture systems, inventory systems and communication systems, telephone exchange, supermarket, at a petrol station, at computer systems, etc (Kumar & Sharma, 2013).

Queues (waiting lines) are a part of everyday life. Every human being wait in queues to buy a ticket of railway ticket, make a bank deposit in the bank counter, start a ride in an pay ground park, etc. We have become familiar to huge amounts of waiting, but still get upset by unusually long waits by (Kumar & Sharma, 2013).
Queuing theory is the study of waiting in all these various fields (Kumar & Sharma, 2013).

A more understandable definition of queue was stated by (QueueIT, 2020), Queuing Theory refers to the mathematical study of the formation, function, and congestion of waiting lines, or queues.
Based on (QueueIT, 2020), a queuing situation involves two parts:
- **Requester of the Service** — it's usually referred to as the customer, job, or request.
- **Provider of the Service** — it's usually referred to as the server or checkout person.
Following we will illustrate a queuing process in a simple format ==>

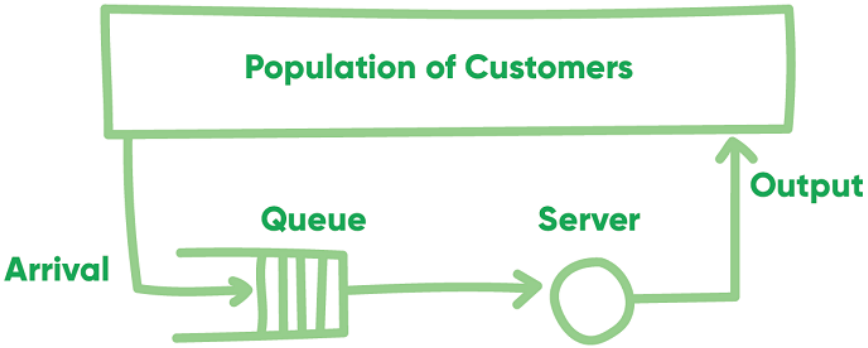

Figure 3.1: Queuing Process by (QueueIT, 2020)

From (QueueIT, 2020) Queuing theory scrutinizes the entire system of waiting in line, including elements like the customer arrival rate, number of servers, and number of customers, capacity of the waiting area, average service completion time, and queuing discipline.

Queuing theory uses the ***Kendall notation*** to classify the different types of queuing systems, or nodes. Queuing nodes are classified using the notation A/S/c/K/N/D where:
A is the arrival process
S is the mathematical distribution of the service time
c is the number of servers
K is the capacity of the queue, omitted if unlimited
N is the number of possible customers, omitted if unlimited
D is the queuing discipline, assumed first-in-first-out if omitted
For Example:
Case of An ATM ==>
It can serve: one customer at a time; in a first-in-first-out (FIFO) order; with a randomly-distributed arrival process and service distribution time; unlimited queue capacity; and unlimited number of possible customers.

3.1.1 Queuing Disciplines

As stated by (Kumar & Sharma, 2013), we can refer queuing disciplines to a priority system by which the next customer to receive service is selected from a set of waiting customers (e.g. FIFO, LIFO e.t.c).
As we already know a queue is a line of customer who requested for a services and they are awaiting for the products or service.
To manage this line of queue, we need to understand the queue discipline *the order of servicing and the behavior of your customers*(Tsernov, 2022).

First-in, first-out queuing (FIFO)

Based on (Kirill, 2022), A FIFO queue is a queue that operates on the first-in, first-out principle, hence the name. This is also referred to as the first-come, first-serve principle.
FIFO queuing refers to a queuing discipline where customers are served in the exact order in which they arrive. The first to join the line is the first one to leave it, all other factors being equal.

Last-in, first-out queuing (LIFO)

Based on (Kirill, 2022), LIFO Queue is a queue that operates on the last-in, first-out sounds illogical at first, but they have some power to it.
Professor Lars Peter Osterdal of the University of Southern Denmark thinks that LIFO actually has the edge over FIFO:
"The problem with a regular queue where you serve first those who arrive first is that people tend to arrive too early."
LIFO forces people to come at staggered times, resulting in shorter queues. Coming early poses more risks, which is why in Osterdal's experiment people chose to come later and spend as little time in the queue as possible.
There was nothing stopping some customers from leaving the queue and rejoining it to gain advantage, which had a negative effect on people trying to play by the rules (Kirill, 2022).
That's why LIFO is usually reserved for solving transportation and logistical problems, rather than issues of customers standing in line.

Priority queuing

Based on (Kirill, 2022), In priority queuing, some customers have a special status which allows them to skip the usual means of queuing.
This type of queuing is most commonly seen in industries where there can be emergency cases.
— For example, healthcare. A patient with a severe case is naturally treated ahead of everyone else.
In non-healthcare industries, this type of queuing is usually called VIP queuing. For instance, business class passengers board the airplane before others.

Service in Random Order (SIRO)

Based on (Rogiest et al., 2015), In the Service-in-random-order (SIRO) which is a well-known as an alternative to the related first-in-first-out (FIFO), SIRO is the scheduling discipline of choice in all cases where no ordering of customer requests can or should be established.
Just before starting a new service, the next customer to receive service is selected in a purely random way. If at that time N customers are present, all N customers join in a contention for the receipt of service, each having a probability $1/N$ to 'win' the contention (e.g., to be selected for service), regardless of the time already spent in system, the service time requirement, and the customer type (Rogiest et al., 2015).

Shortest Processing Time First (SPT)

Based on (Schrage & Miller, 1966) the shortest processing time, is the processing time of jobs that are known upon arrival and preemption without loss of time or processing already accomplished is studied. Priority is assigned to jobs according to the length of processing remaining with highest priority going to the job with least processing time.

3.1.2 Queuing Management

When we think of queues, the words that spring to mind are "annoyance", "lots of people" and "waste of time" (Tsernov, 2022).
But let's put it scientifically: What is a queue? A queue is a line of people waiting for the moment a particular service or product becomes available (Tsernov, 2022).
'The bigger the difference between the demand and supply, the longer the queue'. So then, what is queue management?
Queue management is a set of principles aimed at controlling customer flow and streamlining the queuing experience (Tsernov, 2022).

Waiting line Models Types

Based on (Kumar & Sharma, 2013), Queuing models provide the various types of queuing systems (systems that involve queues of some kind) that arise in daily practice.
Formulas for each model indicate how the corresponding queuing system should perform the service, including the average amount of waiting that will occur, under a diversity of circumstances (Kumar & Sharma, 2013). Therefore, these queuing models are very helpful for determining how to operate a queuing system in the most effective way.
For this, we need to look at two main parameters of queue management, which are the number of channels (or servers) and the number of phases of service (Tsernov, 2022).
Think of channels as the number of stations where you receive service, and phases as the number of steps you need to get full service.
As stated by (Tsernov, 2022), Each parameter can take two values: single (one), or multi (several). Different combinations of channels and phases give us four distinct types of queue management:
Single Channel, Single Phase
A single-channel, single-phase business has only one server. As soon as a customer is attended to, they receive full service (Tsernov, 2022).
Single Channel, Multi Phase
A single-channel, multi-phase business has one server and a multi-step servicing process (Tsernov, 2022).
Multi Channel, Single Phase
A multi-channel, single-phase business has several servers and a one-step servicing process (Tsernov, 2022).
Multi Channel, Multi Phase
A multi-channel, multi-phase business has several servers and a multi-step servicing process (Tsernov, 2022).

3.1.3 Queue Management System

Now that we know what **queue management** is, we can answer the next question:
What is a queue management system? A queue management system (QMS) is a set of tools developed to manage and analyze the flow of visitors (Tsernov, 2022).

Ideally, a queue management system exists to prevent the formation of queues altogether, though its use is not limited to queue managing Tsernov, 2022.
Joining a queue is the first of a customer, job, or request interaction with the business or server, Now taking care of this interaction is by using a queue management system.
Hence in other words, *a queue management system optimizes customer experience.*
Queuing theory singles out four key components: population of customers, method of arrival, service mechanism, and queue characteristics.

Population of Customers

Stated by (Tsernov, 2022), Population of customers refers to the number of visitors you are servicing. Depending on this number, we can describe the population of customers as:
Finite ==> when we know the limit to how many customers can be serviced.
Example: Model of a car repair factory, if it is being repaired, the repair arrival rate becomes zero.
Infinite ==> when there is no limit to how many customers can be serviced, due to random walk-ins.
Example: shopping checkout counters.
Unlimited population of customers is the norm for most businesses.

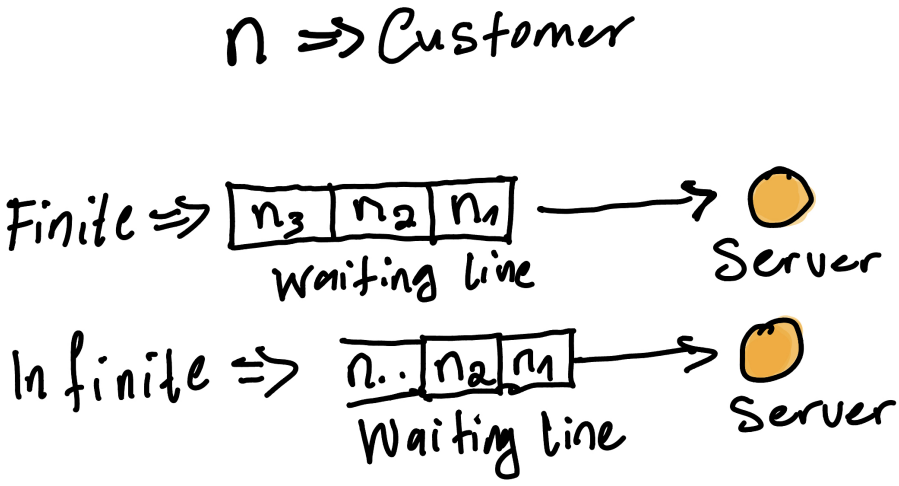

Figure 3.2: Finite and Infinite Customers

Method of Arrival

Stated by (Tsernov, 2022), Method of arrival describes the way in which your customers visit your business. Questions we considered:
- Do customers arrive individually or in groups?
- What is the time interval between two successive arrivals?
- What is the hourly/daily/weekly distribution of customer arrivals?
Analyzing this method of arrival helped us to figure out the arrival pattern. The arrival pattern as stated by (Medhi, 2002) The arrival pattern means the manner in which the arrivals occur, It is specified by the inter-arrival time between any two consecutive arrivals.

Service Mechanism

Service mechanism describes your services and required resources.
Questions we considered:
- How many in-person and self-service do the retailers have?
- How long does it take for a customer to be serviced?
The service mechanism helped us to figure out the service pattern. The service pattern is the manner in which the service is rendered, It is specified by the time taken to complete a service. Medhi, 2002.

Customer Behavior

Based on Tsernov, 2022, Customer behavior describes how visitors act during the moment they join the queue. 'Everyone wants to get patient customers, who join the line and wait however long they need without complaining'.
Types of Customers:
Balking — after seeing how long the line is, a customer leaves without joining it.
Reneging — a customer joins the line, then leaves without being serviced.
Jockeying — a customer joins the line, then moves to another queue.
Colluding — several customers cooperate, with only one of them waiting in the line.

3.2 Theory on Microservices

3.2.1 What are Microservices ?

Firstly we need to ask ourselves what is it before going dependly in the subject. So what exactly are microservices?
As stated by (Daya et al., 2016) Microservices is an architecture style, in which large complex software applications are composed of one or more services. Microservices can be deployed independently of one another and are loosely coupled.
Each of these microservices focuses on completing one task only and does that one task really well. In all cases, that one task represents a small business capability.
Also, microservices can be developed in any programming language. They communicate with each other using language-neutral application programming interfaces (APIs) such as Representational State Transfer (REST).
Microservices also have a bounded context. They don't need to know anything about underlying implementation or architecture of other microservices.

Figure 3.3: Microservices flow on the example of E-commerce application by Gos and Zabierowski, 2020

To better elaborate it we shall separate 4 aspects:

- Small and Focused
- Loosely Coupled
- Language Neutral
- Bounded Context

Small and Focused

By (Daya et al., 2016), Microservices need to focus on a unit of work, and as such they are small. There are no rules on how small a microservice must be. A microservice also needs to be treated like an application or a product. It should have its own source code management repository, and its own delivery pipeline for builds and deployment (Daya et al., 2016). Although the product owner might advocate the reuse of the microservice, reuse isn't the only business motivation for microservices.

Loosely Coupled

As stated by (Daya et al., 2016) Loose coupling is an absolutely essential characteristic of microservices. You need to be able to deploy a single microservice on its own. There must be zero coordination necessary for the deployment with other microservices.

Language Neutral

As stated by (Daya et al., 2016) Microservices are composed together to form a complex application, and they do not need to be written using the same programming language. In some cases Java might be the correct language, and in others it might be Python (Daya et al., 2016).

Bounded Context

As stated by (Daya et al., 2016) bounded context means a particular microservice does not "know" anything about underlying implementation of other microservices surrounding it. If for whatever reason a microservice needs to know anything about another microservice (for example, what it does or how it needs to be called), you do not have a bounded context.

3.2.2 What is a Monolothic Architecture ?

Based on (Gos & Zabierowski, 2020) A monolithic architecture refers to a system or software which has all or most of its functionality (i.e. authorization, business logic, notification module, etc.) within a single program or software.

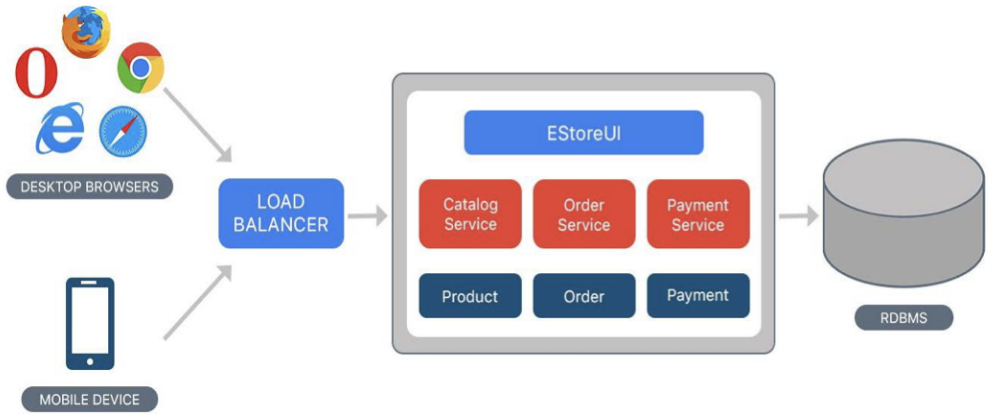

Figure 3.4: Monolithic flow on the example of E-commerce application by (Gos & Zabierowski, 2020)

Difference between Monolothic and Microservices Architecture

Category	Monolothic	Microservices
Code	A single code base for the entire application.	Multiple code bases. Each microservice has its own code base.
Understandability	Often confusing and hard to maintain.	Much better readability and much easier to maintain.
Deployment	Complex deployments with maintenance windows and scheduled down-times.	Simple deployment as each microservice can be deployed individually, with minimal if not zero down-time.
Language	Typically entirely developed in one programming language.	Each microservice can be developed in a different programming language.
Scaling	Requires you to scale the entire application even though bottlenecks are localized.	Enables you to scale bottle-necked services without scaling the entire application.

Table 3.1: Difference between Monolothic and Microservices Architecture by (Daya et al., 2016)

We can not talk of Microservices Architecture without talking of Service Oriented Architecture (SOA). Both have some similarity but are different.

3.2.3 What is a Service Oriented Architecture ?

The World Wide Web Consortium (W3C) refers to Service Oriented Architecture (SOA) 'A set of components which can be invoked, and whose interface descriptions can be published and discovered'. But IBM gave a better explanation which is a Service-oriented architecture (SOA) is an enterprise-wide approach to software development of application components that takes advantage of reusable software components, or services (IBM, 2021).
The comparison is complex and somewhat unfair, because proponents of microservices architecture never put forward a claim that it represents a genuinely new approach to building distributed systems. In that light, comparisons between SOA and microservices are normally suggested by the SOA proponents (Sprott & Wilkes, 2004).
Although in both cases we are indeed talking about a set of services, the ambition of these services is different.
SOA attempts to put these services forward to anybody who wants to use them. Microservices, alternatively, are created with a much more focused and limited goal in mind (Sprott & Wilkes, 2004).

Difference between SOA and Microservices Architecture

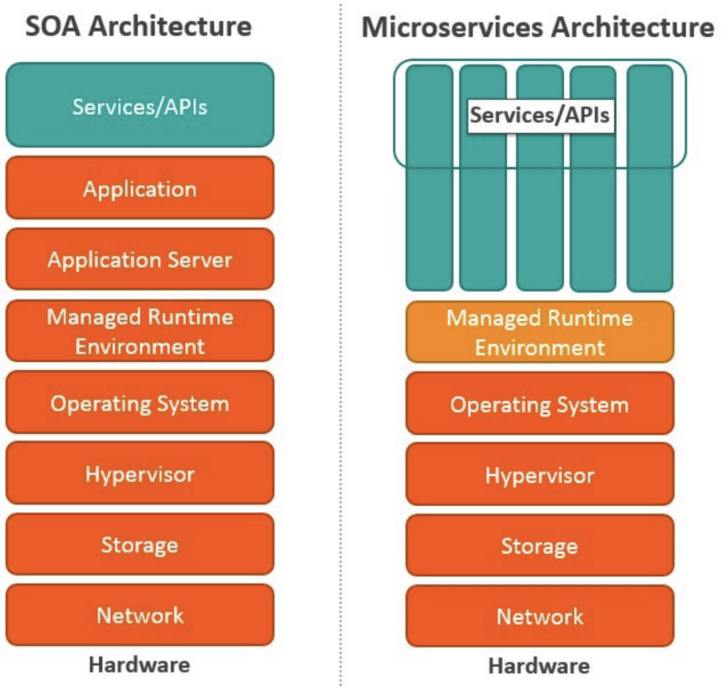

Figure 3.5: Difference between SOA and Microservices Architecture by Medium, 2019

3.2.4 Characteristics of Microservices Architecture

Design For Failures

Circuit Breaker

Based on (Daya et al., 2016), The circuit breaker pattern is commonly used to ensure that when there is failure that the failed service does not adversely affect the entire system. This would happen if the volume of calls to the failed service was high, and for each call we'd have to wait for a timeout to occur before moving on. Making the call to the failed service and waiting would use resources that would eventually render the overall system unstable. **Bulkheads** By (Daya et al., 2016), The bulkhead pattern is applied within microservices themselves. As an example, let's consider a thread pool being used to reach two existing systems. If one of the existing systems started to experience a slow down and caused the thread pool to get exhausted, access to the other existing system would also be affected. Having separate thread pools would ensure that a slowdown in one existing system would exhaust only its own thread pool, and not affect access to the other existing system. An example below:

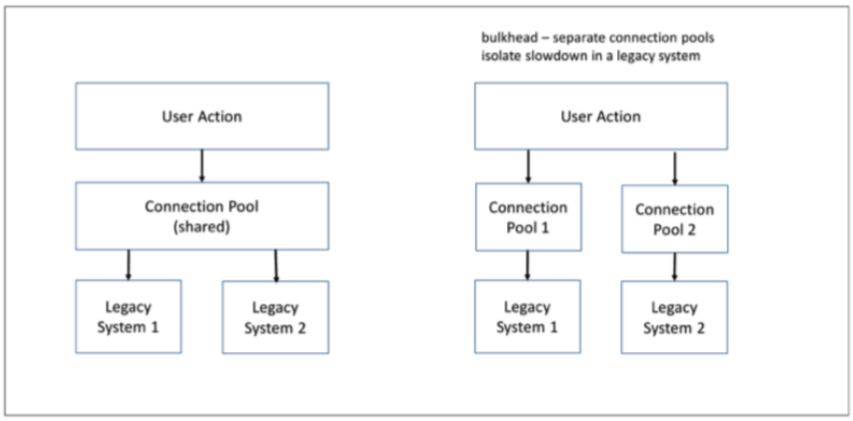

Figure 3.6: Bulkhead: Separate Thread Pool To Isolate Failures By (Daya et al., 2016)

Decentralized Data Management

In a monolithic application, it is easy to deal with transactions because all of the components are part of the monolith. When you move to a microservices architecture that is distributed, you now must potentially deal with transactions that are spread across multiple services. In a microservices architecture, the preference is for BASE (Basically Available, Soft state, Eventual consistency) over ACID (Atomicity, Consistency, Isolation, Durability). We avoid distributed transactions whenever possible. Ideally, you want every microservice to manage its own database.

Discoverability

Based on (Daya et al., 2016), We described earlier, microservices architecture requires making services reliable and fault tolerant. This in turn implies construction of microservices in such a way that, as the underlying infrastructure is created and destroyed, the services can be reconfigured with the location of the other services that they need to connect to.

Inter Service Communication Design

The inter service communication design mostly deals with two conditions:

- Inter Communication with each Microservice
- Inter Communication with each Microservice but through the API Gateway

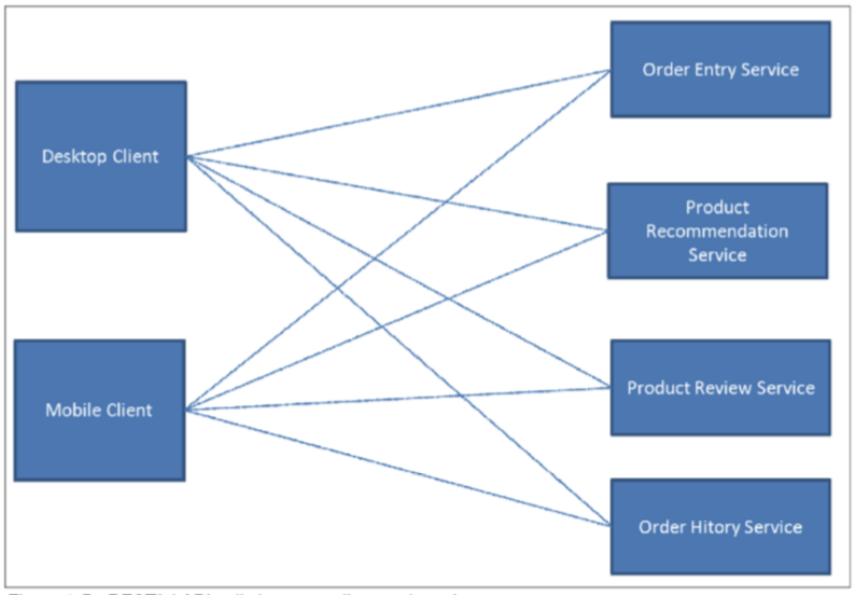

Figure 3.7: Restful Api Calls Between Client and Services By (Daya et al., 2016)

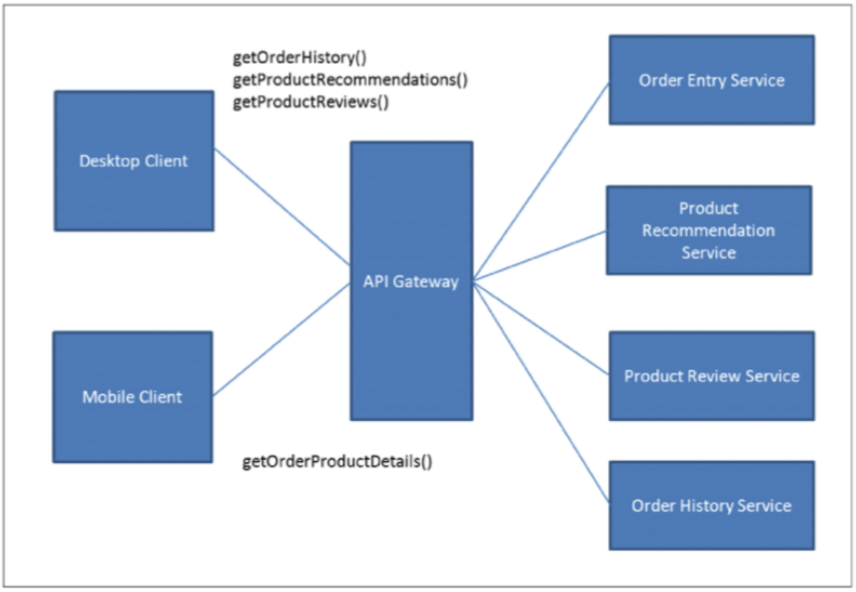

Figure 3.8: Restful Api Calls Between Client and API Gateway By (Daya et al., 2016)

Evolutionary Design

Based on (Daya et al., 2016), Microservices architecture enables an evolutionary design approach. The core idea is that you can introduce new features and capabilities in your application by changing a single microservice. You only need to deploy and release the microservices that you changed.

3.2.5 Challenges Microservices Has

The proposed micro-services approach has got its' own set of drawbacks. Stated by (Dmitry & Manfred, 2014), In practice, micro-services approach means for the developers the additional complexity of:
- Creating a distributed system (Testing is more difficult for distributed systems) (Dmitry & Manfred, 2014).
Probably, it is one of the main problems We must implement the inter-service communication mechanism. Also, very often, we will need some form of distributed transactions (Dmitry & Manfred, 2014).
- Multiple services will require us strong coordination within the team of developers. Or, what is more probable, between the teams of developers (Dmitry & Manfred, 2014).
- Deployment complexity will be increased. We need to deploy and manage many different service types. - The micro-services approach leads to the increased memory consumption. It simply, due to own address space for the each service (Dmitry & Manfred, 2014).
Lastly, One of the biggest challenges is:
- Deciding how to split (partition) the system into micro-services. One obvious approach is to partition services by use case (Dmitry & Manfred, 2014).

3.3 Theory on Software Design

Software design shares many characteristics with design in other fields. Much can be learned about software design by examining various philosophical and methodological viewpoints underlying design in general (McPhee, 1996).
In many cases, these viewpoints have shaped attitudes towards software design and software design methods, techniques, and tools.
Typically, design methodologists from the science and engineering communities characterize design as a type of problem solving or decision theory where the initial conditions, the goal, and the allowable transformation operations are all ill-defined (McPhee, 1996).
Over the years, computing science researchers and practitioners have developed many different software design methodologies, process models, techniques, and tools (McPhee, 1996).
For example: Waterfall Development Model
The classic waterfall development process (Agresti, 1986) (Boehm, 1976) (Royce, 1970), with its emphasis on separate "analyze-synthesize-evaluate" phases, is an echo of the classic scientific method and logical empiricism.
The solution is implemented and tested in an evaluation phase. The classic waterfall separates the "what" from the "how" with no opportunity for feedback (McPhee, 1996).

Hence from these we can understand that the waterfall development model is not the complete development process.
Practitioners recognize the strengths and weaknesses of each style of designing and create a blend of methods that best serves their needs. For example, incremental prototyping may provide the methodological framework for a real-time software project in which formal methods are embedded to ensure correctness of critical parts of the design.

Based on (Maruping et al., 2009) From structured development techniques to rapid application development to object-oriented design, software engineers have continually sought new methodologies and development approaches to address an evolving market for software.
Persistent limitations of extant approaches in effectively addressing requirements change have led to the emergence of a set of software development methodologies collectively referred to as agile methodologies. Some well-known agile methodologies include eXtreme Programming (XP), Scrum, Feature Driven Design, Test Driven Development, Crystal, and Lean Programming (Maruping et al., 2009).

3.3.1 Scrum Framework

Scrum is a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value (Ken Schwaber, 2020).
Scrum is a lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems. Scrum co-creators Ken Schwaber and Jeff Sutherland have written The Scrum Guide to explain Scrum clearly and succinctly (Ken Schwaber, 2020).

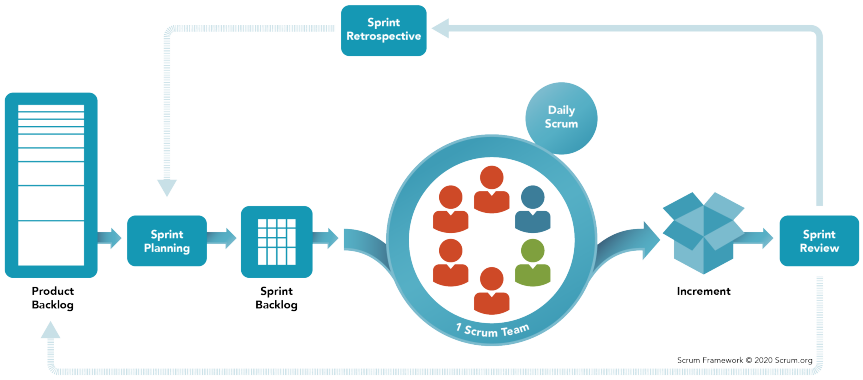

Figure 3.9: Scrum Framework by (Ken Schwaber, 2020)

3.3.2 UML Standard

The Unified Modeling Language (UML) is a general-purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system. **??**. In 1997, UML was adopted as a standard by the Object Management Group (OMG), and has been managed by this organization ever since. In 2005, UML was also published by the International Organization for Standardization (ISO) as an approved ISO standard (ISO/IEC, 2005) Since then the standard has been periodically revised to cover the latest revision of UML.
The UML Diagrams incorporate

- Use Case
- Sequence
- Activity
- Deployment

and many more diagrams in other to help to build a efficient system.

Use Case Diagrams

As stated by (Jacobson et al., 2016),
Use Case: A use case is all the ways of using a system to achieve a particular goal for a particular user. Taken together the set of all the use cases gives you all of the useful ways to use the system, and illustrates the value that it will provide.
Use-Case 2.0: A scalable, agile practice that uses use cases to capture a set of requirements and drive the incremental development of a system to fulfill them (Jacobson et al., 2016).
Use-Case 2.0 drives the development of a system by first helping you understand how the system will be used and then helping you evolve an appropriate system to support the users (Jacobson et al., 2016).It can be used alongside your chosen management and technical practices to support the successful development of software and other forms of the system:

- lightweight
- Scalable
- Versatile
- Easy to use

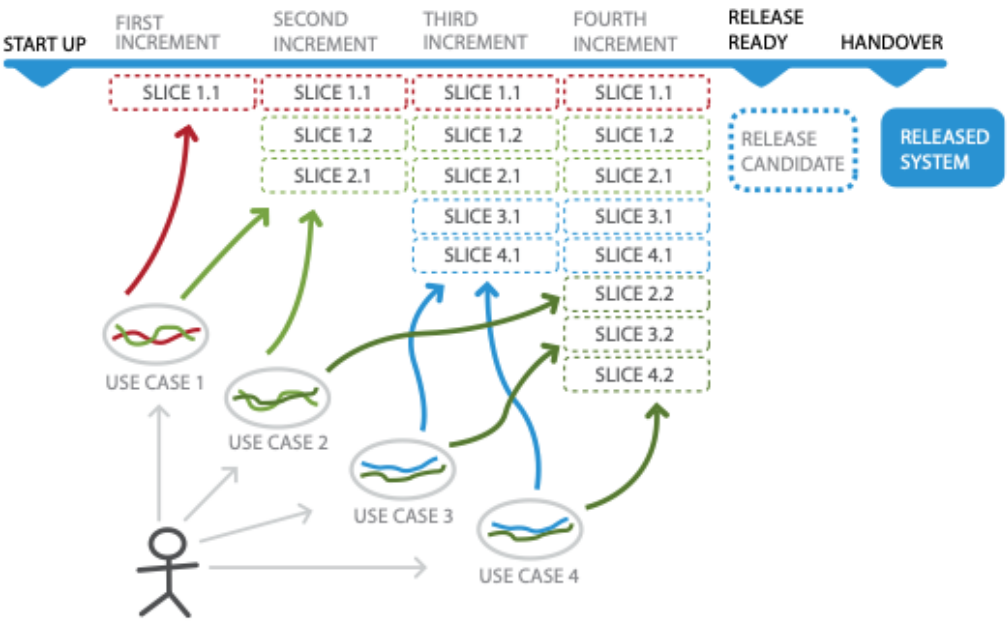

Figure 3.10: Use Cases, Use Case Slices, Increments, AND Releases by (Jacobson et al., 2016)

3.4 Qualitative Research Theory

Based on (Silverman, 2020), the goal of Qualitative research it sets out to overcome several erroneous assumptions about our field that may confuse the beginning researcher. One false assumption is that there is general agreement about the nature of qualitative research.

Qualitative research is the type of research that finds out about people's experiences. It helps us understand what is important for people (Silverman, 2020). The ability to get inside people's heads appears to be what distinguishes our work. By implication therefore, qualitative research is claimed to be concerned with subjective 'meanings' and hence very different from quantitative research's pursuit of objective 'facts' (Silverman, 2020).

Such a conception of our field is very common among established researchers and is often uncontested in the contents of leading academic journals. The vast majority of journal articles assume a consensus about the following aims of qualitative research (Silverman, 2020):

- Understanding human experience
- Treating interviews and focus groups as providing direct access to the contents of people's heads
- Foregrounding the empathetic skills of the researcher to achieve this.

By (Silverman, 2020), at least 90 per cent are using interviews with the aim of seeking to understand the experiences of some group. Contested version of qualitative research which topicalizes perceptions or feelings and relies on asking people what they think or feel.

3.4.1 Covert Research

(Silverman, 2020) stated that, The combination of informed consent and covert research serves as the ultimate research ethical problem. Just as the distinction between overt and covert research often is unclear, (Denzin & Lincoln, 2011) reminded us years back that to make written informed consent mandatory would mean the end of much 'street-style'.
Examples Punch reminded us that the more deviant and secret the activity, the more subjects may fear disclosure and the bigger the responsibility for the researcher to protect them (Silverman, 2020). Still, **Confidentiality** and **Trust** is an even more complex issue.

Confidentiality

Based on (Silverman, 2020), Confidentiality means we are obliged to protect each participant's identity and the location of the research. Qualitative researchers practise this by using pseudonyms or by blurring personal identities, which makes anonymity one form of confidentiality. However, anonymity by pseudonyms also carries connotations of gender, age, class etc. This makes some to prefer to use numbers or particular code to refer to them without providing their information.

Trust

Stated by (Silverman, 2020), Trust refers to the relationship between the researcher and the participants, and to the researcher's responsibility not to 'spoil' the field for others in the sense that potential research subjects are reluctant to be studied (Ryen et al., 2004).
In this way, trust also applies to the report or the discursive practices defining the standards for presenting both the researcher and the work as trustworthy (Fine, 1993). Trust is the classic key to good field relations and is a challenge constantly unfolding during the research process, though more so in ethnographic studies than in other kinds of fieldwork (Silverman, 2020).

Chapter 4

Methods

4.1 Research Methodology

The research method we shall adopt will be qualitative and Quantitative study that seeks to answer the research problem and sub-problems by means of Record Keeping, Informal one to one interview and also Process of Observation. The Process of Observation and Record Keeping (also known as Reflective Journal Log) will used to come up with a body of knowledge relating to the research problem. Methods used are:

- Informal One-to-One Interview
- Record Keeping
- Process of Observation

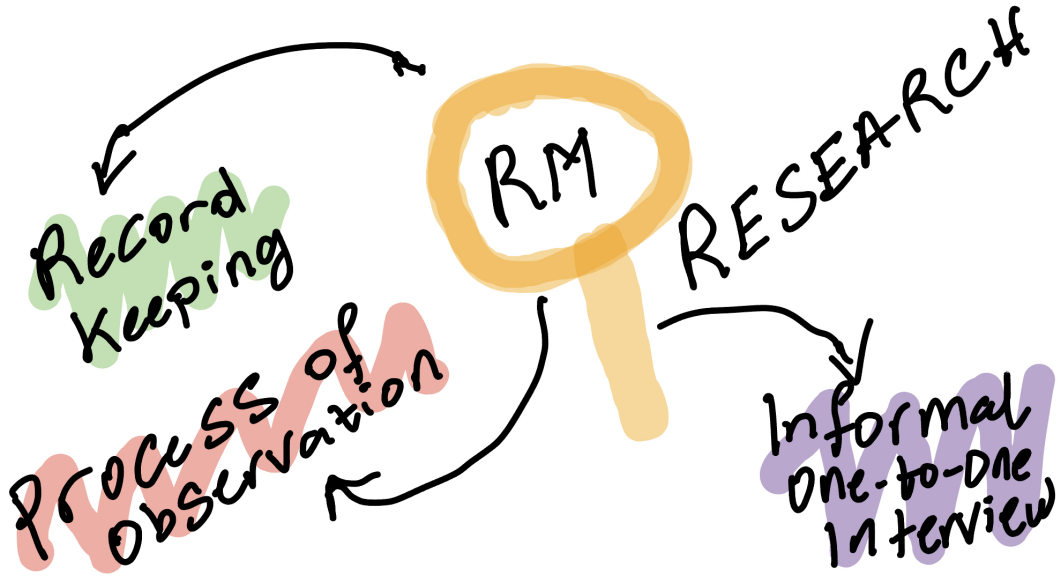

Figure 4.1: Research Method

4.2 Interview

Informal interviews are interviews that take place outside the office in a casual setting, such as over lunch or coffee.
But in our case we did small questionnaire that we got in head and we decided to ask them. It was free to answer for each participants and without disclosing their identity.
Informal interviews are not structured like a traditional interview, although they typically have the same goal.

4.2.1 Why Informal One to One Interview ?

At the beginning we were not able to collect information from head of Retail store, they were not authorised to give us those information.
Hence, This made us to go for informal one to one interview, where this time the cashier were the one to answer our questions if they wanted or not.

4.3 Record Keeping

The skill of recording data involves the documenting of data and observations in a variety of forms in order to preserve it for later use.

4.3.1 Why Record Keeping ?

It helps us to gathered and organize our qualitative and quantitative observations and data to make it useful for analysis and interpretation.

4.4 Process of Observation

An observation procedure is a process used to observe and record a set of activities or behaviors (WorkplaceTesting, 2018).
As stated by (WorkplaceTesting, 2018) In general, an observation procedure or method refers to any research which is based on either formal or informal observation.

4.4.1 Why Process of Observation ?

Due to the fact we could not collect information from head of retail store, we had to do a process of observation by our own eyes.

4.5 Application of Queuing Theory on the Research

In order for us to make a queuing experience optimize, we need two queuing disciplines that we shall put in practices for the development of the system to be effective, efficient and optimal.

The queuing discipline used was:

- Shortest Processing Time First (With Update)
- First In First Out (With Update)

We shall integrate this queuing disciplines in a Queuing Management System that we shall develop in other for the system to be optimize for checkout or self-checkout.
We shall also use only two of the Queuing Waiting Line Model which will be:

- Single Channel, Single Phase
- Multi Channel, Single Phase

With both the Queuing Disciplines and Queuing Waiting Line Model, the feasibility for this system to be optimize is high (shall be demonstrated in our findings and discussion).

4.6 Application of the Microservices Architecture on the Research

Applying the Microservices architecture on our system, it will help our system to be more scalable and adaptable since we shall be dealing with many different types of services and retail markets. Hence Scalability and Adaptability is one of the highest point for this system to be optimal.
We decided to use the microservice architecture because it provides us with this two pattern.

- Small and Focused
- Bounded Context

Due to this architecture it will permit use to Decentralized our Data Management and hence each retail market with know nothing about another market. And since microservices architecture are also build for failure management, using the Circuit Breaker and Bulkheads it will help us to identify exactly which microservice is not working and also not to shutdown the whole system since each microservice is independent on it own.

4.7 Application of Software Design Theory on the Research

Applying Software Design on our system, it will helps us to defined functionality, non-functionality, steps and process for the development of the system and also to establish a well define Activity and Deployment Diagrams.

Hence using frameworks and standard as Scrum Framework and Unified Modelling Language (UML) respectively to design our system will facilitate the development of the software.
The Scrum Framework helps us to iterate every week the on the requirement we had Done, We just Did and last what shall be Done. While with UML it helps us to draw:

- Data Flow Diagram
- Use Case
- Activity
- Sequence and
- Deployment

All this help us in achieving a good requirement engineering gathering.

4.8 Application of Qualitative Research Method on the Research

Our Qualitative research was done using 3 methods as stated above

- Informal One-to-One Interview
- Record Keeping
- Process of Observation

The informal One-to-One Interview the data collected can be seen in the appendix, The Process of Observation was done by our own self which was to observe queues in retail markets this last 3 months both in Rwanda and in Norway. Lastly, the Record Keeping was implemented by keeping the record of every interaction we had with our the participants that accepted to be interview with us.
Our research was a Convert Research we had to put our participants in confidence (confidentiality) and ensure them to have Trust in us, since their private data will not be used shared to someone else and will be destroyed immediately after usage.

Chapter 5

Findings and Discussion

After completing our qualitative research, we have identified some key themes within the research, we used it to organise our data that we collected and to come out with Requirements.
To make sure that the software solution correctly solves the particular research problem, we first correctly understood our research questions. In our introduction chapter We have discover, understood, formulate and analyse on what research problem we shall solve, why such a problem needs to be solved and who (audience) this research will help.
Broadly, this is how our requirement engineering gathering has been done and more details below.

5.1 Requirement Engineering Gathering

A requirement is simply a statement of what the system must do or what characteristics it needs to have.
During the development of the software the requirements needed was:

- User Requirement (what the user need to do)
- Functional requirements (what the software should do)
- Non-Functional requirements (characteristics the system should have)

5.1.1 User Requirements

Here we describe the main requirement the users need the system to do.
General Case:

- He will have the ability to scan barcode of products.
- He will have the ability to Place Order
- He will have the ability to pay after checkout using the payment gateway provided by the system.

Case where the system is working with the Retail Complex or Store:

- He will have the ability to checkout with his client ID
- He will have the ability of scanning QR Code at the entrance of the retail complex.

Case where the system does not work with the retail complex or store:

- He will have the ability to save barcode images and then use it at the checkout to scan previously registered barcode without removing any products from his trolley.

5.1.2 Functional Requirement

Here we describe the main functionality of the system:
Main Functions:

- Barcode Scanning and Capturing
- Checkout Process

Sub Functions:

- Barcode Scanning (if the system works with the Retail Environment)
- Barcode Captiring (if the system is not working with the Retail Environment)
- Payment Gateway (PayPal and Vipps)
- Checkout process (Queuing Management System will be implemented)

Data Flow Diagram

Stated by (Von Rosing et al., 2014), A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. Often it is a preliminary step used to create an overview of the system that can later be elaborated.
From (Von Rosing et al., 2014),It is common practice to draw the context-level data flow diagram first, which shows the interaction between the system and external agents that act as data sources and data sinks. This helps to create an accurate drawing in the context diagram. The system's interactions with the outside world are modelled purely in terms of data flows across the system boundary. The context diagram shows the entire system as a single process and gives no clues as to its internal organization.
Our context-level DFD was then "exploded" to produce a Level 1 DFD that shows some of the detail of the system being modeled. The Level 1 DFD shows how the system is divided into subsystems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole.
It also identifies internal data stores that must be present in order for the system to do its job and shows the flow of data between the various parts of the system (Von Rosing et al., 2014),.
In our system we had 2 data flow diagrams which are mainly:

- Context Diagram
- Level 1 DFD Diagram

The context diagram shows the main data flow of the system, while the Level 1 DFD shows the inner data flow diagram of the system. OD ==> Optimization and Disruption

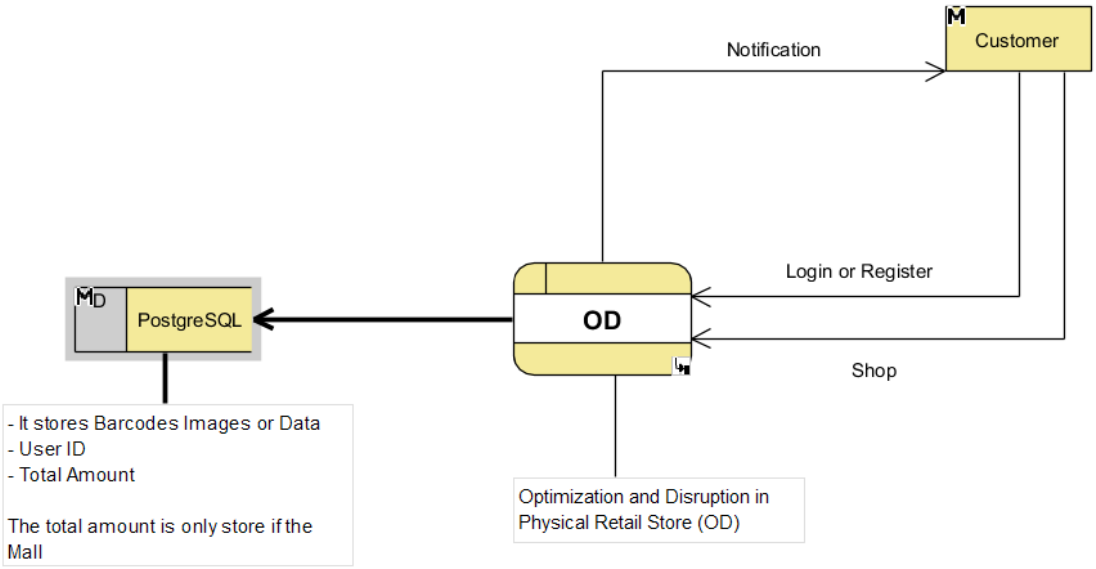

Figure 5.1: Context Diagram

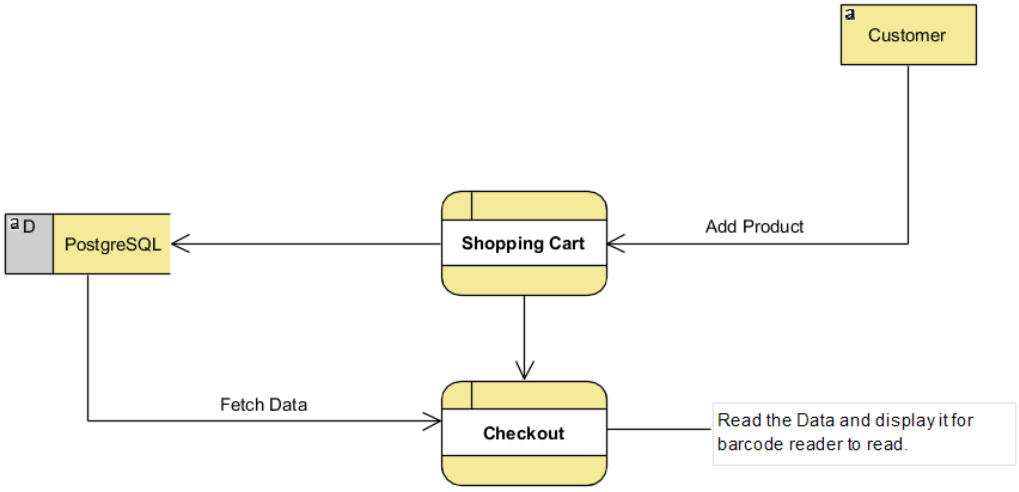

Figure 5.2: Level 1 DFD

User Case Diagram

A use case is a list of actions or event steps typically defining the interactions between a role (actor) and a system to achieve a goal. The actor can be a human or an external entity.

Optimization and Disruption in Physical Retail Environment		
Actors	Use Case (Actions)	Description
Visitor	Register	Register with Username or By Scanning the QR Code Once + Username.
Visitor	Login ==> Pre-condition (Register)	Scan QR Code at Retail Complex Entrance or Login with Username.
Customer	Scan Product ==> Pre-condition (Login)	Customer has to be login before been able to scan barcode of products.
Customer	Place Order ==> Pre-condition (Scan Products)	Customer needs to have products in his cart before placing an order.
System	Auto Creation of the QR Code for the Retail Complex	Automatically created by the system used for authentication at entrance of Retail Complex.

Table 5.1: Use Case Table

Below we shall illustrate the path we shall go around to make our Use Case effective.

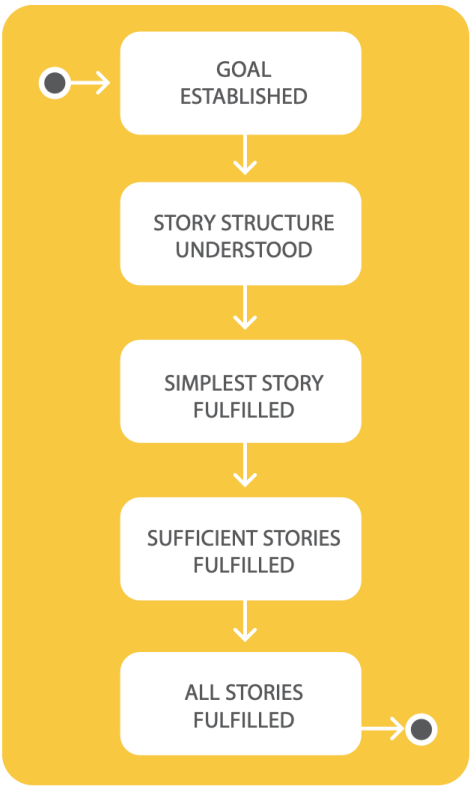

Figure 5.3: Use Case Path by (Jacobson et al., 2016)

Activity Diagram

Activity diagrams are graphical representations of workflows of step-wise activities and actions with support for choice and iteration (McGraw-hill, 2008). They are also intended to model the data flows intersecting with the related activities (J. Rumbaugh & Booch, 1999) (OMG, 2001).
It helps to visualize a certain use case at a more detailed level hence we had activities. The system has 3 activity diagrams which are:

- Initial Activity
- Shopping Activity: Case A
- Shopping Activity: Case B

ODSS => Optimization and Disruption of Shopping System

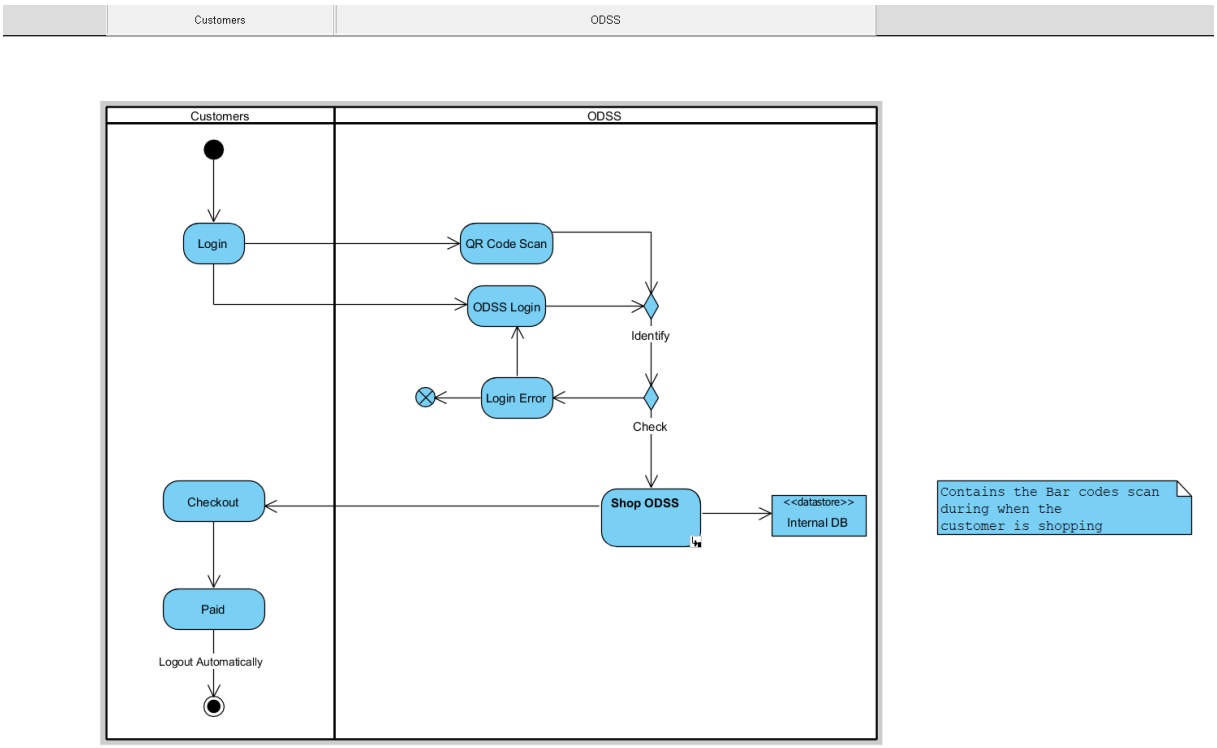

Figure 5.4: Initial Activity Diagram

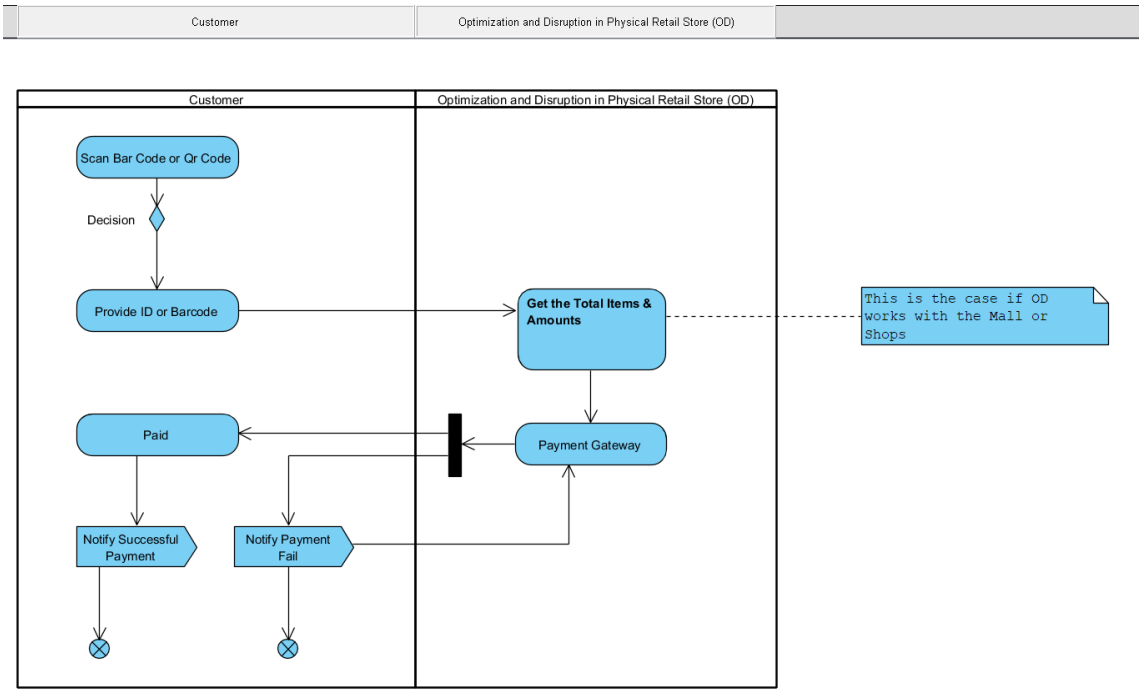

Figure 5.5: Shop Activity Diagram: Case A

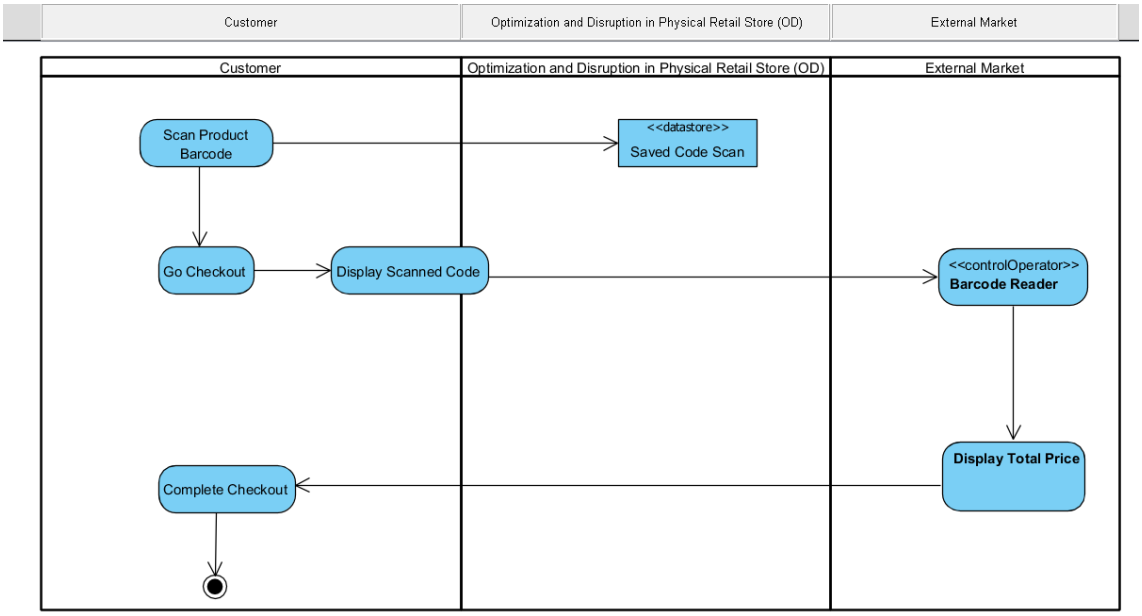

Figure 5.6: Shop Activity Diagram: Case B

5.1.3 Non-Functional Requirement

A non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors.

Performance

During the building of system we 1st took into consideration the performance of the system since the software will be online and will be access by many people at the same time hence we emphasize on its performance.
We did research upon:

- The response time for the application to act upon a user's action such as navigation
- The response time it takes to finish a specific amount of work.
- The load of the system, which is measured as the volume of transactions that the application has to process, such as requests (requests per second), transactions (transactions per second) and pages per second.

Integration

It will be built in way in which it will be flexible so as allow other application to use its services.
Also, the application shall use PayPal and Vipps services as third party services (For Payment Options) as well as being able to integrate other APIs such as Google maps.

Security

The system should be highly secure since only authenticated users gain access to the System. It should respect the CIA which is:
Confidentiality: The system shall ensure confidentiality to prevent sensitive information from reaching the wrong people, while making sure that the right people can in fact get it: Access must be restricted to those unauthorized to view the data.
JWT which will be found in the QR Code placed in front of Retail Store or Complex. Since we used Spring Boot we took advantage of it security approach already implemented which is Okta Security Framework (Authentication and Authorization Service).
Integrity: The system also involves in maintaining the consistency, accuracy, and trustworthiness of data. Only authorized actors shall be able to modify data using authorized operations.
Availability: The system data shall always be available since it is an online system. We shall do auto backup of our data each 24hrs in other to precaution if the is an incident on the central server.

Maintainability

The system will be easily maintainable to allow for additional upgrades that may need to be implemented in the future, it will be designed in such a way that they can be serviced and maintained on a periodic basis.

Portability

The system will be developed mostly as a Web and Mobile app (PWA). Progressive Web App which will permit the mobile version of the website to be installed on any device without any problem.

5.2 Design Phase (Design Specification)

From the theory above we understood that a software design can be said to be the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements.

Now here the Design Phase is where the logical analysis done before is now converted to Physical design. The design phase provides overall guidance on system functions, performance requirements, security requirements, and platform characteristics The design phase in this research comprises of:

- Architectural Design (Architectural Pattern and Software Architecture)
- Process Design (User Interface, App-to-App and App-to-Database)

5.2.1 Architectural Design

The architectural design of a system emphasizes the design of the system architecture that describes the structure, behaviour and more views of that system and analysis. In our system we divided our Architectural Design as:

- Architectural Pattern
- Software Architecture

5.2.2 Architectural Pattern

Stated by (Taylor R.N. & Wiley, 2009) An architectural pattern is a general, reusable solution to a commonly occurring problem in software architecture within a given context. The use of the word "pattern" in the software industry was influenced by similar concepts as expressed in traditional architecture. From these we decided to us the Microservices Architecture as stated above in this research.

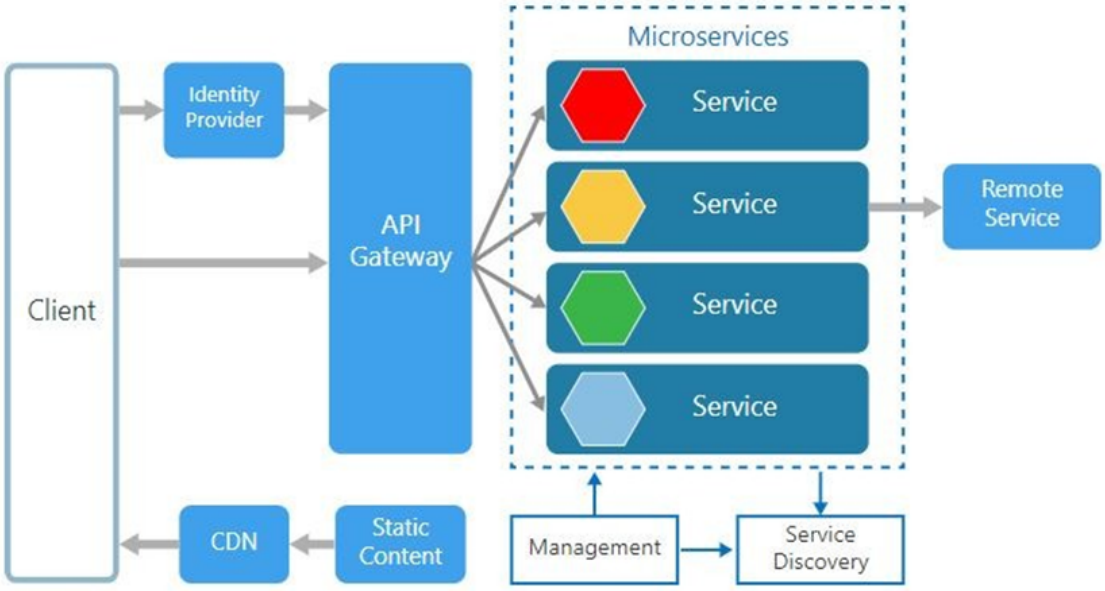

Figure 5.7: Microservices Architecture by (futurefu, 2019)

5.2.3 Process Design

The process design is to determine the workflow of information in the system and implementation requirement the system.

Sequence Diagram A sequence diagram shows the interactions between actors and part (module) of the system arranged in time sequence. It depicts the sequence of messages exchanged between the actors and the module needed to carry out the functionality of the scenario. Here we had two sequence diagrams which are:

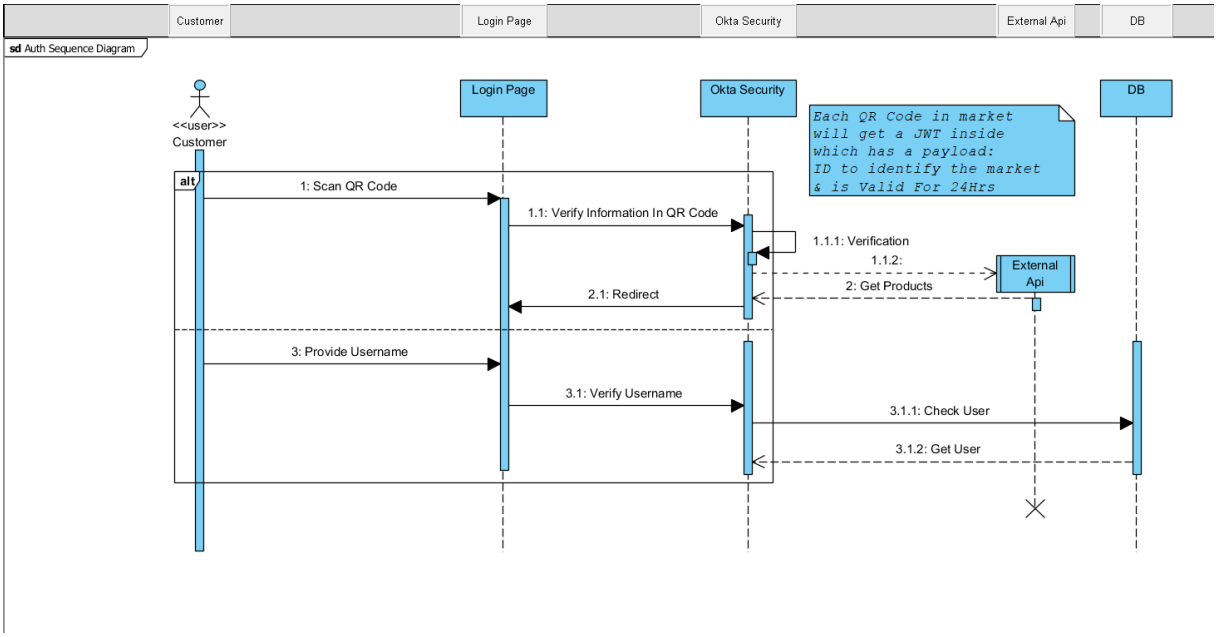

Figure 5.8: Auth Sequence Diagram

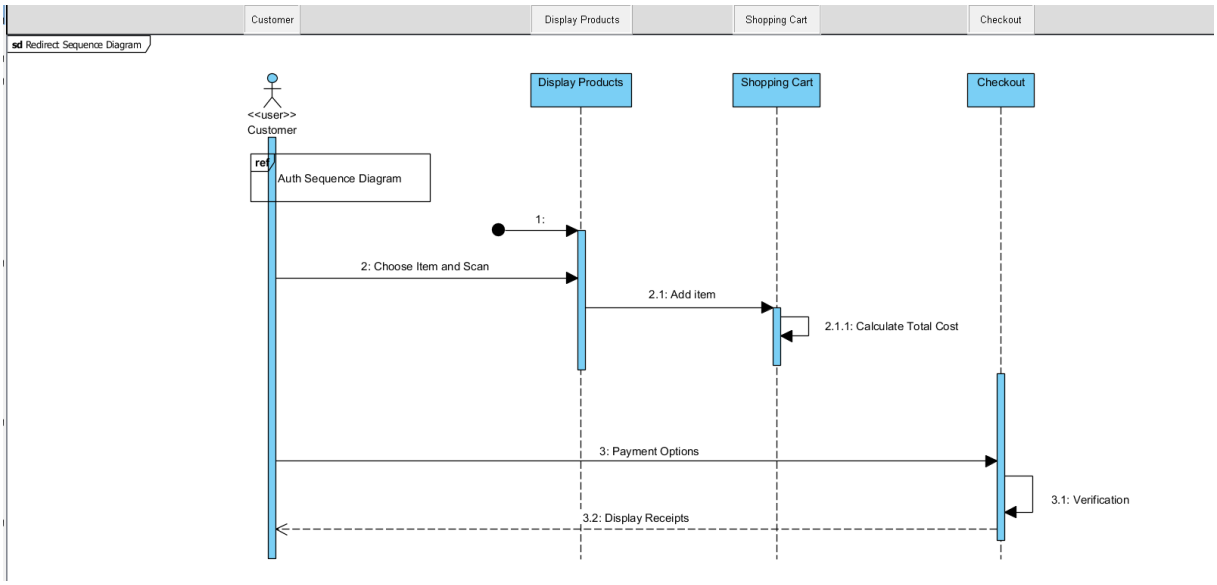

Figure 5.9: Shopping Sequence Diagram

Alt: means there is an alternative condition, the first one is when the condition is true while the second is when else condition.
Ref: It is process to call another sequence diagram. Here our diagrams call the login sequence to verify if the users have successfully been connected to the application.

Deployment Phase

The objective of the Deployment Phase is to know how the system will be deploy and running om end-user devices and to bring it into operation.
During this phase, all the micro-services of the system are deployed onto an AWS Server. After deploying the system, the end users can install the PWA (Progressive Web Application) on their phone.
Main topics are:
- Interaction of the mobile application and server
- Backup Performance

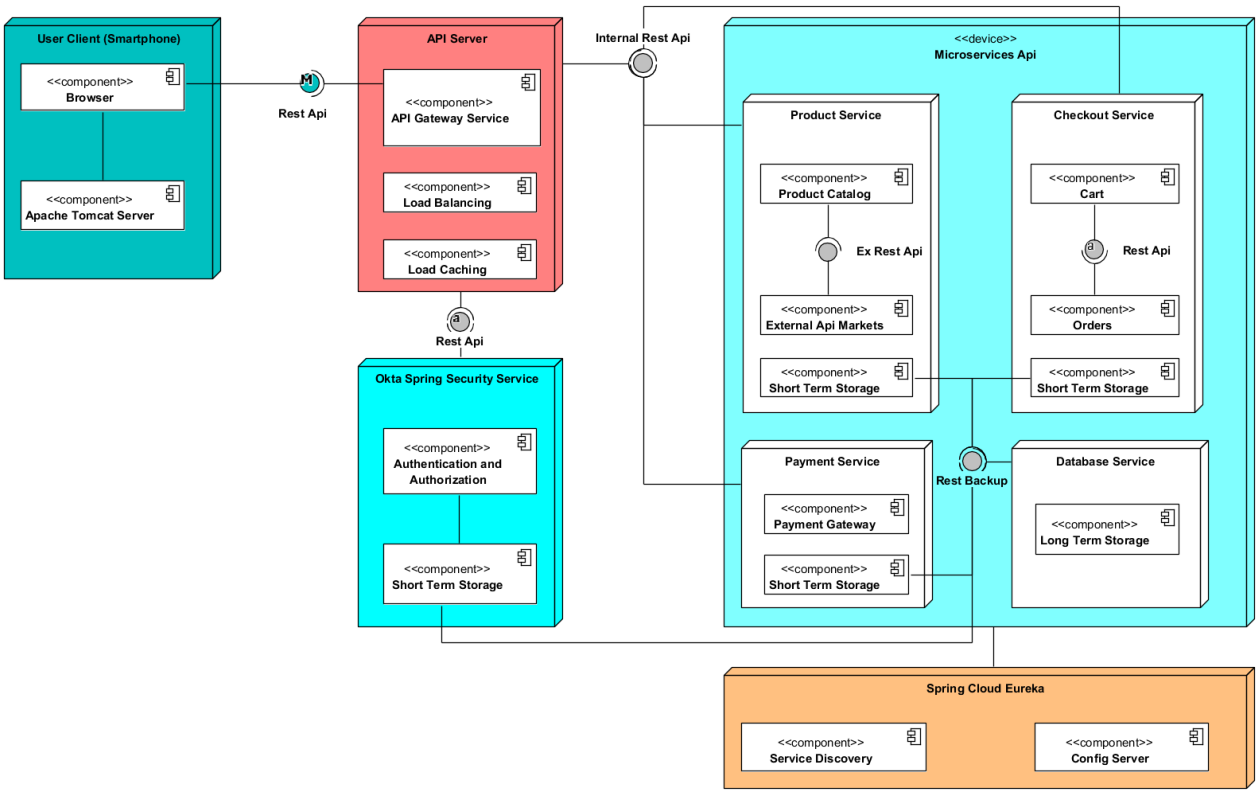

Figure 5.10: Deployment Diagram

5.2.4 Discussion

Lastly not the least we can now answer our research questions
Main Question (MQ):
Could it be that a general system supporting shopping experience for every retail store or complex would be beneficial for solving queue and social distancing in retail stores or complex?
Yes it is possible of having a general system which will integrate all the retail markets into its system and it will be of a big advantage for retail markets who will stress less on managing their queues and focus more on their services.

SQ1: How will the system solve the queuing problem at retail store or complex?
The system will solve the queuing system by applying Two queuing disciplines:

- Shortest Processing Time First (With Update)
- First In First Out (With Update)

and We shall also use only two of the Queuing Waiting Line Model which will be:

- Single Channel, Single Phase
- Multi Channel, Single Phase

SQ2: How will the system communicate with all the different types of retail markets(Shopping centers or stores)
The system will communicate with all the other retail markets through the microservices architecture that will facilitate the integration of other retail markets.

Chapter 6

Conclusions and Recommendation

6.1 Conclusion

In conclusion, The research on the Optimization and Disruption of Physical Retail Shopping Environment (Store or Complex) will be of a big advantage both for the Retailers and also the customer which don't want to spend time in queues for checkout.
In previous studies we discovered, already other system doing the same as Amazon Go or Alibaba Hema which has push many of their customer to them as stated above.
The main aims of our system was to solve Optimize Queue in Retail Environment both in Rwanda and Norway since many people are still doing their shopping physically hence we did research on 4 theory which help us through out our research. Namely:

1. Queuing Theory
2. Microservices Theory
3. Software Design Theory
4. Qualitative Research Theory

6.2 Recommendation

The research was carried out during a period of 5 months and hence this is one of the first aspect we shall point out, having a more depth research for a 6 months more may increase the accuracy of the result we had here.
Secondly we also suggest that a complete requirements on the Non functional requirement should be done in depth in other to reduce the chance the system been over loaded if more millions of users shall be using the system.

Lastly, a future work may be able to carry out a full implementation of the system and doing research not just on Rwanda and Norway but all over the globe.

Appendix A

Data sheets

A.1 Interview Questions

1. Approximately how many clients can you have per day?
2. Which period of the month are they most customers?
    - First Week
    - Second Week
    - Third Week
    - Last Week
3. How many minutes can a customer do while serving him during highest period? (2, 3, 5, 10, 15)
    - 2
    - 3
    - 5
    - 10
    - 15
4. Do customers usually get angry staying in queue?
    - Yes
    - No
5. Do you think prioritizing customer request will be a good idea?
    - Yes
    - No

A.2 Responses of Informal One to One Interview

Meaning of some data:
NaN ==> No Idea .
PRV ==> Not Allowed to Answer.

A.2.1 Simba Kicukiro Response

Start Date ==> 29-03-2022
Answers

Record	Q1	Q2	Q3	Q4	Q5	Cashier
1st 31-03-2022	PRV	Last & First Week	3 or 5	No	No	Private#1
2nd 02-04-2022	PRV	Last & First Week	5	No	No	Private#1
3rd 06-04-2022	NaN	Last Week	3	No	NaN	Private#2
4th 16-04-2022	NaN	Last Week	5	No	NaN	Private#2
5th 20-04-2022	PRV	Last & First Week	5	No	No	Private#1

Table A.1: Simba Kicukiro Kigali Retail Store

A.2.2 T2000 Kigali Response

Start Date ==> 23-04-2022
Answers

Record	Q1	Q2	Q3	Q4	Q5	Cashier
1st 23-04-2022	>80	Last & First Week	5 or 10	Yes	Yes	Private#1
2nd 29-04-2022	>100	Last & First Week	5 or 10	Yes	Yes	Private#2
3rd 04-05-2022	>100	Last & First Week	5 or 10	NaN	NaN	Private#3
4th 07-05-2022	>50	Last & First Week	5 or 10	Yes	NaN	Private#4
5th 13-05-2022	>30	Last & First Week	3 or 5	Yes	Yes	Private#5
6th 14-05-2022	>50	Last & First Week	5 or 10	Yes	Yes	Private#2

Table A.2: T2000 Kigali Retail Store

A.2.3 Rema Grimstad Response

Start Date ==> 02-04-2022
Answers

Record	Q1	Q2	Q3	Q4	Q5	Cashier
1st 03-04-2022	PRV	Second & Third Week	PRV	No	No	Private#1
2nd 09-04-2022	PRV	Second & Third Week	PRV	No	No	Private#1
3rd 16-04-2022	PRV	Second & Third Week	PRV	No	Yes	Private#2
4th 23-04-2022	PRV	Second & Third Week	NaN	No	Yes	Private#3
5th 30-04-2022	PRV	Second & Third Week	PRV	No	Yes	Private#2
6th 04-05-2022	PRV	Second & Third Week	PRV	No	No	Private#1
7th 07-05-2022	PRV	Second & Third Week	PRV	No	Yes	Private#2
8th 13-05-2022	PRV	Second & Third Week	NaN	No	Yes	Private#3
9th 14-05-2022	PRV	Second & Third Week	NaN	No	Yes	Private#3

Table A.3: Rema Grimstad Adger Retail Store

Bibliography

Agresti, W. (1986). The conventional software life-cycle model: Its evolution and assumptions. *in New Paradigms for Software Development*.

Angus, I. (2019). Sf 'cashless ban' faq: How to pay cash at amazon go, and more questions answered. https://www.sfgate.com/news/article/Cashless-ban-FAQ-food-truck-amazon-go-no-cash-14373909.php.

Bizclik. (2020). Rwanda tops first african retail development index. https://businesschief.eu/leadership-and-strategy/rwanda-tops-first-african-retail-development-index-1.

Bizimungu, J. (2015). What growth of online shopping means for the local economy. https://www.newtimes.co.rw/section/read/194868.

Boehm, B. (1976). Software engineerings. *IEEE Transactions on Computers*, *c-25*(12), 1226–1241.

Bosa, D. (2018). Amazon's automated grocery store will launch monday after a year of false starts. https://www.cnbc.com/2018/01/21/amazon-go-automated-grocery-store-is-poised-to-launch.html.

Brown, D. (2018). Dash cart: Amazon's smart shopping cart knows what you're getting, displays your subtotal. https://eu.usatoday.com/story/tech/2020/07/14/dash-cart-amazons-smart-shopping-cart-knows-what-youre-getting/5433528002/.

Day, M. (2018). Amazon go cashierless convenience store opens to the public in seattle. https://www.seattletimes.com/business/amazon/amazon-go-cashierless-convenience-store-opening-to-the-public/.

Daya, S., Van Duy, N., Eati, K., Ferreira, C. M., Glozic, D., Gucer, V., Gupta, M., Joshi, S., Lampkin, V., Martins, M., et al. (2016). *Microservices from theory to practice: Creating applications in ibm bluemix using the microservices approach*. IBM Redbooks.

Denzin, N. K., & Lincoln, Y. S. (2011). *The sage handbook of qualitative research*. sage.

Dmitry, N., & Manfred, S.-S. (2014). On micro-services architecture. *International Journal of Open Information Technologies*, *2*(9), 24–27.

Doctoroff, T. (2018). Alibaba group's hema supermarkets: The real deal with china's 'new retail. https://adage.com/article/cmo-strategy-columns/alibaba-group-s-hema-supermarkets-real-deal-china-s-retail/312345.

Eiendom, A. (2019). The norwegian retail market. https://akershuseiendom.no/markedsinnsikt/artikler/the-norwegian-retail-market.

Fine, G. A. (1993). Ten lies of ethnography: Moral dilemmas of field research. *Journal of contemporary ethnography*, *22*(3), 267–294.

futurefu. (2019). Ai & machine learning, devops, django, internet of things, python, technology update news. https://www.futurefundamentals.com/what-is-microservices-architecture/.

Garun, N. (2016). Amazon just launched a cashier-free convenience store. https://www.theverge.com/2016/12/5/13842592/amazon-go-new-cashier-less-convenience-store.

Gomes, L. (2018). Amazon go has competition: Meet 6 other automated stores transforming retail. https://thecurrentdaily.com/2018/10/24/amazon-go-competition-automated-stores/.

Gos, K., & Zabierowski, W. (2020). The comparison of microservice and monolithic architecture. *2020 IEEE XVIth International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH)*, 150–153.

Heijn, A. (2006). Wiki albert heijn. https://en.wikipedia.org/wiki/Albert_Heijn.

IBM. (2021). In this article, we'll explain the basics of service-oriented architecture (soa) and microservices, touch on their key differences and look at which approach would be best for your situation. https://www.ibm.com/cloud/blog/soa-vs-microservices.

ISO/IEC. (2005). Iso/iec 19501:2005 - information technology - open distributed processing - unified modeling language (uml) version 1.4.3. *Iso.org*.

J. Rumbaugh, I. J., & Booch, G. (1999). The unified modeling language reference manual. *Addison-Wesley*.

Jacobson, I., Spence, I., & Kerr, B. (2016). Use-case 2.0. *Communications of the ACM*, *59*(5), 61–69.

Kelion, L. (2020). Amazon offers till-free shop technology to rivals. https://www.bbc.com/news/technology-51801674.

Ken Schwaber, J. S. (2020). The scrum guide.

Kirill, T. (2022). What is queue theory? guide to queuing theory. https://www.qminder.com/blog/queue-management/queue-theory-guide/.

Kumar, A., & Sharma, G. K. (2013). Queueing theory appr oach with queueing model: A study. *International Journal of Engineering Science Invention*, *2*(2), 1–11.

Malunda, D., Ruzima, M., & Bower, J. (2020). The commercial real estate market in the city of kigali.

Maruping, L. M., Venkatesh, V., & Agarwal, R. (2009). A control theory perspective on agile methodology use and changing user requirements. *Information systems research*, *20*(3), 377–399.

McGraw-hill. (2008). Glossary of key terms. https://highered.mheducation.com/sites/0077110005/student_view0/glossary.html.

McPhee, K. (1996). Design theory and software design.

Medhi, J. (2002). *Stochastic models in queueing theory*. Elsevier.

Medium. (2019). What is service-oriented architecture? https://medium.com/@SoftwareDevelopmentCo
what-is-service-oriented-architecture-fa894d11a7ec.

Myrset, B. O., & Eriksen, K. (2019). Norwegian shopping center survey. *Deloitte report*, (5), 4–5.

OMG. (2001). Omg unified modeling language specification, version 1.4. *UML Revision Task Force*.

QueueIT. (2020). Queuing theory: Definition, history & real-life applications. https://queue-it.com/blog/queuing-theory/.

Research, & Markets. (2021). Food and grocery retail in norway - market summary, competitive analysis and forecast to 2025, 36. https://www.researchandmarkets.com/reports/5322877/food-and-grocery-retail-in-norway-market.

Research, & Markets. (2022). Food & grocery retail market size, share & trends analysis report by product (food cupboard, beverages), by distribution channel (supermarkets & hypermarkets, online), by region (apac, europe), and segment forecasts, 2022 - 2030, 80. https://www.researchandmarkets.com/reports/4613444/food-and-grocery-retail-market-size-share-and.

Rogiest, W., Laevens, K., Walraevens, J., & Bruneel, H. (2015). Random-order-of-service for heterogeneous customers: Waiting time analysis. *Annals of Operations Research*, *226*(1), 527–550.

Royce, W. (1970). Managing the development of large software systems.

Ryen, A. et al. (2004). Ethical issues. *Qualitative research practice*, 230–247.

Schrage, L. E., & Miller, L. W. (1966). The queue m/g/1 with the shortest remaining processing time discipline. *Operations Research*, *14*(4), 670–684.

Silverman, D. (2020). *Qualitative research*. sage.

Sprott, D., & Wilkes, L. (2004). Understanding service-oriented architecture. *The Architecture Journal*, *1*(1), 10–17.

Taylor R.N., D., Medvidović, & Wiley. (2009). Software architecture: Foundations, theory and practices.

Thompson, E. (2018). Amazon's 1st high-tech grocery store opens to the public. https://www.cbc.ca/news/science/amazon-go-grocery-store-1.4497862.

Tsernov, K. (2022). What is queue management? the definitive guide to queuing systems. https://www.qminder.com/blog/queue-management/what-is-queue-management-system/.

Von Rosing, M., Von Scheel, H., & Scheer, A.-W. (2014). *The complete business process handbook: Body of knowledge from process modeling to bpm, volume 1* (Vol. 1). Morgan Kaufmann.

WikipediaFresh. (2022). Amazon fresh. https://en.wikipedia.org/wiki/Amazon_Fresh.

Wingfield, N. (2016). Amazon moves to cut checkout line, promoting a grab-and-go experience. https://www.nytimes.com/2016/12/05/technology/amazon-moves-to-cut-checkout-line-promoting-a-grab-and-go-experience.html.

WorkplaceTesting. (2018). Observation procedure. https://www.workplacetesting.com/definition/2512/observation-procedure.

Worldline. (2020). How queues kill retail stores. https://www.bambora.com/articles/queues-trend/.