|  |
| --- |
|  |
|  |
|  |
| [Architecture Document for Elektronika]  For all your electronic needs |
| **Developed by Group 13** |
| Anusha Vaddavalli [h19anuva@du.se](mailto:h19anuva@du.se) |
|  |
|  |
|  |

Document Control

**Version History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Key changes** |
| 01 | 2021-02-01 | Yasir, Gautham | Solution Overview |
| 02 | 2021-02-25 | Yasir, Gautham, Anusha, Vishnu | Business Context, Conceptual Architecture and Solution Architecture |
| 03 | 2021-03-08 | Yasir | Delivery Considerations |
| 04 | 2021-03-15 | Gautham, Anusha, Vishnu | Introduction |

**Document References**

|  |  |  |
| --- | --- | --- |
| **#** | **Title** | **Link** |
| 1 | Document Template | <http://integture.com.au/blog/what-is-sad/> |
| 2 | Details for Documentation | <http://www.michael-richardson.com/processes/rup_for_sqa/core.base_rup/guidances/guidelines/software_architecture_document_F4C93435.html> |
| 3 | Agile Project Plan and Agile Project Backlog Template | <https://www.smartsheet.com/agile-project-management-excel-templates> |

Contents

[1. Introduction 4](#_Toc442197630)

[1.1 Document Purpose 4](#_Toc442197631)

[1.2 Document Scope 4](#_Toc442197632)

[1.3 Document Audience 4](#_Toc442197633)

[1.4 Definitions 5](#_Toc442197634)

[2. Solution Overview 6](#_Toc442197635)

[2.1 Solution Objectives 6](#_Toc442197636)

[2.2 Solution Scope 7](#_Toc442197637)

[2.2.1 In Scope 7](#_Toc442197638)

[2.2.2 Out of Scope 8](#_Toc442197639)

[2.3 Assumptions 8](#_Toc442197640)

[2.4 Constraints 8](#_Toc442197641)

[2.5 Dependencies 9](#_Toc442197642)

[2.6 Key Architecture Decisions 10](#_Toc442197643)

[3. Business Context 11](#_Toc442197644)

[3.1 Business Capabilities 11](#_Toc442197645)

[3.2 Key Business Requirements 11](#_Toc442197646)

[3.3 Non Functional Requirements 12](#_Toc442197649)

[4. Conceptual Solution Overview 13](#_Toc442197650)

[4.1 Conceptual Architecture 13](#_Toc442197651)

[5. Solution Architecture 19](#_Toc442197652)

[5.1 Information Architecture 1](#_Toc442197653)9

[5.2 Application Architecture 20](#_Toc442197655)

[5.3 Integration Architecture 21](#_Toc442197657)

[5.4 Infrastructure Architecture 21](#_Toc442197659)

[5.5 Security Architecture 2](#_Toc442197661)2

6[. Solution Delivery Considerations 22](#_Toc442197670)

[6.1 Development Considerations 2](#_Toc442197671)2

[6.2 Deployment Considerations 23](#_Toc442197672)

[6.3 Data Migration Considerations 24](#_Toc442197673)

[7. References 24](#_Toc442197675)

# Introduction

## Document Purpose

The usage of the document is to make understand the goals, design and implementation of the case to the stakeholders. Stakeholders for Electronika are administrator who register the items to the site, Customer who use the system to purchase items and vendors who sell the products. The system is designed with user friendly front end interface for both Customers and administrator. Here, customer can be either casual shopper or logged in user where he can login using user registration or Google login. The detailed user stories and architecture are further explained in subsequent sections.

Elektronika is an E-Commerce solution that provides an online shopping platform to users that are interested in buying electronic equipment. It is possible for the users to browse through the products based on categories, search for products, view and write reviews for different products, include products in their shopping cart, manage their shopping cart and even make payments.

The business aspect of an E-Commerce website is simply buying and selling of products over an online platform. Elektronika follows both business models, business to business (B2B) and business to consumer (B2C). Pure business context is not part of our project scope, however Elektronika has a lot of business possibilities.

Like every project, Elektronika also faced constraints that we had to overcome to implement the solution. We faced both technical and non-technical constraints. Some of them directly affected the development of Elektronika and consumed time to overcome. These constraints will be discussed in detail in later sections.

Conceptual solution to the Electronika is described using Use case diagram, Activity diagram and data model diagram. Here use case diagram explains the features of the system from perspectives of stakeholders and their interaction to the use cases. Activity diagram shows the flow of execution of use cases and event of interactions between system and stakeholders. Data model explain how data structure is modelled to store into database and how it mapped.

Logical solution of Electronika is explained using component diagram of architectural representation of the system by visualizing the data storage layer, data service layer, business layer, and presentation layer and interactions among them.

## Document Scope

The document identifies the purpose, limitations, functionality, and future recommendations for Electronika. The documents also include all milestones that we achieved during the designing and development phases. It does not have details for how each sprint was planned and executed. And the document also does not mention the tasks done by each team member.

## Document Audience

The audience for this document includes:

* **Project Team:** [*Yasir Imam, Anusha Vaddavalli, Gautham Gaddam, Vishnu Koneshvaran*]
* **Instructor:** [*Pär Eriksson*]

## Definitions

The following acronyms and definitions will be used in the [*project*] and within this document.

| **Term/Acronym** | **Definition** |
| --- | --- |
| E-Commerce | Ecommerce, also known as electronic commerce or internet commerce, is the purchasing and sale of goods and services over the internet, as well as money and data transfers used to complete these transactions. |
| B2B Model | Business to business electronic commerce, is a form of transaction made between businesses, such as manufacturer and wholesaler rather than individual customer. |
| B2C Model | Business to customer commerce, is a process of selling and goods directly from a business to end user customer. |
| Responsive Website | Responsive web design is a web development which take an approach of enables dynamic changes to appearance of website depending upon the actual window size and orientation of the device. |
| Stakeholders | Stakeholder is defined as an individual or group that takes an interest in any decision or activity of an organization. |
| Dashboard | The dashboard is the front page of the administration UI providing convenient management tasks and some server information. |
| Constraints | Something that limits or restricts someone or something. |
| Cyber Laws | Cyber law is the law governing computers and internet involving all aspects of activities and transactions done through the internet. |
| Dependence | It can be defined as a description of an event that involves change or invalidate decisions. |
| React Framework | React is a JavaScript library for designing user interfaces that is declarative, effective, and scalable. It allows you to build complex user interfaces out of tiny, isolated pieces of code known as components. |
| Scalability | It is a property of a system to handle a growing amount of work by adding resources to the system. |
| Reliability | It is a property of the system to be trustworthy or performing consistently well. |

# Solution Overview

## Solution Objectives

Elektronika is an E-Commerce platform that allows companies to sell their goods to customers all over the world through the internet. Ecommerce, as the term implies, encompasses all types of businesses that use the internet to share data and/or money. E-Commerce businesses achieve sustainable growth by lowering their cost-to-customer, expanding their customer base and offering a one-of a kind customer experience. Making proper use of ecommerce has become more than essential for B2C and other businesses. Now, ecommerce has developed into digital commerce, which encompasses the entire business journey from purchase to distribution through an online experience, the following are the few objectives:

1. Reduce management costs:

Businesses strive to lower their costs in order to increase their sales, automating ecommerce company will help save a lot of money on management.

2. UI design considerations:

Some essential user interface design tips:

* Do not overdesign
* Avoid popup windows

Well defined Product Categories:

Grouping products into categories and subcategories that can describe a range of products, so shoppers can take a quick scan through them and instantly understand what they represent.

Product Search:

Building a search function that helps them easily find what they are looking for, such as:

* Make search omnipresent, search box on every page should be visible and quickly recognizable.
* A search needs to support all kinds of queries such as product names, categories, and product attributes.
* Allow sorting and filtering of results.

3. Design for trust and security:

It is important to create a website that customers can trust. Most customers are concerned about their privacy and whether the platform can protect their personal information by allowing them to make a safe transaction. They would opt to shop elsewhere if the website does not feel trustworthy.

Here are the few methods that can communicate trustworthiness:

1. Include an overview of the business: providing general information, contact information, links to social media, FAQ page.
2. Writing in plain language and avoid legal or internal policy jargon.
3. Share product reviews.
4. Use a secure server.
5. Show attention to detail: make the website look legitimate and professional by avoiding typos, missing images, broken links, 404 errors or other eCommerce UX killing mistakes.

4. Providing a unique customer experience:

There are several ecommerce businesses operating in the market. When a customer searches for a product, the first three links that appear on Google Search engine results page are likely to be clicked and rest of them are either avoided or never used. Providing a unique customer experience is one of the easiest ways to stand out from the crowd. Customer service that is available for 24hours a day, 7 days a week and instant responses to customer inquiries are several other factors to consider.

5. Making responsive website:

It is one of the major objectives of all leading ecommerce businesses. The term responsive refers to a website that can be viewed equally well on a variety of devices with different screen sizes. According to studies, Google could rank a website based on its mobile platform. It means that every website with a responsive design would be rated higher than a website without one.

## Solution Scope

### In Scope

The solution scope defines the scope of Elektronika. The scope of the project will tell the reader about what aspects have been covered by the project.

2.2.1.1 Design goals:

Design of the website contains the following things:

Admin Panel

Login

Dashboard

Manage products

Manage customers

Manage reviews

Manage orders

Casual Users

Browse the website

View products by category

Search for products

Add products to cart

View the cart

Modify the cart

Make payment after logging into stripe api

Logged in Users

Login via Google

Rate any product

Write review for any product

View present order

View order history

2.2.1.2 Stakeholders:

Following are the stakeholders:

- Administrator:

Administrator is not only responsible for creating the Elektronika website but managing the website, adding vendors, adding their products and managing the online orders are also part of his work. The administrator will manage products, customers, reviews and orders by using CRUD operations.

- Vendor:

Vendors are the sellers of specific or list of products. They are added to the website along with their products by the administrator.

- Users:

These are the buyers of the products on the website. The users are divided into 2 types, casual users and logged in users. Casual users/customers can browse through the website, search for products, see products by category, see reviews of other customers, add products to their shopping cart and even modify the shopping cart by adding/deleting/emptying. To place the order, the user needs to login to paypal or klarna to confirm the payment. Logged in customers can rate a product and write reviews for different products. They can also view their present order and order history.

### Out of Scope

There are a few sections that are generally covered by most E-Commerce Websites but unfortunately, they were out of scope for Elektronika. These contain the following:

Vendor Controls:

Elektronika website does not allow much controls to Vendors. Everything related to Vendors are being controlled by the administrator.

Order Management:

Elektronika website does not provide any wishlist and it does not maintain end-to-end delivery. The website is only responsible until the finalization of the payment.

Users/Customers:

Elektronika website does not cover customer loyalty programs. The website also does not cover anything related to the privacy policy. Like most online shopping websites, our website does not give any special recommendations to specific customers based on their purchase history.

## Assumptions

During our work, following assumptions were made:

1. We already have registered vendors and as a test, only 5 vendors were used.
2. It was also assumed that we have products to sell.
3. Elektronika website also assumes that the customers will visit the website for online shopping without making any efforts for marketing.

## Constraints

In general, an E-commerce business experiences a lot of constraints and limitations in its functionality and even day to day operations. As developers, we divide the constraints of our solution into 2 basic types:

1. Technical Constraints
2. Non-Technical Constraints

Technical Constraints:

* Choosing a platform: Thousands of templates are available for E-Commerce solutions that require minimum development knowledge. These templates do not only save the owners from technical hassles but also save a lot of development time.
* Time constraint: As developers of a functional E-Commerce website, we only had 6 weeks of development time.
* Equipment constraint: Keeping in mind, rapidly changing technologies, we chose to develop the web solution using react technologies. To use an updated platform, we also required updated systems, which unfortunately, we did not have. We had to use the systems present in the lab.
* Paid plugins: There are many paid plugins available to handle operational tasks like online payment, authentication of logins, handling inventory etc.
* Vendor controls: According to the requirements provided to us, vendors do not have much control over their products but, in general vendors should be provided some control over their products.

Non-Technical Constraints:

Most non-technical constraints are out of our project scope, however we would like to mention them, as they can be helpful for future developers.

* Security: One of the most important constraint of an E-Commerce solution is security. Not only the website data but also secured customer experience is also very important. Customers need to provide sensitive information like credit card details to confirm payments.
* Trust: Most E-Commerce platforms suffer because the consumers or the vendors do not trust the systems.
* Privacy policy: All E-Commerce platforms have a privacy policy that covers against all legal issues that may (or may not) arise.
* Tax issue: E-Commerce platforms also cater global customers from different countries of the world. Tax policies are different for different countries and this may create problems.
* Product delivery: Customers prefer online shopping for a number of reasons and end-to-end hassle free delivery is one of those reasons. Although it can built up customer trust but, end-to-end delivery is hard to be guaranteed.
* Human Resource: Talented and reliable human resource is required for business and it can be expensive.
* High maintenance cost: Development and maintenance of an E-Commerce solution can be costly. Cyber laws and regulations are also not clear and they can vary from country to country.

## Dependencies

Elektronika is dependent on a number of factors, we divide them into two categories:

1. Technical Dependency
2. Non-Technical Dependency

Technical Dependency: These factors are mainly related to the system requirements. Within our solution, we are using micro services for creating the website. Micro services are modeled as isolated units. However, fully functional systems need cooperation and integration between individual parts. Our solution has the following technical dependencies:

1. Importing the required libraries
2. Data sharing between different units
3. Passing the right props in REACT
4. Connecting with the database
5. Maintenance of the system (adding vendors and products is done by the admin)
6. Payment services are taken externally from paypal and klarna. These services need to be available.

Non-Technical Dependency: The solution also has some non-technical dependencies that include the following:

1. Trust of vendors
2. Trust of Users
3. Availability of products
4. Delivery of orders

## Key Architecture Decisions

|  |  |  |
| --- | --- | --- |
| # | Title | Decision |
| *AD1* | *Developing Presentation Layer* | *Create the front-end using react components* |
| *AD2* | *Developing Application Layer* | *Use Zustand framework and code in redux with Javascript and CSS* |
| *AD3* | *Developing Data Layer* | *Use Firebase with noSQL* |

We use most common Three-tier architecture, a client-server based software in which user interface layer, functional process logic layer and data storage/access layers are developed as independent modules.

Presentation/User interface layer:

In this layer, user will interact with Elektronika and its information to search or see all the products/content of the website. In our case, user has to sign in in order to purchase a product. All the requests that are made by user will be sent to application layer where it is processed and results are sent back to user. The main purpose of this layer is for displaying data on the page and handling user interaction such as follows

1. If buyer has an account, then he needs to sign in using credentials. If not, create an account using sign up button
2. Buyer searches for products
3. Buyer searches through categories to choose particular product.
4. Buyer selects a product and click add to cart button to add it to shopping cart
5. Buyer views all the products by clicking cart button, where all the information about products he wants to buy and total amount will be presented.
6. Buyer clicks checkout and proceeds with payment.

Application layer:

In this layer, it contains the business logic and controls the application by processing commands using core funcitons. For Admin, we use Zustand to simply state management. For User, we code in Redux with JavaScript and CSS.

Data layer:

This layer consists of a read and write access to a database, technically referred to as storage tier. We use Firebase with noSQL.

# Business Context

## Business Capabilities

The primary purpose of an E-Commerce website is simply buying and selling of products over an online platform. The business model followed by Elektronika can either be business to business (B2B) or business to consumer (B2C). The business aspect of the E-Commerce website has a lot of possibilities. Some of them covered by our website are as follows:

1. In general, we realize that customers buy products in bulk. The website will provide discounts for bulk purchase.
2. The website will provide secure login for registered customers.
3. The portal will also have 2 payment methods, Paypal and Klarna. (Fake payment – Stripe)
4. Login possible from social media platforms as well.

## Key Business Requirements

In all aspects of must have ecommerce features, here are few business requirements that we include in this project:

* 1. Account login: It will be more convenient for marketing to require shoppers to create an account so that they might not lose the track of the product and have more benefits over the content of the web page.
  2. Enhanced Shopping Cart: Online shoppers sometimes do not proceed to complete payment due to lengthy checkout process. Considering this case, we include a secured single page check out to enable fast and easy checkout process.
  3. Product comparison and user generated reviews: Customers tend to more likely to purchase a product if they know how it worked with others. To make shopping a fruitful experience, a detailed product description and reviews by other customers will be made easily visible to the user.
  4. Detailed Checkout page: Before allowing users to check out, a detailed information of the products, quantity, amount, and contact information will be displayed for the confirmation by the user before finalising the payment.

## Non Functional Requirements

| Requirement # | Requirement Description | NFR Category | Implication / Action Taken |
| --- | --- | --- | --- |
| NFR.001 | Capacity | CAPABILITY |  |
| NFR.002 | Scalability | SCALABILITY |
| NFR.003 | Reliability | RELIABILITY |  |
| NFR.004 | Security | SECURITY |
| NFR.005 | Maintainability | MAINTAINABILITY |  |
| NFR.006 | Serviceability | SERVICEABILITY |  |
| NFR. 007 | Performance | PERFORMANCE |
| NFR.008 | Availability | AVAILABILITY |

# Conceptual Solution Overview

## Conceptual Architecture

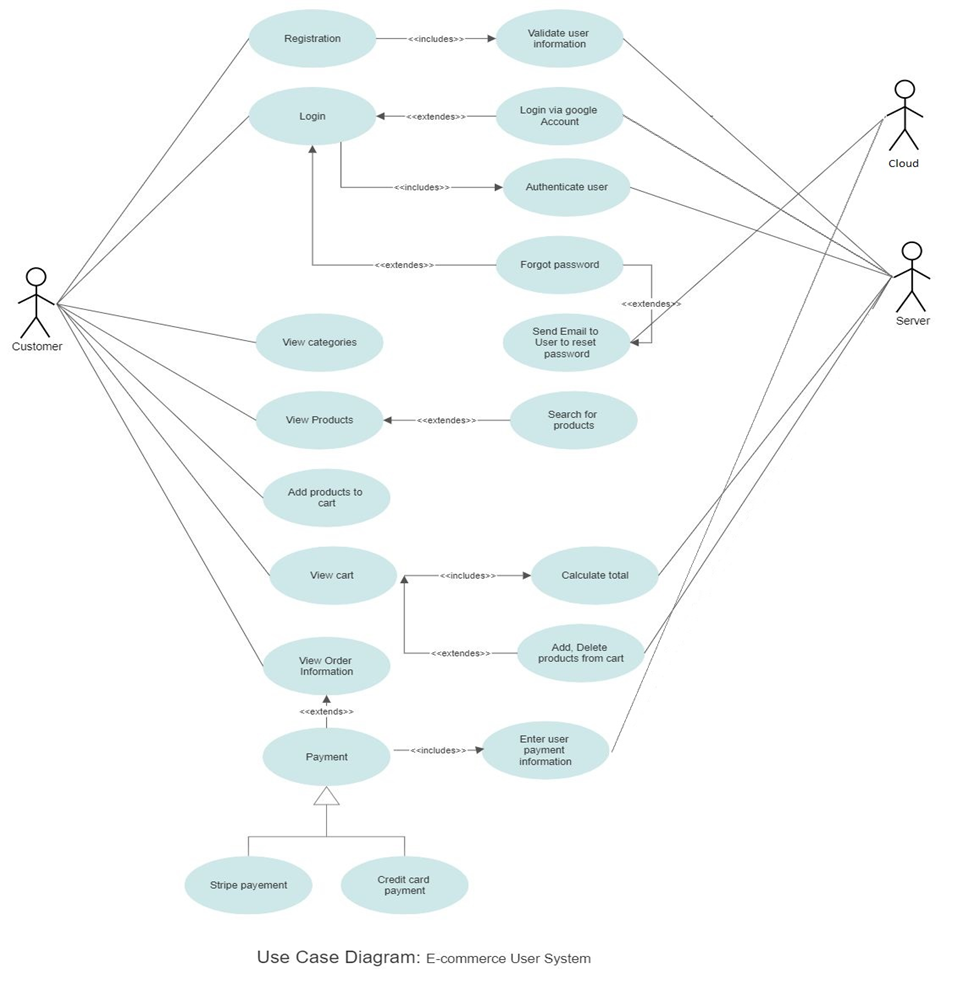
Conceptual modeling of a system visualizes the features and relationships within the domain which specifies the software blueprints. To systematically visualize Elektronika website the following three unified modeling language diagrams are chosen.

Use case Diagram:

A use case diagram consists of the list of actions and its integrations between the system and stakeholders to achieve the goal. The current Electronica website is developed with user specific domain and admin-specific domain and UML diagrams are shown accordingly as follows.

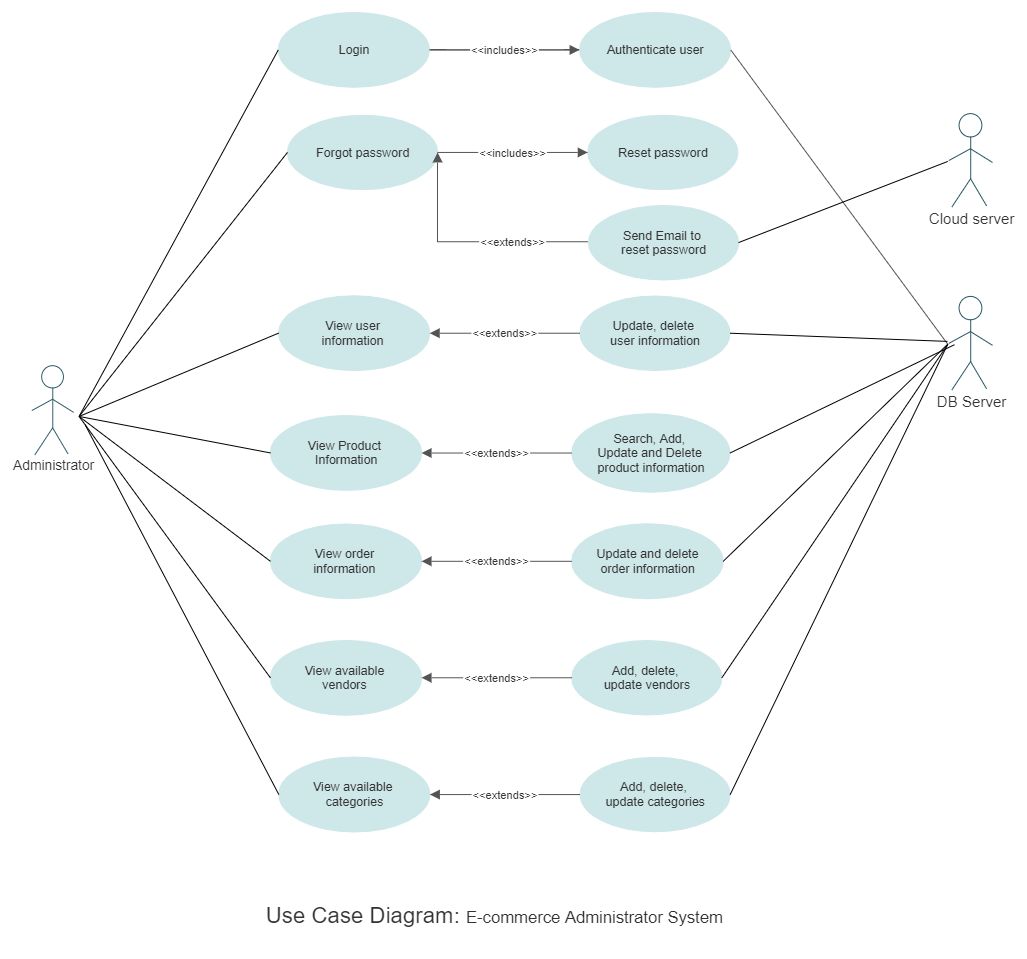
User system:

The user system has two types of customers as active actors who are logged in to customers and guest customers. Further, Electronica communicates with database server (Firebase admin) to store data and cloud server to send email which are passive users. The features of the system are represented as use cases and associations are specified with corresponding actors and use cases.



Administrator System:

The administrator system has admin as an active actor and communicates with database server (Firebase admin) to store data and cloud server to send email which are passive users. The features of the system are represented as use cases and associations are specified with corresponding actors and use cases.



Activity Diagram: An activity diagram explains the sequential execution steps of the use cases by depicting the workflow that explains the behavior of the system.

Activity Diagram of Logged in Customer:The diagram shows all existed sequential flow of events for the user from the system point of view from login to logout operation.

Activity Diagram of the Administrator:The diagram shows all existed sequential flow of events for the admin from the system point of view from login to logout operation.

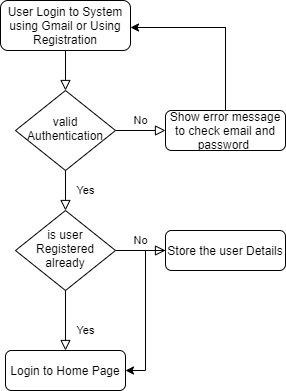


Figure 01: User Login

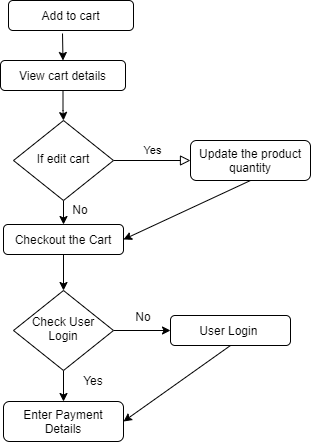


Figure 02: Cart Operations

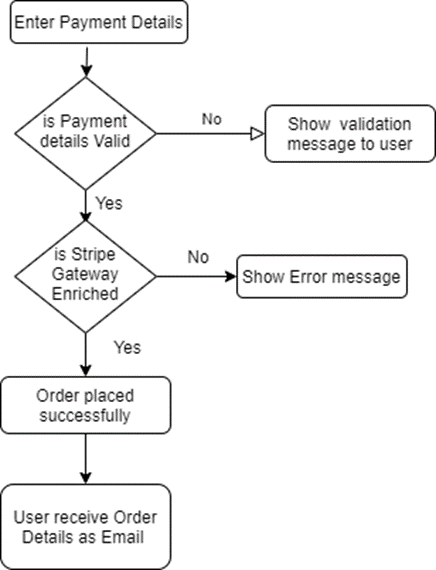


Figure 03: User Payment

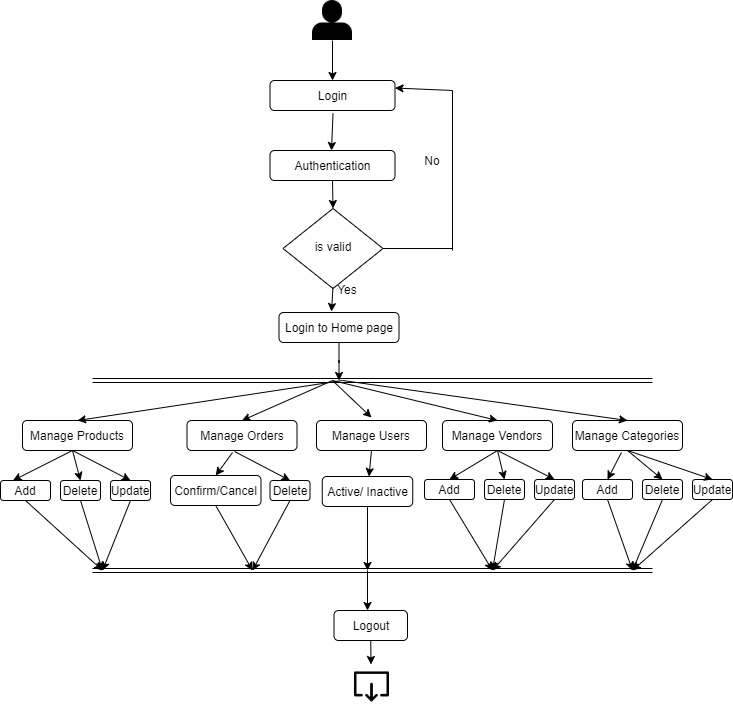


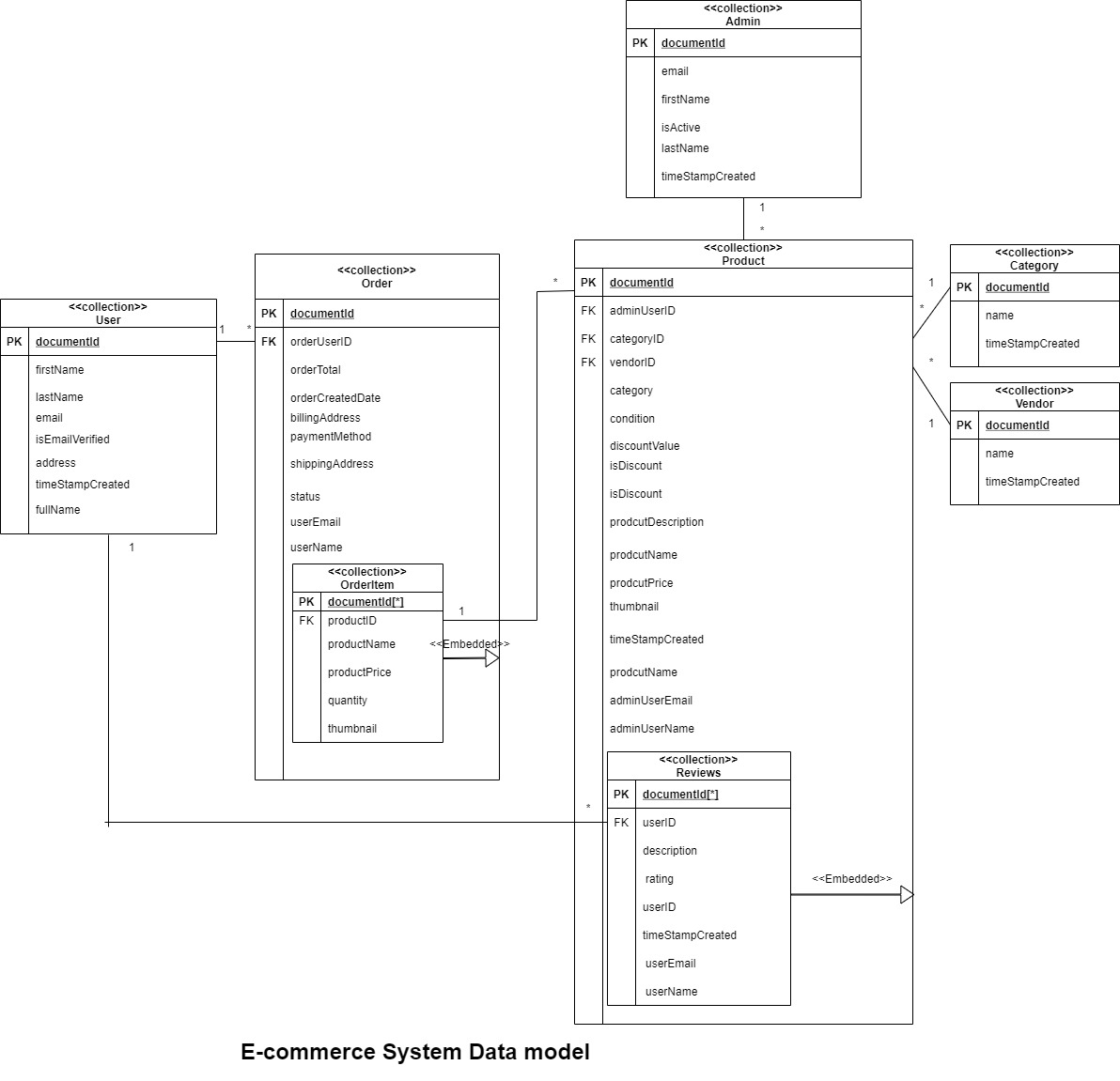
Figure 04: Activity Diagram for Admin

The selected NoSQL database design method using conceptual data model. This design method is currently applied to the e-commerce business area of Electronika. General consideration of NoSQL databases is only to concentrate on solving non-functional requirements to run well on large clusters like performance, availability, scalability and so on.

The database design of NoSQL also can have the same 3 phases like relational database that is drawing conceptual data model, transforming into logical data models and building up specific NoSQL databases.

In the phase of conceptual modelling, data quality and standard are ensured from conceptual design phase through the conceptual data model. Second, data management focusing on the Conceptual data model can be consistent and integrated.

While modelling the logical view, key elements are Document, key-value, graph and family elements. Document data model using UML class diagram is shown below.

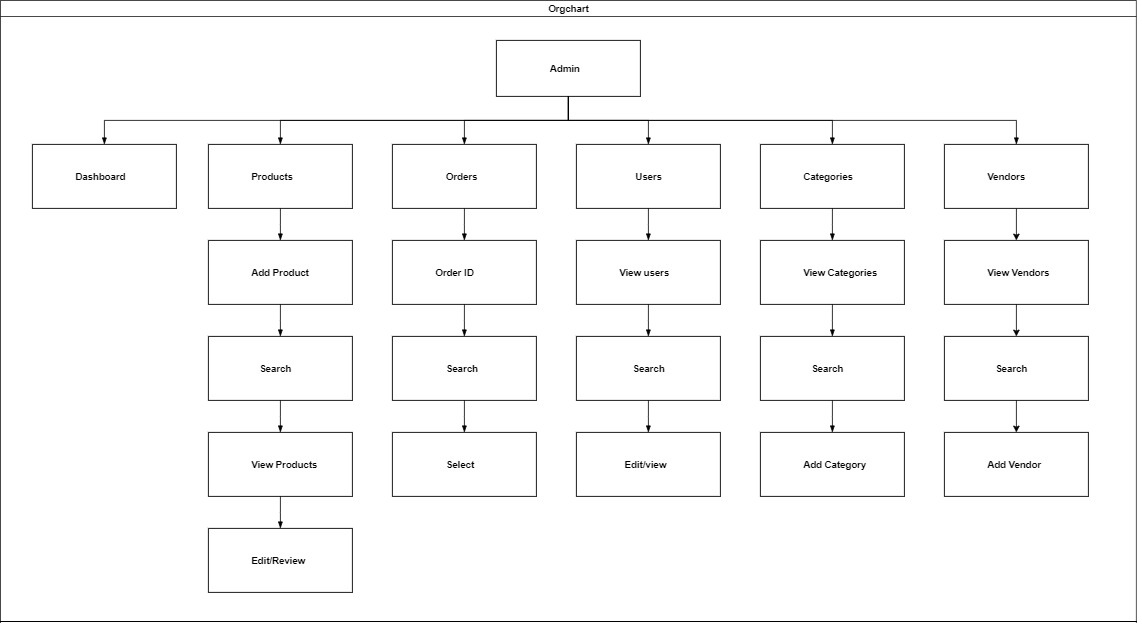


# 

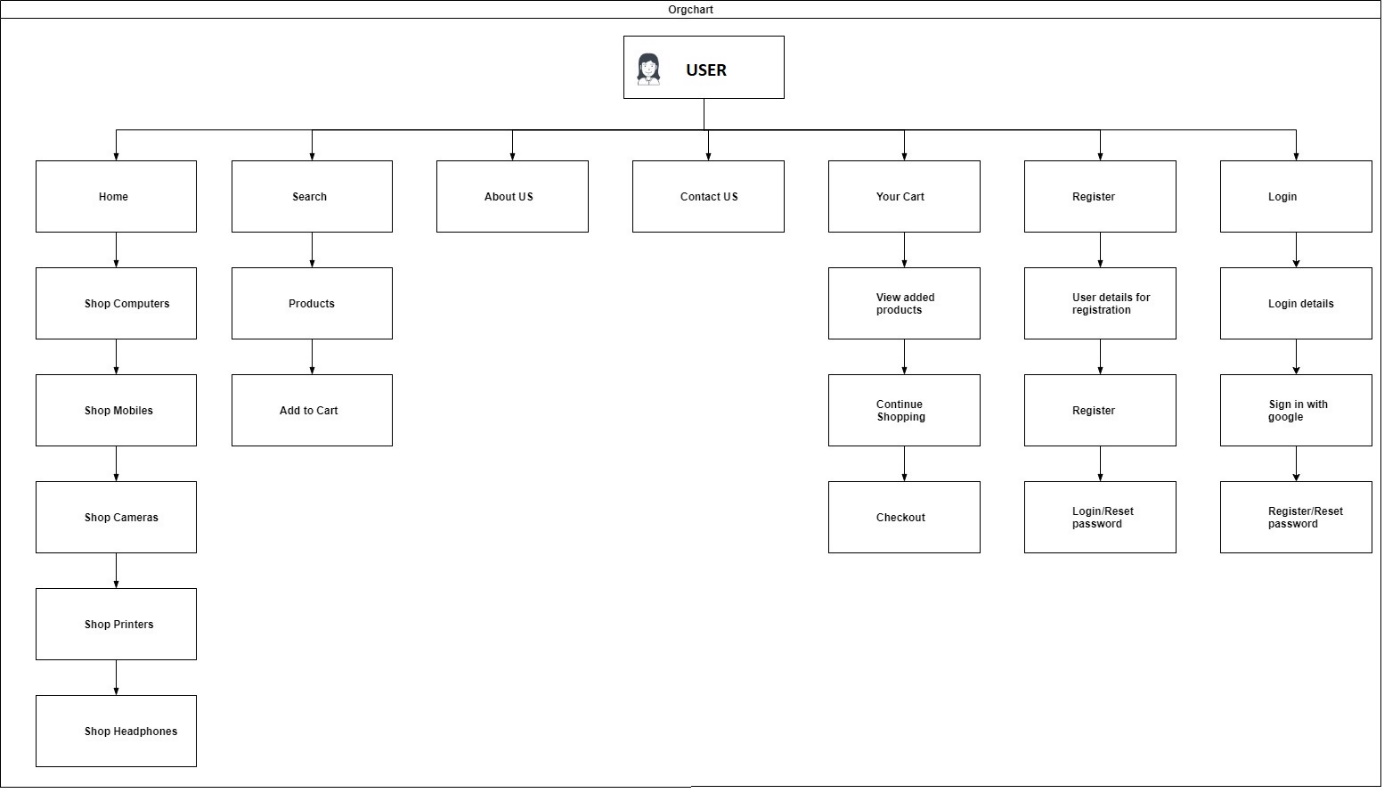
# Solution Architecture

## Information Architecture

Information architecture depicts the blueprint of the system and visually represents the design of the features, infrastructure, and its hierarchy. Further it shows the navigation of the system which categorizes the behavior of the system.



Information Architecture for ADMIN

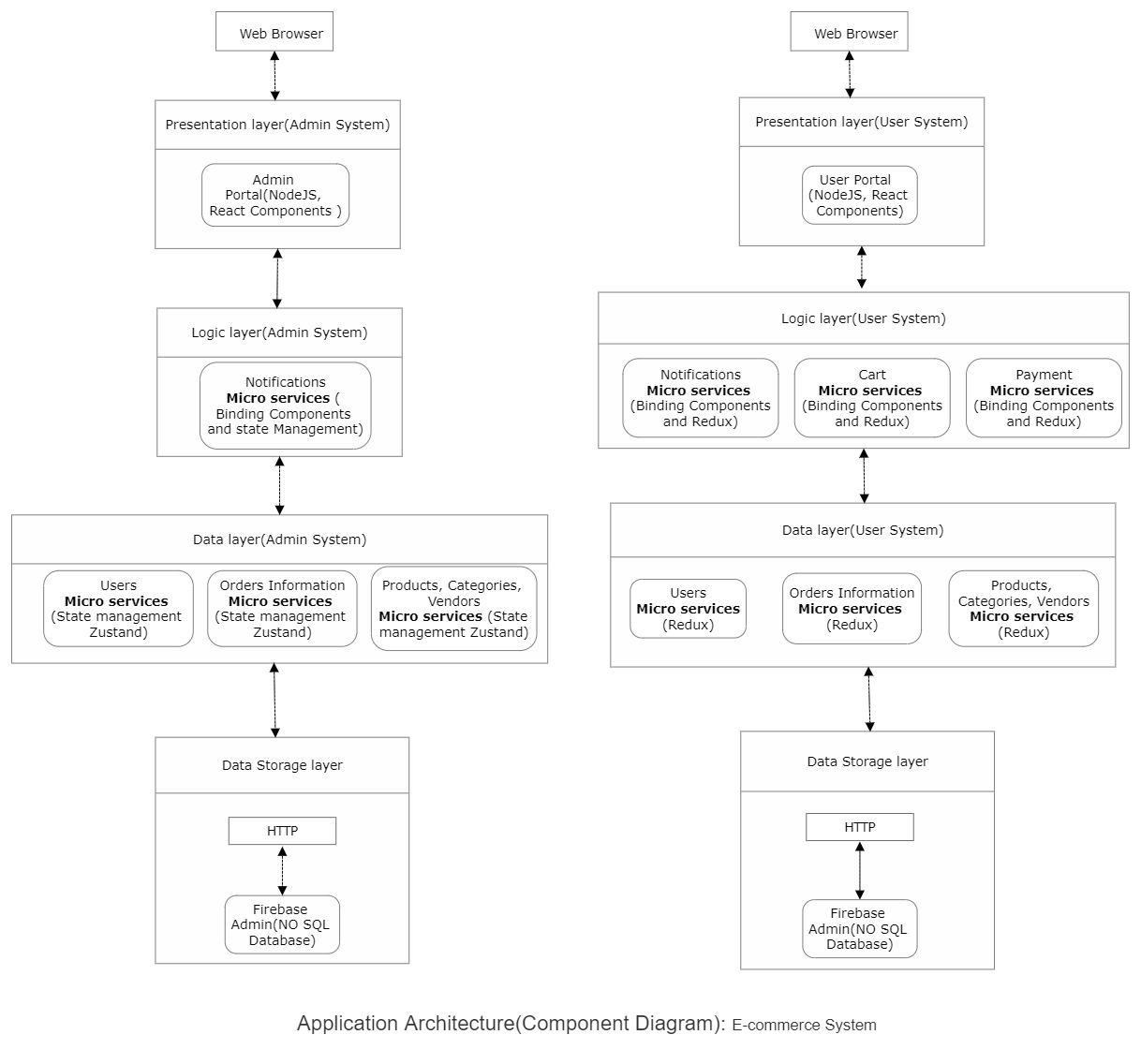
**

Information Architecture for USER

## Application Architecture

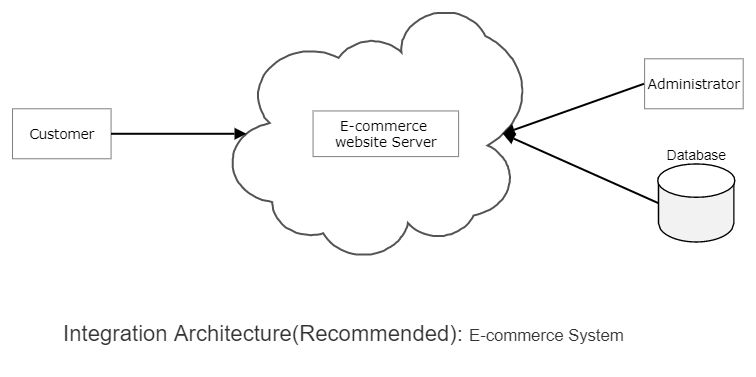
An application architecture shows the technical implementation of the solution by visualizing the integrations and interactions between dependencies such as systems, API, servers, microservices and databases to show the elements and communication flow of the model.

The architecture is divided as Data storage layer using firebase Admin, data service layer for connecting the database with logic layer for state management Zustand or redux, logic layer for binding front end and state management and redux and presentation layer for front end UI features which is implemented using React components.



## Integration Architecture

The integration architecture is concerned to how different architectural blocks are connected to each other. We have already mentioned the detailed connections between each block in application architecture but, as a future recommendation of how the blocks should be integrated, we have the following architecture.



## Infrastructure Architecture

An E-Commerce business does not need a physical building of its own. In fact, a key competitive advantage that ecommerce businesses have over brick-and-mortar stores is not having to invest in physical facilities. Still these businesses do have infrastructure requirements but, they do not include physical things. The infrastructure architecture of an E-Commerce website includes:

* Marketing
* Customer Service
* Information Technology
* Financial Administration
* Human Resource

We also recommend the following criteria for choosing E-Commerce infrastructure.

1. Flexibility: A chosen E-Commerce infrastructure should be flexible and easy to manage for all users. The E-Commerce website should be able to quickly respond to changes and adapt to new technologies and features.
2. Usability: The E-Commerce platform needs to be usable as a single service failure can cause a bad effect on overall efficiency.
3. Scalability: The world of E-Commerce is changing rapidly and new technologies are being introduced every day. Adding new services for customers and making things as smooth as possible for them is the trend that every E-Commerce solution needs to follow.
4. Security: The worldwide web is always under threat of hackers, malwares and security attacks. An E-Commerce platform should be equipped with updated firewalls to make an online shopping experience safe for their customers.

## Security Architecture

Security architecture is out of our scope. However, security is one of the most important aspects of an E-Commerce business. As the online world is rapidly growing and changing, the threats are also growing.

Here are a few recommendations that we propose to benefit for more practical purposes:

* Promote good password hygiene
* Using the https:// protocol
* Choosing a secure E-Commerce platform
* Do not store sensitive user data
* Employ your own website monitor
* Maintain a security focused mindset

# Solution Delivery Considerations

## Development Considerations

For developing Elektronika, we chose REACT. React is a popular tool for building websites and apps. It creates an environment where JavaScript and HTML can be used in the same files. It also renders changing data to the browser very efficiently.

We created reusable components that will always generate the same data in the same way. Users coming to the website would not need to wait for the entire tree to deconstruct and reconstruct when the newest data on the half-pipe is available. That is the quality of using REACT, React components efficiently update to accommodate the changing data.

As the name suggests, react uses a virtual tree method that reacts to changes in input data. When a change occurs, instead of rebuilding the whole Document Object Model (DOM) tree, it decides what changed in its virtual DOM and makes the smallest number of DOM changes necessary, so that the process would be as quick as possible. Almost all current JS frameworks (e.g. Angular, Vue, etc.) are trying to implement similar mechanisms.

Technology is changing rapidly and switching to new technologies in a website involves incrementally transferring the website page by page. This can be a daunting and time taking task.

React makes it very easy to adapt to new technologies. It can easily be used in only some parts of the website, so that we can incrementally refactor our code in React.

Using React components and subcomponents, we can break down our website into the smallest bits possible. It uses the single responsibility principle. Single responsibility principle states that every class in a program should only be responsible for a single part of that program.

For example, in a To-Do list, the hierarchy of React components would include:

* Whole list
* Title
* add a to-do line
* to-do line
  + subtasks within to-dos
* show completed to-dos button

**Reacts components maintain unidirectional data flow:** State flow down the components, and events flow up. When click a button on the child component, it fires an event to the parent component which updates its state, than the UI is re-rendered to reflect state changes for all children affected by the state change. For state management Zustand is used in admin system implementation and Redux is used in the user system implementation.

**Zustand state management:**This framework is fast and scalable state management solution.

**Redux:** This framework helps to behave applications consistently and works with multiple platforms. The successful point is live editing with debugger along with time travelling

**Key features of Redux:**

         Predictable state container

         Testing features of editing while debugging

         Can be reused for other frameworks

**Key features of Zustand:**

         Scalable features

         Hooks the primary state

         Doesn't wrap app into context providers

**Components make React highly reusable:**

React components maintain their property and state. This makes it possible to create highly reusable components that can be dropped into any application.

These features made web components is the future of the web development. React API can be ported over to mobile development and virtual reality apps using React Native Project. React API is the small learning curve compared to other packages of web development.

## Deployment Considerations

Deployment of Elektronika is out of our scope but as data scientists, we would recommend the following as deployment considerations:

Although developers emphasize that it is all about the architecture but most importantly, everything, including deployment, depends on the requirements.

1. Business Case: This is the first stage because all the efforts need to have a business case. If there is no business case, then there is no point to continue.
2. Understanding existing domain: Next step is to break your existing system or systems down to a functional primitive of any architectural components, or data, services, and processes, with the intention being to assemble them as components on the deployment platform.
3. Selecting a deployment platform: Selecting a platform is also an important stage and our recommendation is to select a cloud computing provider.
4. Migrate: This step involves transferring and translating all the architectural assets or files to the new environment, which can be a cloud platform.
5. Deploy the website: Once all the files are transferred to the cloud, it is time to deploy the website and turn it into a production system.
6. Testing: This is the final step before making the website public. In this important stage, we need to test everything based on the requirements provided.

## Data Migration Considerations

The main purpose of Data migration is to enhance performance and competitiveness but an unsuccessful data migration can result in redundancies and ends up creating more problems than it is solving. A strategic data migration plan considers all of these factors:

Data Knowledge: The source data must be thoroughly audited prior to migration or else can lead to unexpected problems.

Clean-up: Any problems with the source data must be addressed as soon as possible. Because of the complexity of the project, additional software tools and third party services may be needed.

Maintaining and protecting: After a certain amount of time, data degrades and becomes unreliable. This necessitates the implementation of controls to ensure data accuracy.

# References

## 1. Scope of the project: <https://keluraevans10.blogspot.com/2019/08/architecture-diagram-for-online-shopping.html>

2. Dependencies for E-Commerce solutions: <https://auth0.com/blog/introduction-to-microservices-part-4-dependencies/>

3. Business features for an E-commerce website: <https://www.searchenginejournal.com/ecommerce-guide/must-have-website-features/#close>

4. What is E-Commerce: <https://searchcio.techtarget.com/definition/e-commerce>

5. Building a website with REACT: <https://www.gatsbyjs.com/blog/2019-01-16-how-to-build-a-website-with-react/>

6. Deployment Consideration for a cloud solution: <https://www.elasticpath.com/blog/depoloying-in-the-cloud>

7. Key features for Ecommerce infrastructure: <https://www.practicalecommerce.com/7-Key-Ecommerce-Infrastructure-Decisions>

8. Ecommerce infrastructure components:

<https://www.purchasecommerce.com/blog/ecommerce-infrastructure-top-influential-growth-components>

9. Security constraints for Ecommerce: <https://uk.pcmag.com/pinnacle-cart/120839/how-to-secure-your-e-commerce-website-6-basic-steps>

10. Disadvantages of Ecommerce solutions: <https://accountlearning.com/limitations-or-disadvantages-of-electronic-commerce/>

11. UML - Behavioral Diagram vs Structural Diagram (visual-paradigm.com)

Shin, K., Hwang, C., Jung, H., & Jung, H. (2017). NoSQL database design using UML conceptual data model based on Peter Chen’s framework. International Journal of Applied Engineering Research, 12(5), 632-636.