

**LA GRANDEE INTERNATIONAL COLLEGE**

**Simalchaur - 8, Pokhara, Nepal**

A Project Report on

“Gadget App”

**Submitted to**

Bachelor of Computer Application (BCA) Program

In partial fulfillment of the requirement for the degree of BCA affiliate to

Pokhara University

**Submitted by:**

|  |  |  |
| --- | --- | --- |
| **Name** | **Program, Semester** | **P.U. Registration Number** |
| Ganesh Tiwari | BCA, 8th Sem | 2021-1-53-0055 |
| Kapil Tiwari | BCA, 8th Sem | 2021-1-53-0059 |
| Prabhat Poudel | BCA, 8th Sem | 2021-1-53-0066 |

Date: 2082/01/18

Acknowledgement

We have presented this report focusing on the topic **“Electromart”**. This report has been prepared for partial fulfilment of the requirement for degree of BCA and to have practical experience.

We are heartily thankful to the faculty of BCA, **LA Grandee International College** and our supervisor **Mr. Sunil Sapkota** for his role to motivate and lead for this report. We obliged towards his constant guidance, supervision and feedbacks which enabled us to prepare a well-executed report.

Further, we express our gratitude to LA Grandee family, classmates, seniors and teachers who have directly and indirectly supported us during our report.

Student's Declarations

**"Electromart"**

We hereby declare that we are the only authors of this work and that no sources other than the mentioned here we have been used in this. We assure you that the work we present here is unique to ourselves and resemblances to another similar project are purely coincidental.

……………………….

**Ganesh Tiwari (2021-1-53-0055)**

……………………….

**Kapil Tiwari (2021-01-53-0059)**

………………………..

**Prabhat Raj Poudel (2021-1-53-0066)**

Program: BCA 8th Semester

Date: 2025-06-27

Supervisor's Declaration

I hereby declare that the project entitled **“Electromart”** has been carried out under my direct supervision by **Ganesh Tiwari, Kapil Tiwari and Prabhat Raj Poudel** during their sixth semester for the partial fulfilment of the requirements for the degree of BCA program under Pokhara University.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Mr. Sunil Sapkota**

(Project Supervisor)

Date: 2025-06-27

****

…………………………..

**Er. Kiran K.C**

Principal

Letter of Approval

We certify that we have examined this report entitled “**Electromart**” and are satisfied with the project defense. It is satisfactory in the scope and qualify as project in partial fulfillment of the requirements for the degree of BCA under Pokhara University.

Date: 2025-06-27

…………………………..

**Mr. Kundan Chaudhary**

Department Coordinator

………………………………….

**b**

External Examiner

…………………………..

**Mr. Sunil Sapkota**

Supervisor

Abstract

Electromart is a dynamic eCommerce application developed to streamline the buying and selling of electronic gadgets. The platform provides an intuitive and responsive user interface, supporting key features such as product categorization, user roles (admin and customer), inventory management, secure payment integration, and order tracking. Electromart offers a wide selection of electronic devices, including smartphones, laptops, tablets, and accessories, aiming to fulfill the diverse needs of modern consumers.

The system enhances the shopping experience through real-time stock updates, advanced product search and filtering, customer reviews, and personalized product recommendations. These features are designed to increase customer satisfaction and operational efficiency for businesses. The development process emphasized creating a user-centric service while also applying full-stack development principles, user interface design strategies, and best practices in eCommerce. This hands-on approach provided valuable experience in building practical, real-world applications.

Beyond its current functionality, Electromart is built with scalability, performance, and security in mind. Its modular codebase allows for future expansion, such as adding promotional tools, loyalty rewards, multi-vendor support, and mobile application integration. Cross-browser compatibility and responsive design were also prioritized to ensure accessibility across all devices. By simulating real-world shopping scenarios, we validated the platform’s ability to handle business-critical operations. Overall, the project strengthened our technical, collaborative, and problem-solving skills, making Electromart a key milestone in our growth as software developers.

Abbreviation

|  |  |
| --- | --- |
| **ACRONYM** | **FULL FORM** |
| API | Application Product Interface |
| APP | Application |
| LGIC | LaGrande International College |
| AUTH | Authentication |
| BAAS | Backend as a Service |
| BCA | Bachelor of Computer Application |
| ITTI | Industrial Technology Transformation Index |
| UI | User Interface |
| IOS | iPhone Operating System |
| DB | Database |
| GIT | Global Information Tracker |
| HTTPS | Hyper Text Transfer Protocol Secure |
| UX | User Experience |
| IT | Information Technology |
| IOS | iPhone Operating System |
| JS | Java Script |
| XSS | Cross-Site Scripting |
| LGIC | LA Grandee International |
| LTD | Limited |
| MS | Microsoft |
| NPM | Node Package Manager |
| OS | Operating System |
| PU | Pokhara University |
| PVT | Private |
| SDK | Software Development Kit |
| SQL | Structured Query Language |
| BaaS | Backend as a Service |
| CSS | Cascading Style Sheets |
| CI | Continuous Integration |
| CD | Continuous Delivery |
| ID | Identification |
| SN | Serial Number |
| CRUD | Create, Read, Update, Delete |
| TS | TypeScript |
| ERD | Entity Relationship Diagram |
| DFD | Data Flow Diagram |
| UML | Unified Modeling language |
| UAT | User Acceptance Testing |

Table of Contents

[List of Tables viii](#_Toc201866091)

[List of Figures ix](#_Toc201866092)

[1. Introduction 1](#_Toc201866093)

[2. Problem Statement 2](#_Toc201866094)

[3. Objective 3](#_Toc201866095)

[4. Background Study 4](#_Toc201866096)

[5. Literature Review 6](#_Toc201866097)

[6. REQUIREMENT DOCUMENT 10](#_Toc201866098)

[7. Methodology 17](#_Toc201866099)

[Fig 5.1: Block Diagram 17](file:///D:\8th%20sem\Project\electromart\documents\final_defence.docx#_Toc201866100)

[Fig 5.2: Iterative model 18](file:///D:\8th%20sem\Project\electromart\documents\final_defence.docx#_Toc201866101)

[8. SYSTEM DESIGN 20](#_Toc201866102)

[9. Gantt Chart 25](#_Toc201866103)

[25](#_Toc201866104)

[Fig 7.1: Gantt Chart 25](file:///D:\8th%20sem\Project\electromart\documents\final_defence.docx#_Toc201866105)

[10. Development Process 27](#_Toc201866106)

[11. Deliverables 28](#_Toc201866107)

[12. Future Enhancements 29](#_Toc201866108)

[13. Conclusion 30](#_Toc201866109)

[14. Reference 31](#_Toc201866110)

List of Tables

[Table 6.1- Requirement Matrix 14](#_Toc201918017)

[Table 6.2 - Work Assigned 16](#_Toc201918018)

List of Figures

[Figure 7.1: Block Diagram 17](#_Toc201918764)

[Figure 7.2 - Iterative Model 18](#_Toc201918765)

[Figure 8.1 - ER Diagram 20](#_Toc201918766)

[Figure 8.2 - DFD Level 0 21](#_Toc201918767)

[Figure 8.3 - DFD Level 1 22](#_Toc201918768)

[Figure 8.4 - DFD Level 2 23](#_Toc201918769)

[Figure 8.5 - UML Diagram 24](#_Toc201918770)

[Figure 8.6 – Use Case Diagram 25](#_Toc201918771)

# Introduction

The Gadget App E-Commerce project is a comprehensive solution for businesses selling electronic gadgets, designed to automate and streamline the online shopping experience. It provides a user-friendly interface with functionalities like managing product inventory, processing orders, handling secure payments, and generating sales and activity reports. The goal is to boost efficiency and profitability for businesses while ensuring a convenient and seamless customer experience. By eliminating manual processes, businesses can focus on growth and enhanced customer service.

The proposed application will offer a wide range of gadgets, including smartphones, tablets, laptops, smartwatches, and accessories, to meet diverse user needs. Advanced features such as real-time stock availability, user-friendly navigation, flexible order and return systems, and secure payment gateways will ensure a smooth shopping experience. This project aims to enhance business efficiency and profitability by automating and streamlining the e-commerce process. With this system in place, businesses can expand their digital sales and serve their customers more effectively.

Moreover, the app is built with scalability and performance in mind, using modern technologies to support high traffic and large product catalogs without compromising speed or reliability. It also integrates analytics tools to help businesses track user behavior, optimize marketing strategies, and make data-driven decisions. With mobile responsiveness and cross-platform compatibility, users can enjoy a consistent shopping experience across all devices. Overall, this project not only simplifies the e-commerce process but also empowers businesses with the tools they need to stay competitive in a rapidly evolving digital marketplace.

# Problem Statement

* Small to mid-sized gadget retailers struggle with managing e-commerce operations due to the absence of a comprehensive and user-friendly mobile application.
* Many rely on manual processes and disconnected tools, causing inefficiencies, errors, and confusion in inventory, order, and customer management—highlighting the need for an efficient and integrated management system.
* There is a growing demand for a customizable solution that can centralize product information and customer data to support smooth and effective operations.
* The absence of a centralized system makes it difficult to store, organize, retrieve, and present data in a way that supports a seamless shopping and administrative experience for both staff and customers**.**

# Objective

* To provide a user-friendly and efficient system that enables sellers to manage gadget products and e-commerce operations effectively and efficiently.
* To collect, store, organize, retrieve, and present product information and customer data for a seamless shopping and management experience.

# Background Study

The Gadget App E-Commerce project is a cross-platform mobile application built using React Native, designed to provide seamless interaction for both buyers and sellers on Android and iOS devices. Its backend is powered by Node.js and Next.js, developed in TypeScript—offering a scalable, high-performance, and maintainable architecture. The system efficiently handles product management, order processing, and customer data storage in real time, enabling users to manage their business operations from anywhere.

To better understand the current landscape of gadget retail operations in our country, we studied existing local platforms such as ITTI Pvt. Ltd. and Nagmani International:

* ITTI Pvt. Ltd. offers a mobile application and web interface that showcases a wide range of electronic gadgets such as laptops, smartphones, and accessories. Their platform provides product listings with technical specifications, promotional offers, and customer support. However, the mobile experience still lacks full seller-side management tools like inventory tracking, real-time updates, and advanced analytics.
* Nagmani International primarily operates through a website, offering renowned brands like ASUS, MSI, and Acer. While the site supports bulk product listings and handles direct customer inquiries, it lacks a dedicated mobile application and offers limited features for real-time inventory and order management, especially for small vendors who want to use the platform for distribution.

In addition to these platforms, we visited a local gadget store to observe how small retailers manage their sales and inventory in real time. The store used a mix of an offline POS system and basic online listings. We noticed several challenges:

* The use of manual and disconnected tools made it difficult to sync inventory and order records efficiently.
* Staff struggled with frequent errors and confusion while tracking stock and customer orders.
* The absence of centralized systems meant that data entry was repetitive and time-consuming.

From these insights, we identified a clear opportunity to develop a **comprehensive and customizable solution** for gadget retailers. Our application is intended to address these gaps by providing:

* A mobile-first system where sellers can easily **add, edit, or remove products**, manage inventory, and process orders.
* A structured database for **real-time storage and retrieval** of customer and product data.
* Advanced features like **search filters by brand, price, or specs**, and **reporting tools** to track sales and performance.
* An intuitive user interface designed to reduce complexity for store staff and improve the shopping experience for customers.

Ultimately, the Gadget App is designed to empower small and mid-sized electronics businesses by improving operational efficiency, reducing human error, and offering a more modern, digital approach to retail management in the growing e-commerce landscape of our country.

# Literature Review

The days of building native applications separately for each mobile platform are fading. React Native has emerged as a powerful cross-platform development framework, allowing developers to build apps for both iOS and Android using a single codebase. This significantly reduces development time and cost while ensuring a consistent user experience. React Native is increasingly being adopted in various domains, including gadget-related mobile applications that offer users information about tech devices, specifications, comparisons, reviews, and shopping options.

In Nepal and other countries, there is a growing trend of mobile apps that provide comprehensive information on electronic gadgets, such as smartphones, laptops, wearables, and accessories. These apps enable users to explore products, compare features, and make informed buying decisions. Many of these apps are developed using React Native because of its ability to deliver smooth UI, native performance, and rapid deployment. These applications also often include notification features, product reviews, and real-time pricing updates, making them valuable for tech-savvy users.

The primary objective of such apps is to bridge the gap between consumers and technology by offering accurate and up-to-date gadget-related content. React Native plays a key role in ensuring that both Android and iOS users receive the same quality experience. Through features like component-based architecture and reusable code, developers can update and maintain apps efficiently.

Research has shown that React Native apps improve development portability, making it easier for teams to build and maintain applications across platforms. For instance, developers can manage gadget-related product databases, display device specifications, and handle user interactions within a unified environment.

One of the main benefits of using React Native in this context is the centralization of gadget data, which allows users to browse products, view specs, and receive news and alerts in one place. This enhances the user experience and encourages engagement. Studies indicate that consumers prefer apps that are responsive, lightweight, and frequently updated—features that React Native supports well. Additionally, frameworks like React Native reduce the learning curve and development effort, making it easier for startups and small teams to launch and scale gadget-focused apps. Dorn and Naz also suggest that many users remain unaware of the latest gadgets due to fragmented information sources. React Native apps that aggregate this information into a user-friendly interface can solve this issue.

Ultimately, the ability to search, compare, and stay informed about gadgets within a single unified mobile platform makes React Native an ideal choice for building gadget comparison and shopping apps. These applications serve not only as informational tools but also play a key role in influencing purchasing decisions.

**Study of Existing Systems**

Nepal’s e-commerce sector has been rapidly growing, particularly in the field of electronics and gadgets. Consumers are increasingly turning to online platforms for convenience, variety, and better pricing. Despite this progress, the sector faces several challenges:

* **Limited Digital Literacy**: A large portion of the population, especially in rural areas, lacks the digital literacy needed to comfortably use e-commerce platforms.
* **Logistics and Delivery Issues**: Inconsistent delivery networks, especially in remote areas, hinder smooth operations and affect customer satisfaction.
* **Trust and Security Concerns**: Many users remain skeptical about online payments, product authenticity, and return policies.
* **Urban-Centric Services**: Most e-commerce services are concentrated in urban areas, leaving rural consumers underserved.
* **Price Sensitivity and Competition**: The market is highly price-sensitive, with intense competition leading to narrow profit margins.

**Analysis of Existing E-commerce Platforms**

Several e-commerce platforms currently operate in Nepal, each with its own strengths and limitations.

**Electromart:**

**Strengths:**

* Specializes in electronic gadgets, offering a niche and focused product range.
* Features an intuitive, user-friendly mobile and web interface.
* Offers competitive pricing and regular discounts on top gadget brands.
* Provides detailed product information, reviews, and comparison tools for informed decision-making.
* Supports various payment methods including digital wallets and cash on delivery.

**Limitations:**

* Hard to reach customers in rural and semi-urban areas because of delivery problems.
* Dependence on third-party delivery partners may impact consistency and reliability.
* Return and refund processes may be slow or cumbersome for some users.
* Limited physical presence or service centers for after-sales support.

# REQUIREMENT DOCUMENT

This requirement documentation outlines the functional, non-functional requirements and the tools and technologies required for the **Electromart**.

**Functional Requirements**

* **User Registration and Authentication**  
  Enable users to sign up and log in securely using email/password or third-party providers (Google, Apple, etc.) using Supabase Auth.
* **User Profiles**  
  Each user (customer/admin) will have a profile to manage personal data, order history, preferences, and saved items.
* **Product Listings and Categories**  
  Admins can add, update, and manage gadgets with images, descriptions, specs, prices, and availability. Products will be categorized (e.g., Mobile, Laptop, Accessories).
* **Search and Filter**  
  Users can search for products by name, category, brand, or price range and apply filters for sorting results (by popularity, price, ratings).
* **Product Details Page**  
  Users can view detailed information, reviews, and related products for each item.
* **Cart Management**  
  Users can add/remove products in the cart, update quantities, and proceed to checkout.
* **Wishlist**  
  Users can add products to a Wishlist for later viewing or purchasing.
* **Order Placement and Management**  
  Users can place orders, choose delivery addresses, and payment methods. Admins can manage orders (status updates, cancellations).
* **Ratings and Reviews**  
  Users can rate products and leave reviews; admins can moderate the reviews.
* **Dashboard (Admin Panel)**  
  Admin interface for managing products, orders, user accounts, and analytics.

**Non-Functional Requirements**

* **Cross-Platform Compatibility**  
  The application must work seamlessly on iOS, Android (mobile), and desktop (web via Next.js).
* **Scalability & Performance**  
  Ensure high performance and scalability using serverless architecture with Supabase and optimized rendering with Next.js.
* **Clean Code Architecture**  
  Use modular, reusable TypeScript components with proper folder structures and state management (e.g., Zustand, Redux, or Context API).
* **Responsive and Intuitive UI/UX**  
  Provide a user-friendly experience using responsive designs, accessible components, and consistent branding.
* **Security**  
  Implement secure authentication, protect customer data, and prevent common threats (XSS, CSRF).
* **Future Growth**  
  Architecture should support adding new features like subscription plans, promotions, or affiliate systems without major restructuring.

**Tools and Technologies**

**Development Tools**

* **IDE:** Visual Studio Code
* **Version Control:** Git
* **Repository Hosting:** GitHub
* **Database & Backend as a Service (BaaS):** Supabase (PostgreSQL, Auth, Realtime)

**Frontend**

* **Mobile App:** React Native (with Expo)
* **Web App:** Next.js with TypeScript
* **UI Libraries:** Tailwind CSS, Native Base, for fast and consistent UI components

**Package Managers**

* **npm:** For dependency and package management

**Other Tools**

* **CI/CD:** GitHub Actions (for automated testing and deployments)
* **Analytics:** PostHog, Mixpanel, or Firebase Analytics
* **Payment Integration:** E-Sewa, Bank deposit and cash on delivery (for handling online payments)

**Requirement Matrix**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirement  ID | Requirement Description | Type | Priority | Status |
| FR-01 | Users can securely register and log in using email/password or OAuth (Supabase) | Functional | High | Completed |
| FR-02 | Users can browse products by category and search/filter items | Functional | High | Completed |
| FR-03 | Users can view product details, specs, images, and reviews | Functional | High | Completed |
| FR-04 | Users can add/remove items in the cart and update quantity | Functional | |  | | --- | | High | | |  | | --- | | Completed | |
| |  | | --- | | FR-05 | | Users can add gadgets to their Wishlist for later viewing | |  | | --- | | Functional | | Medium | |  | | --- | | Completed | |
| FR-06 | Users can place orders with delivery and payment information | Functional | High | Completed |
| FR-07 | Users can write and view product reviews | Functional | Medium | Completed |
| FR-08 | Admins can manage product listings, inventory, and orders | Functional | High | Completed |
| NFR-01 | The application must be responsive on both mobile and web | Non-Functional | High | Completed |
| NFR-02 | UI should be intuitive and user-friendly using reusable TypeScript components | Non-Functional | High | Completed |
| NFR-03 | The app architecture must support scalability and easy feature additions | Non-Functional | High | Completed |
| NFR-04 | User data and payment information must be securely stored (Supabase + HTTPS) | Non-Functional | High | Completed |
| NFR-05 | CI/CD pipeline setup using GitHub Actions for automatic deployment | Non-Functional | Medium | Completed |

Table .- Requirement Matrix

**How we used Iterative Methodology in our project (Electromart)?**

**Sprint Planning:** At the beginning of each week, we conducted a sprint planning meeting to define the tasks and goals for the upcoming sprint based on project priorities and assigned modules like frontend, backend, database, and documentation.

**Daily Stand-ups:** Every day, we held short stand-up meetings to discuss our progress, align tasks, address blockers, and adjust workload if required. These stand-ups ensured all three of us stayed in sync across development stages.

**Sprint Review:** At the end of each week, we organized a sprint review meeting where we demonstrated the completed work, received feedback from peers and mentors, and ensured the sprint goals were met effectively.

**Sprint Retrospective:** After the sprint review, we reflected on what went well, what challenges we faced, and what could be improved in the upcoming sprints. This allowed us to iteratively optimize our development process. We are a group of 3 students were confined to the development of the final year project and here is the work division chart among us:

|  |  |  |  |
| --- | --- | --- | --- |
| S.N | Name | Work Assigned | Remarks |
| 1. | Ganesh Tiwari | Week 1: Project Setup, Home Screen UI, Login/Signup Screen Week 2: UI for Authentication Week 3: Product Listing Page, Filter UI Week 4: Cart & Wishlist UI Week 5: Checkout Page UI Week 6: Admin Dashboard UI & UI Enhancements Week 7: UI Bug Fixes & Responsiveness Testing | Week 1: Satisfied  Week 2: Satisfied  Week 3: Refinement  Week 4: Satisfied  Week 5: Refinement  Week 6: Satisfied  Week 7: Satisfied |
| 2. | Kapil Tiwari | Week 1: Database Schema (Users, Products, Orders) Week 2: Supabase Auth Setup, Role Config Week 3: Product CRUD APIs (Create, Read, Update, Delete) Week 4: Cart & Wishlist API (Relations & Joins) Week 5: Order API & Mock Payment Flow Week 6: Admin Product Management APIs Week 7: Error Handling & Backend Testing | Week 1: Satisfied  Week 2: Satisfied  Week 3: Refinement  Week 4: Refinement  Week 5: Refinement  Week 6: Satisfied  Week 7: Satisfied |
| 3. | Prabhat Raj Poudel | Week 1: Requirement Specification, Tech Stack Documentation Week 2: Auth Flow Diagram & API Contracts Week 3: UI/User Flow Documentation Week 4: Sequence Diagrams, Test Case Drafts Week 5: Deployment Plan, Flow Charts Week 6: Admin Guide, Bug Reporting Week 7: Final Report, README, Presentation Slides | Week 1: Satisfied  Week 2: Satisfied  Week 3: Refinement  Week 4: Satisfied  Week 5: Refinement  Week 6: Satisfied  Week 7: Satisfied |

Table . - Work Assigned

# Methodology

The Iterative Model is a software development approach that involves repeating a sequence of steps—such as planning, designing, implementing, and testing—until the desired level of quality is achieved. This model is used for the development of this Android-based e-commerce application, as it ensures that each iteration is thoroughly tested, reviewed, and improved. By incorporating feedback in each cycle, the development team can continuously enhance the app’s performance, usability, and functionality, leading to a more reliable and user-friendly final product.

**Block Diagram:**



Figure .: Block Diagram

**Iterative Model**

Iteration 1

Requirement

Iteration 3

Iteration 2

Design

Deployment

Maintenance

Design

Coding

Testing

Implementation

Review

Coding

Testing

Implementation

Review

Review

Analysis

Design

Coding

Testing

Implementation

Figure . - Iterative Model

The Iterative Model for the development of the Android-based Gadget E-Commerce application typically consists of the following phases:

1. **Requirement Gathering:** In this phase, we gather the project objectives, scope, and requirements. We also define the features for the mobile application, such as product catalog management, order processing, and secure payment systems, and identify the resources required for the project.
2. **Analysis:** During this phase, we analyze the gathered requirements in detail. Any potential issues or gaps are identified, and we develop diagrams (such as flowcharts or wireframes) to help clarify the system's functionality and user interactions.
3. **Design:** In this phase, we create detailed designs for the system. This includes designing the user interface (UI) for the Android app, database schema using MongoDB, and defining the software architecture for the backend using Node.js/Next.js and TypeScript.
4. **Coding:** This phase involves writing the actual code for the mobile application and backend system. The Android app is developed using Java/Kotlin, while the backend is implemented using Node.js and Next.js, with MongoDB for data storage.
5. **Testing:** In this phase, we rigorously test the Android app and backend to ensure they meet the defined requirements. We perform functional testing, performance testing, and security testing to verify the system works as expected.
6. **Implementation:** Once the system passes the testing phase, it is deployed to production. The Android app is made available on the Google Play Store, and the backend is deployed to a cloud service, making the app ready for use by customers and sellers.
7. **Review:** After deployment, we continue to maintain and support the system. This includes fixing bugs, enhancing features based on user feedback, and ensuring the app stays updated with new technology and security patches.

# SYSTEM DESIGN

System design is the process of planning, structuring, and organizing the components of an application to meet specific business or user needs. It involves outlining how different modules of the system interact, the flow of information, and how users engage with the application. In **Electromart**, an e-commerce app focused on gadgets, commonly used tools in system design include **Data Flow Diagrams (DFDs)**, **Flowcharts**, and **ER Diagrams**.

**ER Diagram:**

An ER diagram shows the relationships between various entities in a database. Each entity represents a real-world object like a user, product, or order, and contains attributes. The ER diagram helps in designing the database structure efficiently and maintaining data consistency and integrity throughout the app.



Figure . - ER Diagram

**Admin Adds Product**

Admin users can upload gadgets with details like Product Name, Description, Price, Quantity, Brand, Specifications, and Images.

**User Browses & Orders**

Users can browse the product catalog, view gadget details, and add items to their cart. Once confirmed, they place orders by providing delivery information and making payments.

**User Wishlist**

Users can add their favorite gadgets to the Wishlist for future consideration.

**Dataflow Diagram:**

A **Data Flow Diagram (DFD)** visually represents how data moves through the Electromart system. It outlines user inputs (such as placing orders or signing up), processes (like order processing and payment verification), data stores (like user info or product lists), and external systems (like payment gateways or shipping APIs). This helps understand the overall functionality and interaction between components.

**DFD Level 0**

This Level 0 DFD shows a high-level overview of the **Electromart** system, capturing interactions between the **Customer**, **Admin**, and the **System**. The central process involves browsing products, placing orders, handling wishlists, managing payments, and maintaining product listings. It provides a simplified snapshot of how the user and admin interact with the e-commerce platform.



Figure . - DFD Level 0

**DFD Level 1**

Level 1 DFD expands on the core system to reveal more detail on sub-processes. It breaks the Electromart system into the following sub-processes:

* **Browse & Search Products:** Customers explore available gadgets.
* **Add to Cart / Wishlist:** Customers store items temporarily or for later.
* **Place Order:** Finalizing order with delivery info and payment.
* **Manage Products:** Admin adds/updates/deletes products.
* **Manage Orders:** Admin tracks and updates order status.
* **Process Payment:** Payments are processed and verified.

This level shows how data flows between customers, system processes, and data stores like the product catalog, order history, and user profiles.



Figure . - DFD Level 1

**DFD Level 2 (From the perspective of Customer)**

Level 2 DFD explores deeper into how the **Customer** interacts with the system. It focuses on the user’s flow from browsing to purchase:

* **Search Gadget → View Gadget Details → Add to Cart / Wishlist**
* **Proceed to Checkout → Enter Address → Make Payment**
* **Track Orders → View Order History**

Each process connects with related data stores like the product list, Wishlist data, and payment details. This level of detail is helpful in understanding the customer experience journey in the Electromart app.



Figure . - DFD Level 2

**UML Diagram**

Figure . - UML Diagram

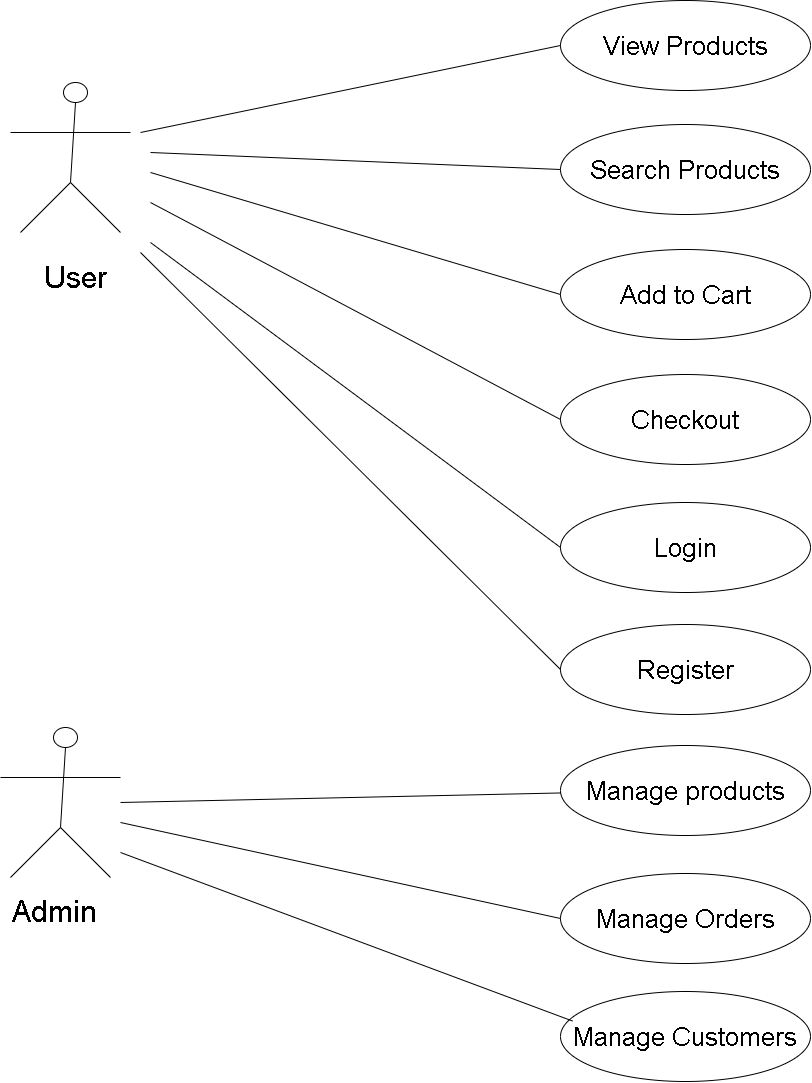
**Use Case Diagram**

Figure . – Use Case Diagram

# Development Process

The development phase of the Electromart mobile application involved implementing frontend and backend components, integrating essential APIs, and conducting thorough testing to ensure the application met all specified requirements. Using the Iterative model, we engaged in an iterative process that allowed us to adapt to evolving user needs and prioritize features based on feedback and market demands. This phase was crucial in transforming the initial concept into a functional, user-friendly app offering a seamless online shopping experience for gadgets.

The development process was organized into two-week sprints, each focused on achieving specific milestones and implementing key features. Regular sprint planning and retrospective meetings helped maintain consistent progress. Daily stand-up meetings promoted effective team communication and collaboration, enabling us to address technical challenges early in the cycle. Notable features developed included product listing and filtering, a secure checkout system, user account management, and an admin dashboard for inventory and order tracking.

To ensure code quality and application stability, we adopted continuous integration practices using tools like GitHub. This streamlined the code review process and ensured that the application remained in a deployable state throughout the development lifecycle. Additionally, user acceptance testing (UAT) offered crucial feedback that guided our iterative improvements and refinements.

In conclusion, the development phase of the Gadget App using the Agile methodology was a collaborative and adaptive process. Through effective sprint planning, user-focused development, continuous integration, and thorough testing, we delivered a feature-rich and high-performance mobile application. This approach not only helped us meet project timelines but also ensured the delivery of a robust and engaging product that meets the expectations of both gadget enthusiasts and online shoppers.

# Deliverables

The deliverables of this project are as follows:

* **Database Management System** to store information about gadgets, orders, customers, and transaction history.
* **Customizable Order and Payment Processes** for customers to place orders, track deliveries, and make secure payments.
* **Real-time Inventory Management** to track stock availability, update product listings, and manage order fulfillment.
* **Security Features** to protect sensitive customer data and transactions, ensuring secure logins, payments, and data storage.
* **User-Friendly Mobile Interface** designed for easy navigation, allowing customers to browse products, place orders, and manage their accounts seamlessly.
* **Product Management** system for sellers to add, update, or remove gadgets from the catalog, track sales, and manage product listings.

# Future Enhancements

# Conclusion

The Gadget E-Commerce Application project is an efficient and scalable solution for businesses looking to automate and streamline their online sales and inventory management. With its wide range of customizable features and functionalities, this project can be tailored to meet the unique needs of different e-commerce retailers. The user-friendly mobile interface, along with real-time inventory management and secure payment processing, ensures a seamless shopping experience for both customers and sellers. Overall, it is a valuable tool for businesses aiming to enhance their operations and drive growth in the competitive e-commerce market.

# Reference

Rouse, M. (2017). *Mobile application development (mobile app development)*. TechTarget. <https://www.techtarget.com/searchsoftwarequality/definition/mobile-application-development>

ITTI Pvt. Ltd. (n.d.). *ITTI Computers - Your Trusted IT Training Institute in Nepal*. Retrieved May 7, 2025, from <https://www.itti.com.np>

Nagmani International. (n.d.). *Official distributor of ASUS and other tech brands in Nepal*. Retrieved May 7, 2025, from <https://www.nagmani.com.np>