**HANOI UNIVERSITY OF INDUSTRY**

**FACULTY OF INFORMATION TECHNOLOGY**

****

**ASSIGNMENT REPORT**

**Subject: Introduction to Software Engineering**

**TOPIC: APPLYING THE PROTOTYPING PROCESS MODEL TO SPECIFY REQUIREMENTS FOR THE ADMIT AUCTION WEBSITE**

|  |  |
| --- | --- |
| **Instructor:** | **Ms. Nguyen Thi My Binh** |
| **Group of students performed:** |  |
| **1. Nguyễn Tuấn Anh** | **Student Code: 2022602757** |
| 1. **Trần Văn Bằng** | **Student Code: 2022601111** |
| 1. **Nguyễn Quốc Phú** | **Student Code: 2022603590** |
| 1. **Lầu Minh Tuấn** | **Student Code: 2022602833** |
| 1. **Phạm Văn Tùng** | **Student Code: 2022600857** |
| **Class Code:** | **2022DHCNTTTA01** |
| **Group:** | **2** |

**Hanoi - 2024**

# TABLE OF CONTENTS

[TABLE OF CONTENTS 1](#_Toc16228)

[LIST OF FIGURES 2](#_Toc13916)

[PREAMBLE 3](#_Toc9194)

[Chapter 1: Theoretical Basis Of The Prototyping Model 5](#_Toc30674)

[1.1. Basic concepts of software engineering 5](#_Toc10897)

[1.1.1. Software 5](#_Toc14945)

[1.1.2. Software engineering 6](#_Toc14457)

[1.2. The prototyping model 7](#_Toc1162)

[1.2.1. General introduction of process model 7](#_Toc8312)

[1.2.2. Definition of the Prototyping Model 8](#_Toc4802)

[1.2.3. Phases of the Prototyping Model 8](#_Toc22105)

[1.2.4. Advantages and Disadvantages 9](#_Toc13859)

[Chapter 2: Specification Requirements For Building Admit Auction Website 12](#_Toc26952)

[2.1. Introduction 12](#_Toc8638)

[2.1.1. Purpose 12](#_Toc26204)

[2.1.2. Scope 12](#_Toc25457)

[2.1.3. Definitions, Acronyms and Abbreviation 14](#_Toc31872)

[2.1.4. Overview 15](#_Toc32605)

[2.2. Overall Description 16](#_Toc20827)

[2.2.1. Product perspective 16](#_Toc31183)

[2.2.2. Product functions 17](#_Toc23191)

[2.2.3. User characteristics 18](#_Toc18818)

[2.2.4. General constraints 19](#_Toc10727)

[2.2.5. Assumptions and dependencies 20](#_Toc7801)

2.3. Specific Requirements 22

[2.3.1. Use case diagram 22](#_Toc26612)

[2.3.2. Functional requirements 23](#_Toc7326)

[2.3.3. Nonfunctional requirements 49](#_Toc30529)

[Chapter 3: Conclusion 51](#_Toc14757)

[3.1. Achievements 51](#_Toc1795)

[3.2. Knowledge Gained 51](#_Toc32502)

[3.3. Skills Acquired 52](#_Toc24040)

[REFERENCES 53](#_Toc30693)

# LIST OF FIGURES

|  |  |
| --- | --- |
| Figure 1.1 Software Engineering | 6 |
| Figure 1.2 Software Quality | 7 |
| Figure 1.3 Prototyping Model | 8 |
| Figure 2.1 System Perspective Diagram | 17 |
| Figure 2.2 The General Use Case Diagram | 22 |

# PREAMBLE

**Reasons for choosing the topic:**

Information technology (IT) and software have become pivotal elements in enhancing and boosting the efficiency of business operations in the modern world. Amid increasingly fierce competition, businesses not only need to maintain traditional methods but also continuously innovate and adopt advanced technological solutions to meet market and customer demands. The application of IT in business activities brings numerous outstanding benefits, such as increased work efficiency, reduced operational costs, improved service and product quality, and new opportunities for market expansion and business development. With the continuous advancement of technology, management software, data analysis tools, and automation applications have been transforming the way businesses operate, creating smarter and more flexible business environments. Therefore, the study and application of IT and software in business activities are not just trends but essential factors determining the success and sustainability of businesses in the digital age.

The Software Requirements Specification (SRS) is one of the most critical documents in software development. It details the functional and non-functional requirements that the software system must meet, acting as a bridge between customers, system analysts, developers, and other stakeholders. The importance of SRS in the development process cannot be overstated. Firstly, it clearly and specifically defines project requirements, reducing the risk of errors or misunderstandings during design and development. Clear requirements allow developers to focus on necessary features, avoiding wasted resources on unnecessary functions. Additionally, the SRS serves as a foundation for testing and quality assurance, ensuring the final product meets customer and user expectations. Furthermore, the SRS is crucial for project management and progress tracking, providing a standard reference framework to monitor progress and assess the completion of development stages. If requirements change, the SRS helps evaluate the impact on the overall system and development plan.

By applying information technology, building software to support business activities will not only help AdmitAuction increase revenue and reach more customers, providing better customer experiences, but also enhance the company's competitiveness against other rivals, especially in an era where many companies are accelerating their adoption of IT. This technological advancement allows AdmitAuction to streamline operations, reduce costs, and improve accuracy in transactions. Moreover, it positions the company to swiftly adapt to market changes and customer demands, ensuring long-term sustainability and growth in a rapidly evolving digital landscape.

Based on the above requirements, our team has chosen the topic for our major project: "Applying the Prototyping Process Model to Specify Requirements for the Admit Auction Website." This project focuses on understanding the prototyping process model and specifying the requirements for the company's website.

**Acknowledgments:**

For a student of the Faculty of Information Technology, knowledge of the course Introduction to Software Engineering is indispensable, because the course will provide basic knowledge about the software development process and process models. software development, requirements specification analysis and software design, etc. From there, after completing the module, students can understand what software technology is and the practical applications in the process. create a software product.

During the process of implementing the project, we received a lot of help from Ms. Nguyen Thi My Binh - lecturer of Introduction to Software Technology. We would like to sincerely thank her for equipping us with valuable knowledge during our study at Hanoi University of Industry. She dedicatedly taught, advised, and equipped us with the most necessary knowledge throughout the study and research process, and created favorable conditions to help us carry out this topic.

With limited experience and limited knowledge, we have tried our best but our topic still cannot avoid shortcomings. We hope to receive comments and advice about shortcomings in our topic from you so that we can learn from experience for the next big assignments.

We sincerely thank you!

# Chapter 1: Theoretical Basis Of The Prototyping Model

## 1.1. Basic concepts of software engineering

### 1.1.1. Software

**Software is:** *(1) instructions (computer programs) that when executed provide desired features, function, and performance; (2) data structures that enable the programs to adequately manipulate information, (3) descriptive information in both hard copy and virtual forms that describes the operation and use of the programs.*

Software is the key element in the evolution of computer-based systems and products and one of the most important technologies on the world stage. Over the past 50 years, software has evolved from a specialized problem solving and information analysis tool to an industry in itself. Yet we still have trouble developing high quality software on time and within budget.

In order to build software that is ready to meet the challenges of the twenty-first century, you must recognize a few simple realities:

- Software has become deeply embedded in virtually every aspect of our lives, and as a consequence, the number of people who have an interest in the features and functions provided by a specific application has grown dramatically. When a new application or embedded system is to be built, many voices must be heard. And it sometimes seems that each of them has a slightly different idea of what software features and functions should be delivered. It follows that a concerted effort should be made to understand the problem before a software solution is developed.

- The information technology requirements demanded by individuals, businesses, and governments grow increasingly complex with each passing year. Large teams of people now create computer programs that were once built by a single individual. Sophisticated software that was once implemented in a predictable, self-contained, computing environment is now embedded inside everything from consumer electronics to medical devices to weapons systems. The complexity of these new computer-based systems and products demands careful attention to the interactions of all system elements. It follows that design becomes a pivotal activity.

- Individuals, businesses, and governments increasingly rely on software for strategic and tactical decision making as well as day-to-day operations and control. If the software fails, people and major enterprises can experience anything from minor inconvenience to catastrophic failures. It follows that software should exhibit high quality.

- As the perceived value of a specific application grows, the likelihood is that its user base and longevity will also grow. As its user base and time-in-use increase, demands for adaptation and enhancement will also grow. It follows that software should be maintainable.

These simple realities lead to one conclusion: software in all of its forms and across all of its application domains should be engineered. And that leads us to the topic of this part - **software engineering.**

### 1.1.2. Software engineering

Software engineering encompasses a process, a collection of methods (practice) and an array of tools that allow professionals to build high quality computer software.

A diagram of a quality focus

Description automatically generated

###### Figure 1.1 Software engineering

Software is important because it affects nearly every aspect of our lives and has become pervasive in our commerce, our culture, and our everyday activities. Software engineering is important because it enables us to build complex systems in a timely manner and with high quality.

The international standard on software product quality suggests that software quality comprises six main attributes, show in figure below:

A diagram of software quality

Description automatically generated

###### Figure 1.2 Software quality

- Functionality: The capability to provide functions which meet stated and implied needs when the software is used.

- Reliability: The capability to provide failure-free service

- Usability: The capability to be understood, learned, and used.

- Efficiency: The capability to provide appropriate performance relative to the amount of resources used.

- Maintainability: The capability to be modified for purposes of making corrections, improvements, or adaptation.

- Portability: The capability to be adapted for different specific environments without applying actions or means other than those provided for this purpose in the product.

## 1.2. The prototyping model

### 1.2.1. General introduction of process model

A process is a collection of activities, actions, and tasks that are performed when some work product is to be created. An activity strives to achieve a broad objective and is applied regardless of the application domain, size of the project, complexity of the effort, or degree of rigor with which software engineering is to be applied. An action encompasses a set of tasks that produce a major work product. A task focuses on a small, but well-defined objective that produces a tangible outcome.

In the context of software engineering, a process is not a rigid prescription for how to build computer software. Rather, it is an adaptable approach that enables the people doing the work (the software team) to pick and choose the appropriate set of work actions and tasks. The intent is always to deliver software in a timely manner and with sufficient quality to satisfy those who have sponsored its creation and those who will use it.

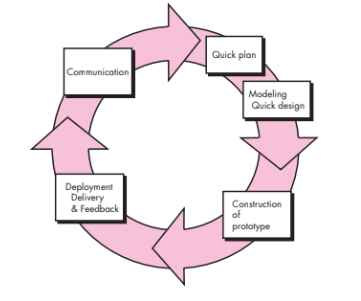
### 1.2.2. Definition of the Prototyping Model

The prototyping model in software development is an iterative approach that prioritizes early user validation and feedback. It involves creating a limited-functionality version of the software, known as a prototype, early in the development cycle. This prototype focuses on core features and functionalities, aiming to provide a concrete representation of the intended user experience and core functionalities.

The key aspect of the prototyping model lies in its iterative nature. The initial prototype serves as a starting point for gathering feedback from stakeholders, including users and clients. This feedback is then used to refine and evolve the prototype in subsequent iterations. This cycle of building, evaluating, and refining continues until a satisfactory version is achieved. This final, refined prototype then becomes the blueprint for developing the full-fledged software application.

The prototyping model is particularly valuable for projects where requirements are unclear, user needs are not fully understood, or innovative features are being explored.

### 1.2.3. Phases of the Prototyping Model



**Figure 1.3 Prototyping Model**

**Step 1: Requirement Gathering and Analysis(Communication):**This is the initial step in designing a prototype model. In this phase, users are asked about what they expect or what they want from the system.

**Step 2: Quick plan**: The quick plan for prototyping involves outlining core functionalities, user testing methods, and an iteration strategy. This ensures focused development and valuable user feedback to refine your software concept.

**Step 3: Quick Design:**This is the third step in the Prototyping Model. This model covers the basic design of the requirement through which a quick overview can be easily described.

**Step 4: Build a Prototype:**This step helps in building an actual prototype from the knowledge gained from prototype design.

**Step 5: Beployment, Delivery and feedback:**This step describes the preliminary testing where the investigation of the performance model occurs, as the customer will tell the strengths and weaknesses of the design, which was sent to the developer.

**Step 6: Refining Prototype:**If any feedback is given by the user, then improving the client’s response to feedback and suggestions, the final system is approved.

**Step 7: Implement Product and Maintain:**This is the final step in the phase of the Prototyping Model where the final system is tested and distributed to production, here the program is run regularly to prevent failures.

### 1.2.4. Advantages and Disadvantages

**Advantages of Prototyping Model:**

- The customers get to see the partial product early in the life cycle. This ensures a greater level of customer satisfaction and comfort.

- New requirements can be easily accommodated as there is scope for refinement.

- Missing functionalities can be easily figured out.

- Errors can be detected much earlier thereby saving a lot of effort and cost, besides enhancing the quality of the software.

- The developed prototype can be reused by the developer for more complicated projects in the future.

- Flexibility in design.

- Early feedback from customers and stakeholders can help guide the development process and ensure that the final product meets their needs and expectations.

- Prototyping can be used to test and validate design decisions, allowing for adjustments to be made before significant resources are invested in development.

- Prototyping can help reduce the risk of project failure by identifying potential issues and addressing them early in the process.

- Prototyping can facilitate communication and collaboration among team members and stakeholders, improving overall project efficiency and effectiveness.

-Prototyping can help bridge the gap between technical and non-technical stakeholders by providing a tangible representation of the product.

**Disadvantages of the Prototyping Model:**

- Costly concerning time as well as money.

- There may be too much variation in requirements each time the prototype is evaluated by the customer.

- Poor Documentation due to continuously changing customer requirements.

- It is very difficult for developers to accommodate all the changes demanded by the customer.

- There is uncertainty in determining the number of iterations that would be required before the prototype is finally accepted by the customer.

- After seeing an early prototype, the customers sometimes demand the actual product to be delivered soon.

- Developers in a hurry to build prototypes may end up with sub-optimal solutions.

- The customer might lose interest in the product if he/she is not satisfied with the initial prototype.

- The prototype may not be scalable to meet the future needs of the customer.

- The prototype may not accurately represent the final product due to limited functionality or incomplete features.

- The focus on prototype development may shift away from the final product, leading to delays in the development process.

- The prototype may give a false sense of completion, leading to the premature release of the product.

- The prototype may not consider technical feasibility and scalability issues that can arise during the final product development.

- The prototype may be developed using different tools and technologies, leading to additional training and maintenance costs.

- The prototype may not reflect the actual business requirements of the customer, leading to dissatisfaction with the final product.

# Chapter 2: Specification Requirements For Building Admit Auction Website

## 2.1. Introduction

### 2.1.1. Purpose

The purpose of this Software Requirements Specification (SRS) document is to define the requirements for the development of the SuperEBid system. This system is intended to enable AdmitAuction to transition from traditional auction methods to a web-based auction platform. The SRS will outline the necessary features and functionalities required to meet the needs of corporate sellers, individual buyers, and other stakeholders.

### 2.1.2. Scope

The scope of the Admit Auction website project encompasses several key areas of development and functionality, including but not limited to:

1. User Registration and Authentication:

- Implementation of a secure user registration system allowing users to create accounts using email addresses, social media logins, or other authentication methods. Integration of authentication protocols to ensure secure access and protection of user data.

1. Item Listing and Management:

- Development of a user-friendly interface for sellers to list items for auction, including the ability to upload images, provide detailed descriptions, and set starting bid prices. Tools for sellers to manage their listed items, including options to edit or remove listings before the auction begins.

1. Bidding Process:

- Real-time auction capabilities allowing users to place bids on items with immediate updates on the current highest bid. Implementation of automated bidding options, such as proxy bidding, where users can set maximum bid amounts.

1. Payment Processing:

- Integration with secure payment gateways to facilitate transactions between buyers and sellers once an auction concludes. Support for various payment methods, including credit/debit cards, PayPal, and other popular online payment systems.

1. User Account Management:

- Comprehensive account management features for users to view their bidding history, manage their active auctions, and update personal information. Notification system to alert users about auction status changes, outbid notifications, and auction outcomes.

1. Security Measures:

- Implementation of robust security protocols to protect user data and prevent unauthorized access. Regular security audits and updates to ensure ongoing protection against emerging threats.

1. Admin Panel:

- Development of an administrative interface for site administrators to monitor auctions, manage user accounts, and handle disputes or reported issues.

1. Reporting and Analytics:

- Tools for generating reports and analytics to provide insights into auction performance, user activity, and site traffic.

1. Third-Party Integrations:

- Integration with third-party services for enhanced functionalities such as payment processing, shipping calculators, and identity verification.

By achieving these objectives, the Admit Auction Website aims to become a leading online auction platform that attracts a diverse user base, fosters a thriving marketplace, and delivers a high level of satisfaction to both buyers and sellers. The platform’s design and functionality will be driven by user needs and market trends, ensuring continuous improvement and innovation.

### 2.1.3. Definitions, Acronyms and Abbreviation

This subsection provides the definitions of all terms, acronyms, and abbreviations required to properly interpret the Software Requirements Specification (SRS). This information is essential for ensuring that all stakeholders have a common understanding of the terms used throughout the document.

#### 2.1.3.1. Definitions

Admit Auction Website: An online platform where users can list, bid on, and purchase admission tickets through an auction process.

Bid: An offer made by a user indicating the price they are willing to pay for a ticket.

Auction: A process in which items (in this case, admission tickets) are sold to the highest bidder.

User: Any individual or entity that interacts with the Admit Auction Website, including buyers, sellers, and administrators.

Seller: A user who lists tickets for auction on the website.

Buyer: A user who places bids on tickets listed for auction.

Administrator: A user with special privileges to manage and maintain the website, including overseeing auctions and user activities.

Reserve Price: The minimum price that a seller is willing to accept for an auctioned ticket.

Buy Now Option: A feature allowing a buyer to purchase a ticket immediately at a fixed price, bypassing the auction process.

Auction Duration: The specified time period during which an auction is active and bids can be placed.

#### 2.1.3.2. Acronyms and Abbreviations

-API: Application Programming Interface

-DBMS: Database Management System

-HTTP/HTTPS: Hypertext Transfer Protocol/Secure, used for secure web communications.

-SQL: Structured Query Language

-SRS: Software Requirements Specification

-UI: User Interface

-UX: User Experience

### 2.1.4. Overview

AdmitAuction, established in 1970, has been a prominent auctioneer in South Australia, specializing in auctions for government salvage, company liquidation, and special personal items like paintings and jewelry. Facing intense competition from online auctioneers, AdmitAuction seeks to modernize its operations through the development of SuperEBid, an online auction platform. This platform will cater to existing corporate sellers, individual buyers, and new clients across different demographics and geographic regions.

The SuperEBid system aims to achieve the following objectives:

1. Provide a Transparent and Fair Auction Process:

- Ensure that all users have an equal opportunity to participate in auctions, with transparent bidding processes and real-time updates. Implement features such as bid history and current highest bid visibility to maintain transparency.

1. Maximize Seller Profits:

- Enable sellers to reach a broad audience, increasing the potential for higher final bid prices. Offer tools for sellers to effectively manage their listings, such as scheduling start and end times, setting reserve prices, and using promotional features.

1. Enhance Buyer Experience:

- Provide buyers with a wide selection of items, intuitive search and filter options, and detailed item descriptions and images. Ensure a smooth bidding process with features like automatic bidding and notifications for bid status changes.

1. Ensure Secure Transactions:

- Implement secure payment gateways to protect both buyers and sellers. Utilize robust authentication and authorization mechanisms to safeguard user accounts and personal information.

1. Facilitate User Engagement and Satisfaction:

- Design an intuitive and responsive user interface that works seamlessly across various devices, including desktops, tablets, and smartphones. Offer customer support and dispute resolution services to handle any issues that may arise during the auction process.

1. Leverage Modern Web Technologies:

- Build the platform using scalable and maintainable technologies to handle high traffic and ensure reliable performance. Implement data analytics and reporting tools to provide insights into user behavior, auction performance, and site usage.

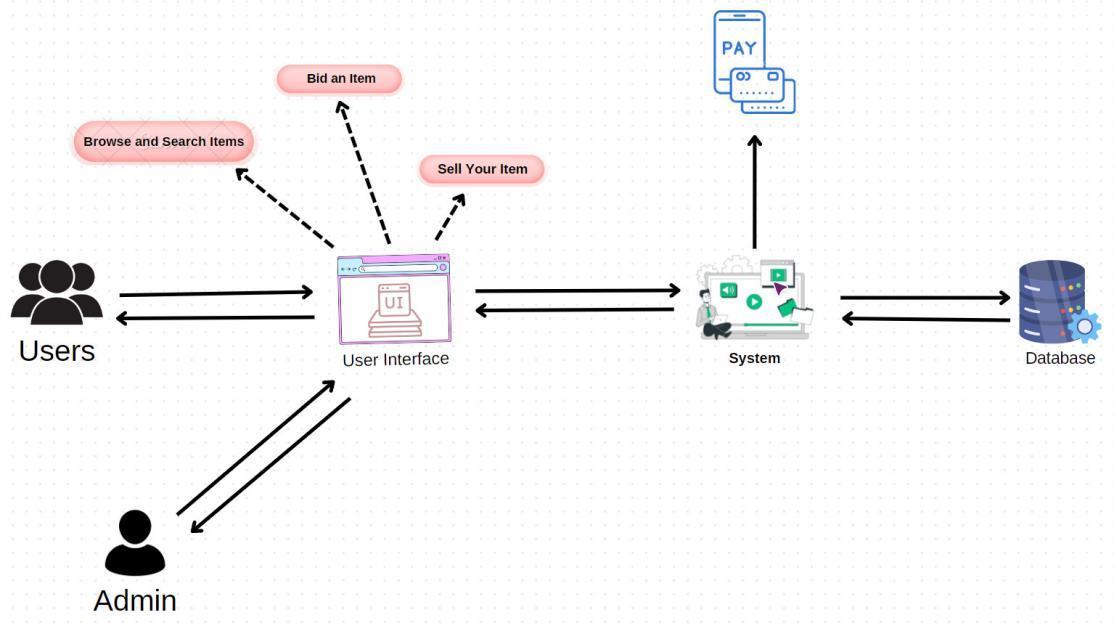
By achieving these objectives, the Admit Auction Website aims to become a leading online auction platform that attracts a diverse user base, fosters a thriving marketplace, and delivers a high level of satisfaction to both buyers and sellers. The platform’s design and functionality will be driven by user needs and market trends, ensuring continuous improvement and innovation.

## 2.2. Overall Description

### 2.2.1. Product perspective

SuperEBid is designed to modernize AdmitAuction's traditional auction methods with a web-based platform, interfacing with external systems for payment processing and data management. It will interact with credit card systems for payment transactions and an MS-SQL 2000 database for data storage. The user interface includes screens for member registration, browsing/searching items, bidding, and profile management, ensuring intuitive navigation and easy user interaction. Compatibility with web browsers like MS Internet Explorer and Netscape Navigator is required. Software and hardware interfaces are detailed to support efficient operations, alongside secure network protocols and memory constraints. The system will support both user-initiated and unattended operations with robust backup and recovery plans. Site adaptation includes importing initial data and configuring settings to comply with local regulations and user preferences.

Below is a conceptual block diagram illustrating the major components and interfaces of the SuperEBid system:



###### Figure 2.1 System perspective diagram

### 2.2.2. Product functions

Given below are the major functions that can be performed using the AdmitAuction system.

The system will:

- Allow users to register by providing necessary details including username, address, telephone, email, account name, password, and credit card information.

- Verify the credit card details with the credit card system.

- Send an email to the user with instructions to complete the registration and verify the email address.

- Provide a list of categories for users to explore.

- Display subcategories and a list of items available for bidding within each subcategory.

- Show details such as initial bidding price, time left for bidding, number of bids, current bidding price, bidding history, and highest bidder.

- Display comments and ratings for sellers and bidders.

- Allow users to search for items using various criteria like basic search, by seller, by bidder, or by shop.

- Retrieve and display a list of items available for bidding based on the search criteria.

- Allow users to place bids on items after browsing or searching.

- Require users to enter a maximum bidding price, ensuring it is higher than the current bidding price.

- Refresh the screen every 3 minutes to show the latest data.

- Allow users to register as sellers by providing necessary details and credit card information.

- Validate the credit card during registration.

- Send instructions on how to sell items via email or postal mail.

- Allow registered sellers to list items for sale by selecting categories and subcategories.

- Require sellers to provide item descriptions, duration, minimum price, pictures, payment, and postage details.

- Display item details for confirmation before listing.

- Allow users to view their buying and selling history, member profile, and feedback.

- Enable users to participate in the feedback forum, respond to feedback, and manage their feedback settings.

- Allow sellers to place their credit card on file for automatic monthly payments.

- Automatically charge the seller's credit card for their invoice amount each month.

- Automatically generate monthly invoices for seller charges.

- Notify sellers about their charges and allow them to view their balance under MyProfile.

- Allow administrators to log in to the system with administrative privileges.

- Allow administrators to withdraw items from auction that violate auction policies.

- Enable administrators to analyze sellers' activities, including statistics on age, selling preferences, price range, and location.

- Enable administrators to analyze buyers' activities, including statistics on age, shopping preferences, price range, and location.

### 2.2.3. User characteristics

**Sellers:**

-This kind of user is looking to sell new, near-new, or run-out products.

-Needs: Easy listing process; secure payment processing; ability to track sales and receive timely payments.

-Tech Savviness: High, comfortable with online platforms for selling goods.

**Customer:**

-These kinds of users have personal profiles tracking their buying history.

-Feedback and ratings from transactions help build a trusted community.

-Availability of FAQs, online tutorials, and support for addressing user queries.

**Administrators:**

-Responsible for managing the platform and analyzing user activity. They need comprehensive management tools.

-Strong technical and management skills.

-Need tools for analysis and reporting.

-Oversee compliance and user support.

### 2.2.4. General constraints

The development and operation of the SuperEBid system will be guided by several constraints. The system must comply with local and international regulations, including consumer protection, tax, and data privacy laws, as well as financial regulations like PCI DSS. Hardware must support server specifications and real-time updates. Integration with external credit card processing and email systems is essential. SuperEBid must handle multiple concurrent users and database transactions without performance issues, and log all transactions for audit purposes. Administrative and user controls are required for auction management and user profile management. The system should use a higher-order programming language and implement reliable network protocols. High uptime (99.9%) and fault tolerance are critical for continuous operation. Safety and security measures, including data encryption, access controls, and regular audits, must be enforced. Addressing these constraints will ensure SuperEBid's compatibility, performance, reliability, and security, enabling a successful transition to a modern web-based auction platform.

### 2.2.5. Assumptions and dependencies

This subsection of the SRS should list each of the factors that affect the requirements stated in the SRS. These factors are not design constraints on the software but are, rather, any changes to them that can affect the requirements in the SRS. For example, an assumption may be that a speciÞc operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the SRS would then have to change accordingly.

#### 2.2.5.1. Assumptions (OS, Internet, Browser, Frameworks, Payment)

**Availability of Operating System and Hardware:**

- Assumption: The Auction Ticket Website will operate on common operating systems such as Windows, macOS, and Linux.

- Impact if Changed: If a specific operating system is not available or incompatible, the requirements for development and deployment of the software will need to be adjusted to be compatible with an alternative operating system.

**Internet Access and Bandwidth:**

- Assumption: Users will have stable internet access with sufficient bandwidth to load and interact with the website.

- Impact if Changed: If users have poor internet connectivity, the user interface and user experience may be affected, requiring a review and possible adjustment of performance and page load requirements.

**Browser Support:**

- Assumption: The website will be supported on common browsers such as Chrome, Firefox, Safari, and Edge.

- Impact if Changed: If one or more browsers are no longer supported or are incompatible, the requirements for user interface development and functionality need to be adjusted to ensure compatibility and a good user experience.

**Technology and Software Frameworks:**

- Assumption: Technologies and software frameworks such as HTML, CSS, JavaScript, and frameworks like React or Angular will be used.

- Impact if Changed: If there are changes in technology or software frameworks, the requirements for design, development, and testing may need to be updated to align with the new technology.

**Payment and Third-party Services Integration:**

- Assumption: Payment services and third-party services like PayPal, Stripe will be available and functioning normally.

- Impact if Changed: If these services are unavailable or malfunction, the requirements for payment integration and related functions will need to be reviewed and adjusted.

#### 2.2.5.2. Dependencies

**Dependency on Web Service Providers:**

- Dependency: The availability and performance of the chosen web service provider for hosting the website.

- Impact if Changed: If the web service provider experiences issues or does not meet performance requirements, the requirements for infrastructure and system architecture may need to be changed to find an alternative provider or improve performance.

**Dependency on Database Systems:**

- Dependency: Using a specific database management system like MySQL or PostgreSQL.

- Impact if Changed: If the database system encounters scalability issues or does not meet performance requirements, the design and implementation requirements for the database may need to be adjusted.

**Dependency on Notification and Email Services:**

- Dependency: Using services such as SMTP, SendGrid to send notifications and emails.

- Impact if Changed: If the notification or email services are unavailable or malfunctioning, the requirements for user communication and notifications will need to be adjusted to use an alternative service.

**Dependency on Regulations and Policies:**

- Dependency: Compliance with regulations and policies related to security, data protection, and online transactions.

- Impact if Changed: If there are changes in regulations or policies, the requirements for security, privacy, and data management will need to be updated to ensure compliance.

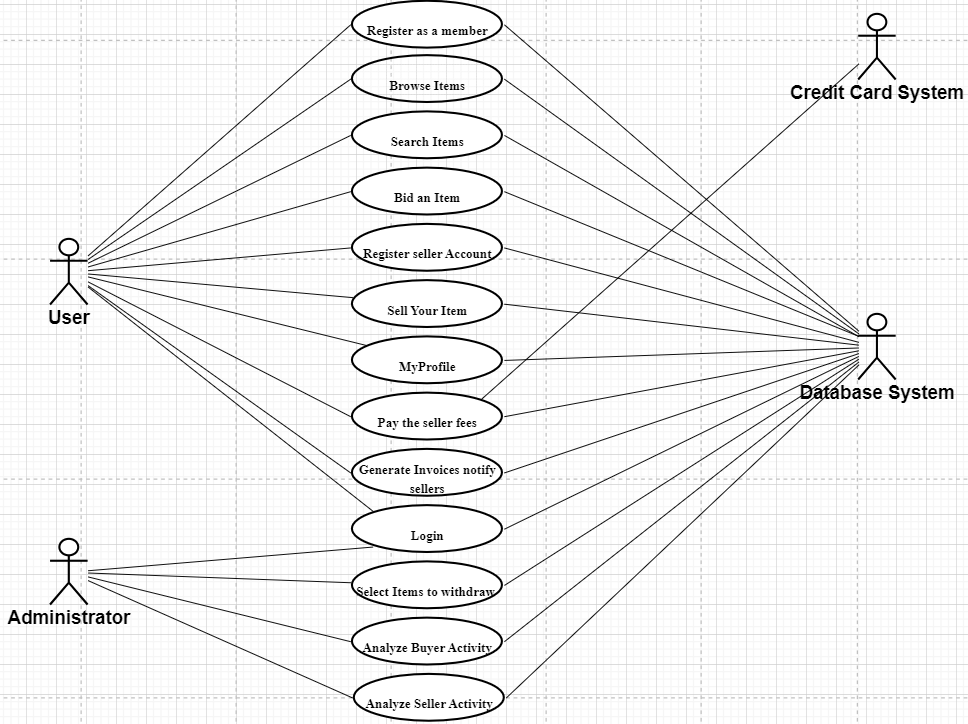
**Dependency on Development Team:**

- Dependency: The availability and skills of the software development team.

- Impact if Changed: If the development team lacks the necessary skills or resources, the project timeline and quality may be affected, requiring adjustments to the requirements for time and resources.

**2.3. Specific Requirements**

### 2.3.1. Use case diagram



**Figure 2.2 The general use case diagram**

### 2.3.2. Functional requirements

#### 2.3.2.1. Primary Requirements

##### 2.3.2.1.1. Register as a member

If the user chooses the Register button to register as a member, the system will display a series of screens to capture the user information. The system requires the user to provide the user name, address, telephone, email, account name, password, credit card information. The system will verify the user credit card with the credit card system. If the details are ok, the system will send an email to the user with an instruction and the user needs to access the email and respond to that to finish the registration. In doing that, the system verifies the validity of the member email address.

**Use case report:**

**a, Use case name:** Register as a member

**b, Brief description:**

- This use-case allows users to register as a member.

**c, Flow of events**

**\*Basic flow:**

1. User chooses the Register button:

-The use-case starts when the user chooses the Register button, the system will display a series of screens to capture the user information.

1. User selects type of register:User selects some details to register:

-The system requires the user to provide the user name, address, telephone, email, account name, password, credit card information.

-The system will verify the user credit card with the credit card system.

1. User verifies to finish the registration.

- If the details are ok, the system will send an email to the user with an instruction and the user needs to access the email and respond to that (in doing that, the system verifies the validity of the member email address).

**\*Alternative flows:**

1. SQL database system unavailable:

- If the system cannot connect SQL database system, the system will display the error message.SQL database system unavailable:

1. Account name is duplicated:

- User, unaware that the account name already existed, chooses the Register button. The system will display a message about the duplicated and refused register.User, unaware they are already registered, chooses to register as a seller. The system detects that the user's details match an existing seller account and displays a message.

**d, Special requirements**

**e, Pre-conditions**

- User has a valid email address or postal address.

- Users must have internet connection.

**f, Post-condition**

- User is registered as a member.

- User receives instructions on how to sell via email or postal mail.

- Credit card information is validated.

##### 2.3.2.1.2. Browse Items

The system provides a number of categories. The user can click into a category, the system will retrieve a list of subcategories on screen. The user can select a subcategory, the system will display a list of items that are available for bidding. The system will show the initial bidding price, the time left before close for bidding, number of bids, the current bidding price. You can see the bidding history and the higher bidder. The user can see the comments for the sellers which are given by the buyers in terms of their delivery, product etc. Users also can see the comments of the bidders which are given by the sellers. In doing that, users should have some information to form a picture about the bidder’s profile when deciding to place their bids. If the comments about the seller are not good, the buyer has the choice of not to buy.

**Use case report:**

**a, Use case name: Browse Items.**

**b, Brief description:**

- The system allows users to select one or more items that are available for bidding.

**c, Flow of events**

**\*Basic flow:**

1. User chooses a category:

- The use-case starts when the user clicks into a category, the system will retrieve a list of subcategories and display the subcategory on screen.

1. User selects a subcategory:

- The system will display a list of items that are available for bidding. The system will show the initial bidding price, the time left before close for bidding, number of bids, the current bidding price.

1. User sees information of subcategory:

- You can see the bidding history and the higher bidder. The user can see the comments for the sellers which are given by the buyers in terms of their delivery, product etc. Users also can see the comments of the bidders which are given by the sellers.

**\*Alternative flows:**

1. SQL database system unavailable:

- If the system cannot connect to the SQL database system, the system will display the error message.

**d, Special requirements**

**e, Pre-conditions**

- User has a valid email address or postal address.

- Users must have internet connection.

**f, Post-condition**

-None

##### 2.3.2.1.3. Search Items

If the user clicks on the Search button, the system provides a set of search criteria for the user to choose. The user enters the search criteria such as basic search, by seller, by bidder or by shop. The system will retrieve from the database with a list of items currently available for bidding.

**Use case report:**

**a, Use case name:** Search Items for Bidding

**b, Brief description:**

- The use-case allows a user to search for items currently available for bidding using various search criteria such as Basic Search, By Seller, By Bidder, or By Shop.

**c, Flow of events**

**\*Basic flow:**

1. The use case begins when the user navigates to the "Search Items" page.
2. The user clicks on the "Search" button.
3. The system displays a set of search criteria options: Basic Search, By Seller, By Bidder, By Shop.
4. The user selects one of the search criteria options.
5. The system prompts the user to enter the relevant search details based on the chosen criteria.

- Basic Search: The user enters keywords related to the items.

- By Seller: The user enters the seller's name or ID.

- By Bidder: The user enters the bidder's name or ID.

- By Shop: The user enters the shop's name or ID.

1. The user submits the search query.
2. The system retrieves the list of items from the database that match the search criteria.
3. The system displays the list of items to the user.
4. The use case ends when the user reviews the search results. Special requirements

**\*Alternative flows:**

1. Invalid Search Criteria Selection:

In step 4 If the user selects an invalid or unavailable search criteria:

- The system displays an error message indicating the invalid selection.

- The user selects a valid search criteria and continues with the process.

1. No Items Found:

In step 6 If the system does not find any items matching the search criteria:

- The system displays a message indicating that no items were found.

- The user may choose to modify the search criteria or end the search process.

**d, Special requirements**

- The system should provide real-time search suggestions as the user types search criteria.

- The system should handle a large volume of items efficiently to ensure quick retrieval and display.

- The user interface should be intuitive and user-friendly to facilitate easy searching.

**e, Pre-conditions**

-There are items available in the database for bidding.

**f, Post-condition**

-None

**g, Extension point**

- Integration with advanced search functionalities like voice search or image search can be considered.

- The search results can be further enhanced with sorting and filtering options based on user preferences.

##### 2.3.2.1.4. Bid an item

After the user has used the Browse Items or Search Items function, a list of items are shown on screen with their initial or current bidding price etc. The user double clicks on an item, the detailed item information will be shown. The user has the option to choose to place a bid or not. The use case starts when the user chooses to bid. The user will need to enter their maximum bidding price. The system will ensure the price entered is bigger than the current bidding price. In order that the user is shown with the latest data on screen, the screen is refreshed every 3 minutes by re-enquiring the database.

**Use case report:**

**a, Use case name:** Bid an item

**b, Brief description:**

- The use-case allows a user to place a bid on an item after browsing or searching for available items. The user can view detailed information about the item and enter a maximum bidding price. The system ensures the entered bid is higher than the current bid and updates the bid data accordingly.

**c, Flow of events**

**\*Basic flow:**

1. The use case begins when the user double-clicks on an item from the list of items displayed after using the Browse Items or Search Items function.
2. The system displays detailed information about the selected item, including the current bidding price and other relevant details.
3. The user chooses the option to place a bid.
4. The system prompts the user to enter their maximum bidding price.
5. The user enters their maximum bidding price.
6. The system verifies that the entered bid is higher than the current bidding price.
7. If the entered bid is higher, the system accepts the bid and updates the current bidding price.
8. The system confirms to the user that their bid has been successfully placed.
9. The screen is refreshed every 3 minutes to ensure the user sees the latest bidding data by re-enquiring the database.
10. The use case ends when the user is shown the updated bidding status.

**\*Alternative flows:**

1. Invalid Bid Amount:

In step 6 If the entered bid is not higher than the current bidding price:

- The system displays an error message indicating that the bid must be higher than the current bid.

- The user is prompted to enter a valid bid amount.

1. System Error:

In step 7 If there is an error updating the bid in the system:

- The system displays an error message indicating that the bid could not be placed due to a system error.

- The user may choose to retry placing the bid or cancel the bid process.

**d, Special requirements**

- The system must refresh the screen every 3 minutes to display the latest bidding data.

- The system should handle concurrent bids and ensure data consistency.

- The user interface should provide clear feedback messages for both successful and unsuccessful bid attempts.

**e, Pre-conditions**

- The user is logged into the system.

- The user has used the Browse Items or Search Items function and has a list of items displayed on the screen.

**f, Post-condition**

- The user's bid is successfully placed and the current bidding price is updated.

- The user sees the updated bidding status on the screen.

**g, Extension point**

- Integration with a notification system to alert users when they are outbid on an item.

- Option for users to set automatic incremental bidding up to a specified maximum amount.

- Implementation of real-time updates for bid status without the need for a 3-minute refresh interval.

##### 2.3.2.1.5. Register Seller Account

If the user chooses the Sell button, the system asks whether the user is a registered seller or not. If the user is not a registered member, the user can choose to register as a member. Otherwise, the user can login and can go through the Seller Account screen to enter the details of their credit card or postal address, pay fee method etc. e\_Bay will email or send to the postal address of the seller with the instructions how to sell. The system will check with the credit card system to validate the credit card at the time of registration.

**Use case report:**

**a, Use case name:** Register Seller Account

**b, Brief description:**

- This use-case allows users to register by seller account.

**c, Flow of events**

**\*Basic flow:**

1. User selects Sell button from menu:

- The use-case starts when the user selects the Sell button from the menu (screen). The system will display the optional selection.

1. User selects type of register:

- User selects whether to register as Seller or not (choose to log in as member), the system will receive the request and move to the next step.

1. Users choose login as Seller:

- When the user chooses "Register as a Seller", the system will present a form for the user to enter the details of their credit card or postal address, pay fee method etc.

1. User enters all the information:

- After the user has entered all the information, the system emails or sends it to the postal address of the seller with instructions on how to sell. At the same time, the system will check with the credit card system to validate the credit card at the time of registration.

**\*Alternative flows:**

1. User Clicks "Sell" Button While Already Logged In:

- User is already authenticated on the platform. The System recognizes the user is logged in and skips the prompt for seller status.

1. SQL database system unavailable:

- If the system cannot connect to SQL database system, the system will display the error message.

1. User Chooses to Register but is an Existing Seller:

- User, unaware they are already registered, chooses to register as a seller. The system detects that the user's details match an existing seller account and displays a message.

**d, Special requirements**

**e, Pre-conditions**

- Users have access to the e\_Bay platform.

- User has a valid email address or postal address.

- Users must have internet connection.

**f, Post-condition**

- User is registered as a seller.

- User receives instructions on how to sell via email or postal mail.

- Credit card information is validated.

**g, Extension point**

- If user is not a seller, he/she can select login as member

##### 2.3.2.1.6. Sell your item

If the user has a seller account, the seller is allowed to list an item. The user will need to choose the category and subcategory for the item they want to sell. When the user chooses a category, the system retrieves the list of available subcategories from the database to help the user to select. The system will then ask the seller to describe the item such as duration, minimum price. The seller will provide the picture and details of the items in a series of screens. The user also needs to indicate the layout, the payment and postage cost and countries will go etc. The system will display a screen that will provide the item details for confirmation before the item is listed.

**Use case report:**

**a, Use case name:** Sell Your Item

**b, Brief description:**

- This use-case allows users to sell their item through the system.

**c, Flow of events**

**\*Basic flow:**

1. User login seller account:

- If the user has a seller account, the system will retrieve the information and allow users to list an item.

1. User chooses the category to sell:

- When the user chooses a category, the system will retrieve the list of available subcategories from the database to help the user to select and request the user to describe the item such as duration, minimum price.

1. Users enter all the information about item:

- The seller will provide the picture and details of the items in a series of screens, the layout, the payment and postage cost and countries will go etc. The system will display a screen that will provide the item details for confirmation before the item is listed.

1. User enters all the information:

- After the user has entered all the information, the system emails or sends it to the postal address of the seller with instructions on how to sell. At the same time, the system will check with the credit card system to validate the credit card at the time of registration.

**\*Alternative flows:**

1. Seller Cancels Listing Process:

- At any point during the listing process, the seller chooses to cancel. The system prompts the seller to confirm the cancellation. The listing process is aborted, and no details are saved if the user selects “YES”, and back to the previous step if choosing “NO”.

1. SQL database system unavailable:

- If the system cannot connect to SQL database system, the system will display the error message.

1. Invalid or Missing Information:

- The seller enters incomplete or invalid information in any of the fields. If any information is invalid or missing, the system displays an error message indicating the specific issue.

**d, Special requirements**

**e, Pre-conditions**

- User must be a registered seller and logged into their account.

- Users must have internet connection.

**f, Post-condition**

- The item is successfully listed for sale on the platform.

- The listing details are stored in the system database.

**g, Extension point**

-If the user misses some information or lacks detail, they can cancel the process or go back to the previous step.

##### 2.3.2.1.7. My profile

The user is expecting to have similar function like the **My eBay**. The system will display all of the buying and selling history. Users would be able to review their member profile and their feedback if they choose tab My Profile. If user chose tab Community, they will be able to join the feedback forum so that they can reply to their feedback received, follow up to feedback left, hide their feedback and mutual feedback withdrawal etc, system access the database to save these information. Every eBay member has a profile in the Feedback Forum. A profile has basic information about the member and a list of feedback left by their trading partners from previous transactions. For each transaction, only the buyer and seller can rate each other by leaving feedback, the system ensures that. When user choose give feedback system display a screen for feedback consists of a positive, negative, or neutral rating, and a short comment, user must fill all of these information. Leaving honest comments about a particular eBay member gives other Community members a good idea of what to expect when dealing with that member. Once it is left, the system take it and the feedback becomes a permanent part of the member's profile in the database.

**Use case report:**

**a, Use Case Name:** My Profile

**b, Brief Description:**

-This use case extends the "My Profile" functionality to include features similar to My eBay, allowing users to view their buying and selling history, manage feedback, and participate in a feedback forum.

**c. Flow of Events**

**\*Basic Flow:**

(1) Select "My eBay" Option

- User selects the "My eBay" option from the main menu.

- System displays the user's buying and selling history.

(2) Access "My Profile" Tab

- User selects the "My Profile" tab.

- System displays the member's profile, including basic information and feedback history.

(3) Access "Community" Tab

- User selects the "Community" tab.

- System displays options to join the feedback forum, reply to feedback, follow up on feedback, hide feedback, and mutual feedback withdrawal.

(4) Join Feedback Forum

- User opts to join the feedback forum.

- System confirms and grants access to the feedback forum features.

(5) Leave Feedback

- User selects an option to leave feedback for a transaction.

- System displays a feedback form with options for a positive, negative, or neutral rating, and a short comment field.

- User fills out the feedback form and submits it.

- System validates the feedback form ensuring all fields are filled.

- System updates the member’s profile with the new feedback and saves it in the database.

**\*Alternative Flows:**

(1) Empty Feedback Form Submission

- If the user submits an incomplete feedback form, the system prompts the user to fill all required fields.

(2) Feedback Reply and Follow-Up

- User selects an option to reply to feedback received or follow up on feedback left.

- System displays a form for replying or following up.

- User submits the reply/follow-up.

- System updates the feedback history accordingly.

(3) Hiding Feedback

- User selects an option to hide certain feedback.

- System processes the request and updates the feedback visibility.

**d, Special Requirements**

- The system should securely store and display transaction history details.

- Feedback once submitted should be immutable.

- Only involved transaction parties (buyer and seller) can leave feedback for each other.

- Feedback visibility controls and mutual feedback withdrawal must follow specific community guidelines and policies.

**e, Pre-Conditions**

- The user has a valid account with the system.

- The user is successfully logged in.

**f, Post-Conditions**

- The user's profile page reflects any updates made to transaction history, feedback, or other profile information.

**g, Extension Points**

- Implement a notification system to alert users of new feedback received.

- Allow users to filter and search their transaction history.

- Integrate reputation scores based on user feedback.

##### 2.3.2.1.8. Pay the seller fees

Sellers will place their credit card (Visa or MasterCard) on their account for regular monthly payments. Each month AdmitAuction will automatically charge the seller invoice amount directly from their credit card on file. The credit card will normally be charged 7 to 10 days after receipt of the invoice

**Use case report:**

**a, Use Case Name:** Pay Seller Fees

**b, Brief Description:**

-The "Pay Seller Fees" use case enables sellers on AdmitAuction to securely store their Visa or MasterCard credit card information for automatic monthly payments.

**c, Flow of Events**

**\*Basic Flow:**

(1) User Login

- User logs into their AdmitAuction account.

- System authenticates the user and displays the account dashboard.

(2) Navigate to Billing Section

- User navigates to the "Account Settings" or "Billing" section.

- System displays the billing and payment information page.

(3) Add or Verify Credit Card Information

- User adds new credit card information (Visa or MasterCard) or verifies existing credit card information.

- System validates the credit card details and securely saves the information.

(4) Invoice Generation and Notification

- At the end of each month, the system generates an invoice for the seller based on their activities.

- System sends an invoice notification to the seller's registered email address.

(5) Hold Invoice and Charge Credit Card

- System holds the invoice for 7 to 10 days after sending the notification to the seller.

- After the hold period, the system automatically charges the seller's credit card on file for the invoice amount and records the payment transaction details in the database.

**\*Alternative Flows:**

(1) Adding or Updating Credit Card Information Fails

- User enters new credit card information or updates existing card details.

- System attempts to validate the credit card information and finds an error (e.g., invalid number, expired card).

- System displays an error message indicating the issue with the credit card information.

- User corrects the credit card information and resubmits.

- System validates and saves the corrected credit card information securely.

**d, Special Requirements**

- The system must ensure secure handling of credit card information.

- The system should provide timely notifications for invoices, payment confirmations, and payment failures.

**e, Pre-Conditions**

- The user has a valid account with the system.

- The user is successfully logged in.

- The user must have a valid Visa or MasterCard credit card added to their account.

- The user must have an outstanding invoice generated by the system.

- The system must have sent the invoice notification to the user.

**f, Post-Condition**

- The seller's credit card is charged the invoice amount.

- The seller receives a payment confirmation notification.

- The seller's account is updated to reflect the paid status of the invoice.

**g, Extension Points**

- Sellers place their credit card (Visa or MasterCard) on their account for regular monthly payments.

- Each month, AdmitAuction automatically charges the seller's invoice amount directly from their credit card on file.

- The credit card is normally charged 7 to 10 days after receipt of the invoice.

##### 2.3.2.1.9. Generate Invoices and notify seller

The system will automatically generate the invoice for the seller related to their seller charges every month. A notice will be sent to the seller to remind them about the charges. The seller can signin under MyProfile to see their monthly balance.

**Use case report:**

**a, Use case name:** Generate Invoices and notify seller

**b, Brief description:**

- This use case generates monthly invoices and notifies sellers

**c, Flow of events**

**\*Basic flow:**

1. Generate Monthly Invoice

- The use case begins with the system generating monthly invoices for all sellers based on their selling charges.

- The system calculates the total charges and creates an invoice.

(2) Notify Seller

- The system sends a notification email to each seller, informing them of their monthly charges. The email contains a summary of the invoice and a link to log in to their MyProfile to view detailed information.

(3) Seller Sign-In to MyProfile

- Sellers log into their MyProfile page to view their monthly balance and detailed invoice. The system displays the detailed invoice information such as itemized charges and total balance.

**\* Alternative Flows:**

(1)Email Notification Failure

- If the system fails to send the notification email, an error log is created. The use case attempts to resend the notification up to a specified number of times. After repeated failures, the system logs the issue and alerts the administrator.

(2) Database Unavailability

- If the system cannot connect to the database to retrieve seller charge data, an error message is logged. The use case will end and an alert is sent to the admin for immediate attention.

(3) Quit

- Sellers have the option to quit viewing their profile or invoice details at any time. The system saves the current state and the use case ends.

**d, Special Requirements**

- The system must ensure that the email notifications are sent securely to protect sensitive seller information.

- The invoice generation process must be optimized for large datasets to prevent performance degradation.

**e, Pre-Conditions**

- Sellers must have valid email addresses registered in the system.

- The system database must contain up-to-date seller charge information for the billing period.

**f, Post-Conditions**

- Sellers are notified about their monthly charges via email.

- Invoices are accurately generated and stored in the seller’s profile for review.

- System logs are updated with the status of email notifications and invoice generation.

**g, Extension Points**

Add to Shopping Basket:

- Sellers can add any charges or discounted services directly from the invoice page to their shopping basket for future transactions.

Add to Shopping List:

- Sellers can add recurring charges to their shopping list for easier future transactions and budgeting.

#### 2.3.2.2. Secondary Requirements

##### 2.3.2.2.1. Login

The administrator can login to the system as an administrator to manage and oversee the auction platform. This process involves entering valid credentials, which the system then verifies against stored administrator account information. If the login is successful, the administrator gains access to the system's administrative functions. If the login fails, appropriate error messages are displayed to guide the administrator.

**Use-case Report:**

**a, Use-case name:** Login

**b, Brief description:**

- This use case allows the administrator to log in to the system with their credentials, granting access to administrative functions and tools necessary for managing the auction platform.

**c, Flow of events:**

**\*Basic flow of events:**

(1) The administrator accesses the login:

- The administrator navigates to the login page and enters their username and password. The system checks the credentials against stored administrator account information.

(2) System logs the login attempts:

- The system logs all login attempts, including details such as the time of the attempt, the IP address from which the attempt was made, and the outcome (success or failure). This is done for security and auditing purposes.

**\*Alternative flows:**

(1) SQL database system unavailable:

- If the system cannot connect to the SQL database, an error message is displayed to the administrator, informing them of the issue and suggesting they try again later or contact technical support.

(2) Administrator cancels login processing:

- If the administrator decides to cancel the login process (e.g., by clicking a "Cancel" button), the system clears the login form fields (username and password) and returns the administrator to the main page or a neutral starting point within the system.

(3) Administrator account locked:

- If an administrator account is locked due to exceeding the allowed number of failed login attempts, the system displays an error message explaining the situation and providing instructions for unlocking the account, such as contacting technical support.

**d, Special requirements:**

- The system must ensure the security and privacy of administrator login details.

- The system should be able to handle a high volume of login attempts without performance degradation.

- The login process should comply with industry standards for security, including encrypted communication and secure storage of credentials.

**e, Pre-conditions:**

- The administrator must have an active internet connection.

- The administrator must have valid credentials (username and password).

- The system's SQL database must be available and accessible.

**f, Post-condition:**

- Upon successful administrator login, the system grants access to the administrator dashboard and relevant management tools.

- Upon failed login attempts, the system displays an appropriate error message indicating invalid credentials or other relevant issues.

**g, Extension points:**

- Implement multi-factor authentication (MFA) for added security.

- Integrate with single sign-on (SSO) solutions for seamless authentication across multiple systems.

##### 2.3.2.2.2. Select Items to withdraw from auction

The administrator can choose the items to be withdrawn from auction. This functionality is crucial for maintaining the integrity and legality of the auction process. Items might need to be withdrawn if they infringe on the auction's practices, such as being illegal, misrepresented, or violating platform policies. The system provides tools for the administrator to identify, review, and remove such items to ensure that all listed items comply with legal and auction house standards.

**Use-case Report:**

**a, Use Case Name:** Select Items to withdraw from auction

**b, Brief description:**

- This use case allows the administrator to select items to withdraw from the auction. Items may need to be withdrawn if they are found to be illegal, misrepresented, or otherwise in violation of auction house policies. The process ensures that the auction platform remains compliant with legal standards and maintains its integrity.

**c, Flow of Events:**

**\*Basic Flow:**

(1) Administrator selects items to check:

- The administrator navigates to the item management section and selects items for review. The system retrieves the relevant item information from the database.

(2)User chooses the category to withdraw:

- The administrator reviews the item details to determine if it meets legal and auction house standards. If an item is found to be non-compliant, the administrator selects it for withdrawal. The system removes the item from the active auction list and updates the database accordingly.

**\*Alternative Flows:**

(1)Administrator Cancels Withdrawing Process:

- At any point during the withdrawal process, the administrator may choose to cancel. The system prompts the administrator to confirm the cancellation. If the administrator confirms, the process is aborted, no items are withdrawn, and the system returns to the previous state.

(2) SQL database system unavailable:

- If the system cannot connect to the SQL database, an error message is displayed, informing the administrator of the issue. The administrator may retry the operation later or contact technical support.

**d, Special Requirements:**

- The system must ensure that only authorized administrators can withdraw items from the auction.

- Detailed logging of all withdrawal actions for auditing and accountability purposes.

**e, Pre-conditions:**

- The administrator must have an active internet connection.

- The system database must contain accurate and up-to-date item data.

**f, Post-conditions:**

- The auction platform reflects the removal of specific items from the auction list.

- The system logs the withdrawal action for future reference.

**g, Extension Points:**

- If the administrator determines that an item is compliant after initial consideration, they can cancel the withdrawal process at any time.

- Integrate a notification system to inform sellers of the withdrawal of their items and the reasons behind the action.

##### 2.3.2.2.3. Report on the Sellers' Business Activities

Administrators can analyze sellers' activities, obtaining statistics such as age demographics, selling preferences, price ranges, and geographic locations. This process involves setting criteria, generating detailed reports, and exporting them for further analysis. These reports help in understanding seller behavior, aiding in informed decision-making and platform management.

**Use Case Report**

**a, Use Case Name:** Analyze Seller Activity

**b, Brief Description:**

- This use case allows administrators to analyze the activity of sellers on the platform. This analysis can include generating statistics on demographics, selling preferences, pricing strategies, and location distribution.

**c, Flow of Events:**

**\*Basic Flow:**

(1) Administrator Login:

- The administrator logs in to the system using their credentials. The system verifies the login and grants access to seller activity analysis tools.

(2) Select Analysis Criteria:

The administrator defines the desired analysis criteria, such as:

- Seller age range

- Preferred selling categories

- Price range of listed items

- Geographic location of sellers

(3) Generate Report:

- Based on the chosen criteria, the system retrieves relevant data from the database. The system processes and analyzes the data to generate reports.

(4) View and Export Reports:

- The system displays the generated reports to the administrator. Reports can include charts, graphs, and tables summarizing seller activity. The administrator can export the reports in various formats (e.g., PDF, CSV) for further analysis or sharing.

**\*Alternative Flows:**

(1) Invalid Login:

- If the administrator enters incorrect login credentials, the system displays an error message and prompts them to re-enter the information.

(2) Database Unavailable:

- If the system cannot connect to the database, an error message is displayed. The administrator can retry later or contact technical support.

(3) Insufficient Data:

- If insufficient data exists for the chosen criteria, the system informs the administrator and may suggest alternative criteria or a broader timeframe.

**d, Special Requirements:**

- The system should ensure data privacy, and only authorized administrators can access seller activity information.

- The analysis tools should be efficient and handle large datasets without significant delays.

**e, Pre-conditions:**

- The administrator is logged in to the system with appropriate permissions.

- The system database contains accurate and up-to-date seller activity data.

**f, Post-conditions:**

- The administrator gains insights into seller behavior through generated reports.

- Reports are readily available for viewing, exporting, and further analysis.

**g, Extension Points:**

- Implement drill-down functionality within reports to explore specific seller segments in greater detail.

- Integrate with data visualization tools for more interactive and customizable report formats.

- Allow exporting reports with options to filter or group data based on specific needs.

##### 2.3.2.2.4. Report on the buyer business activities

- Administrators can analyze buyers' activities by obtaining statistics on demographics, shopping preferences, price ranges, and geographic locations. This involves setting analysis criteria, generating detailed reports, and exporting them for further review. These insights help understand buyer behavior, making the platform more efficient and user-centric.

**Use Case Report:**

**a, Use Case Name:** Analyze Buyer Activity

**b, Brief Description:**

- This use case allows administrators to analyze the activity of buyers on the platform. This analysis can include generating statistics on demographics, shopping preferences, price range, and location distribution.

**c, Flow of Events:**

**\*Basic Flow:**

1. Administrator Login:

- The administrator logs in to the system using their credentials.

- The system verifies the login and grants access to buyer activity analysis tools.

1. Select Analysis Criteria:

The administrator defines the desired analysis criteria, such as:

- Buyer age range

- Preferred shopping categories

- Price range of purchases

- Geographic location of buyers

1. Generate Report:

- Based on the chosen criteria, the system retrieves relevant data from the database.

- The system processes and analyzes the data to generate reports.

1. View and Export Reports:

- The system displays the generated reports to the administrator. Reports can include charts, graphs, and tables summarizing buyer activity.

- The administrator can export the reports in various formats (e.g., PDF, CSV) for further analysis or sharing.

**\*Alternative Flows:**

1. Invalid Login:

- If the administrator enters incorrect login credentials, the system displays an error message and prompts them to re-enter the information.

1. Database Unavailable:

- If the system cannot connect to the database, an error message is displayed. The administrator can retry later or contact technical support.

1. Insufficient Data:

- If insufficient data exists for the chosen criteria, the system informs the administrator and may suggest alternative criteria or a broader timeframe.

**d, Special Requirements:**

- The system should ensure data privacy and only authorized administrators can access buyer activity information.

- The analysis tools should be efficient and handle large datasets without significant delays.

**e, Pre-conditions:**

- The administrator is logged in to the system with appropriate permissions.

- The system database contains accurate and up-to-date buyer activity data.

**f, Post-conditions:**

- The administrator gains insights into buyer behavior through generated reports.

- Reports are readily available for viewing, exporting, and further analysis.

**g, Extension Points:**

- Implement drill-down functionality within reports to explore specific buyer segments in greater detail.

- Integrate with data visualization tools for more interactive and customizable report formats.

- Allow exporting reports with options to filter or group data based on specific needs.

### 2.3.3. Nonfunctional requirements

#### 2.3.3.1. Privacy & security

- The system will show the privacy and security issues related to online auctioning appropriately.

#### 2.3.3.2. Frequently asked questions

- The system will provide this information to help the users to use the system more easily.

#### 2.3.3.3. Availability

- The customer must be able to access online 7 days 24 hours.

#### 2.3.3.4. Accuracy

- The application must provide accurate stock updates and customer order information.

#### 2.3.3.5. Response time

- A customer must receive status on their transaction within five seconds when a bid is placed.

#### 2.3.3.6. Interface Guideline

- The system must comply with the interface guideline which is defined by the design guide of the company.

#### 2.3.3.7. Training

- Before the delivery of the system to the user, there should be training provided to the user supporters who are responsible to answer questions from the users by email.

#### 2.3.3.8. Browser compliance

- The system should be able to run under MS Internet Explorer or Netscape Navigator.

#### 2.3.3.9. Online tutorial

- The system will provide an online tutorial for the user.

# Chapter 3: Conclusion

## 3.1. Achievements

Our project on "Applying the Prototyping Process Model to Specify Requirements for the Admit Auction Website" has resulted in significant achievements:

- Successful Application of Prototyping Process Model: We effectively applied the prototyping process model to specify the requirements for the Admit Auction website. Through iterative prototyping, we refined and finalized software requirements, ensuring alignment with stakeholders' expectations.

- Enhanced Technical Skills: Engaging in this project enhanced our technical skills in software development methodologies and requirements engineering. We gained practical experience in requirement elicitation, analysis, and documentation, enhancing our proficiency in these areas.

- Effective Team Collaboration: Throughout the project, we demonstrated effective teamwork and collaboration. By leveraging each team member's strengths and expertise, we successfully completed project milestones and deliverables within the specified timeframe.

## 3.2. Knowledge Gained

Our project facilitated the acquisition of valuable knowledge in several areas:

- Understanding of Prototyping Process: We deepened our understanding of the iterative nature of prototyping and its role in refining software requirements. This knowledge will be invaluable in future projects requiring requirements specification and validation.

- Requirement Engineering Concepts: Engaging in requirement elicitation and analysis activities familiarized us with fundamental requirement engineering concepts. We gained insights into gathering, documenting, and validating requirements, essential for developing high-quality software systems.

## 3.3. Skills Acquired

Additionally, our project helped us develop and refine several key skills:

- Communication and Stakeholder Engagement:\*\* Engaging in stakeholder communication honed our ability to effectively convey ideas, negotiate requirements, and address feedback. We improved our communication skills, facilitating productive interactions with stakeholders throughout the project lifecycle.

- Problem-Solving and Adaptability:\*\* Addressing technical challenges and adapting to changing project requirements honed our problem-solving and adaptability skills. We learned to approach complex problems systematically and explore innovative solutions to overcome obstacles encountered during the project.

- Time Management and Prioritization:\*\* Managing project timelines, tasks, and deliverables improved our time management and prioritization skills. We gained experience in allocating resources effectively, ensuring efficient project execution and timely delivery of outcomes.

Moving forward, we are committed to leveraging the knowledge and skills gained from this project to excel in future endeavors. We remain dedicated to continuous learning and improvement, striving for excellence in software engineering and beyond.

# REFERENCES

[1] Cockburn, A. (2000). Writing Effective Use Cases. Addison-Wesley Professional.

[2] IEEE Computer Society. (1998). IEEE Guide to Software Requirements Specifications. IEEE Computer Society Press.

[3] Introduction To Software Engineering Materials. Lecturer Ms. Nguyễn Thị Mỹ Bình. Hanoi University of Industry.

[4] Phùng Đức Hòa, Hoàng Quang Huy, Hoàng Văn Hoành, Nguyễn Đức Lưu, Trịnh Bá Quý. ( 2019 ). “*Giáo trình Nhập môn công nghệ phần mềm*”. Thống Kê Publisher.

[5] Robertson, S., & Robertson, J. (2006). Mastering the Requirements Process. Addison-Wesley Professional.

[6] Vũ Thị Dương, Phùng Đức Hòa, Nguyễn Thị Hương Lan. (2015 ). “Giáo trình Phân tích thiết kế hướng đối tượng”. Khoa học và Kỹ thuật Publisher.

[7] Wiegers, K., & Beatty, J. (2013). Software Requirements. Microsoft Press.