
ONLINE BIDDING SYSTEM

OBJECT ORIENTED SOFTWARE ENGINEERING

MOSIUOA WESI

324207316021

JUNE 24, 2025

**ANDHRA UNIVERSITY COLLEGE OF ENGINEERING
VISAKAPATNAM 530001**

Contents

MOSIUOA WESI.....	0
324207316021	0
1. Project Title & Description.....	4
Project Title: Online Bidding System	4
2. Problem Statement.....	4
3. Project Scope.....	4
4. Software Requirements	5
Scope	5
Description, Acronyms and Abbreviations	5
Overall description	5
Product Functions	5
User classes and characteristics.....	5
Design and Implementation.....	6
Specific Requirements	6
Functional Requirements.....	6
Non-functional Requirements.....	6
Participating Actors:	6
Interface Requirements	6
Conclusion	6
5. System Requirements for Online Bidding System.....	7
Overview	7
Hardware Requirements	7
Software Requirements.....	7
6. System Design Document For Online Bidding System	7
Scope.....	7
System Context	8
Architecture	8
Design Consideration	8
Assumption.....	8
Constraints.....	8
System Architecture	8
Components Diagram.....	8

6.	UML Diagrams	9
1)	Use Case Diagram.....	9
2.	Object Diagram	10
3.	Class Diagram	12
4.1	Sequence Diagram	13
4.	Collaboration Diagram.....	14
5.	Activity Diagram.....	15
6.	State Chart Diagram.....	17
7.	Component Diagram.....	18
8.	Deployment Diagram	19
7.	Test Specification for Online Bidding System	21
	Item Testing	21
	Test Specification.....	22
	Unit Testing	22
	System Testing.....	22
	Acceptance Testing	23
	Regression Testing	23
	Performace Testing.....	23
	Security Testing.....	23
	Usability Testing.....	23
	Compatibility Testing	24
	Smoke Testing	24
	Exploring Testing	24
12.	Manual /Guide for Online Bidding System	24
1.	User Manual.....	24
	Target Audience	24
2.	Programmer Guide	25
	Developers	25
	Contents	25
3.	Administrator Guide (Installation Instruction)	25
	Contents	25
13.	Project Plan and Schedule	26
1.	Project Schedule and Milestone	26

2.	Resource allocation	26
3.	Resource allocation	27
14.	Quality Plan : Setting Quality Assurance Procedures	27
	Purpose.....	27
	Components.....	27
	Quality Objective	27
	Standards and Guidelines	28
	Quality Assurance Procedures	28
	Quality Control Measures	28
	Roles and Responsibilities	28
	Documentation Requirements.....	28
	Review and Audit Plan.....	28
15.	Conclusion.....	29

1. Project Title & Description

Project	Title:	Online	Bidding	System
---------	--------	--------	---------	--------

The Online Bidding System is a web-based program created to automate and streamline online auctions, allowing users to display goods and place competitive real-time bids. It gives vendors a place to list their goods and choose how long an auction will go, and it allows consumers to bid. Only the highest bid at the conclusion of the auction will be the winner, according to the procedure. Key features of the system include real-time bid display, user authentication, auction status updates, and administrative management over users and listings. Through a more effective, safe, and user-friendly online experience, this technology seeks to replace conventional manual auction procedures.

2. Problem Statement

Traditional auctioning and bidding procedures are frequently limited by time, place, and manual coordination in today's digital economy, which reduces efficiency and participation. While bidders struggle to find auctions in real-time, ensuring transparency and placing competitive bids safely and securely, sellers struggle to attract a larger audience for their goods and services.

Furthermore, current solutions are either too expensive, inflexible or do not offer a user-friendly platform that would enable small enterprises or individuals to engage in competitive and equitable online bidding.

To address these limitations online bidding system will be developed to facilitate a dynamic, real-time bidding environment where sellers can list their items and bidders can compete in auctions through a secure scalable and intuitive platform. The system aims to provide roles such as admin, seller and bidder each with tailored functionalities. Core features include items listing, bid tracking, real-time updates, role-based access and bidding listing management.

The system leverages object-oriented design and software engineering patterns to ensure modularity, reusability and maintainability and integration.

3. Project Scope

The creation of a safe and effective platform that enables real-time bids from customers and product listings by sellers is part of the Online Bidding System's scope. The complete auction process is supported, including registration, product listing, bidding, automatic winner selection, and auction closure. In order to manage users, auctions, and platform settings, the system also offers administrative features. Real-time updates, time-bound auctions, bid validation, user roles and permissions, and guaranteeing data integrity throughout all transactions are important areas of concern.

4. Software Requirements

Software Requirement defines the software requirements for the Online Bidding System by providing a detailed overview of system's intended functionality, behaviour and constraints. The system will be enable users to list items, bid on items, manage auctions and monitor results in a secure and interactive online environment.

Scope

Online Bidding System is a web-based application that engages activities between registered sellers and bidders, with an admin responsible for managing the entire system. It supports real-time bid updates, notifications and items tracking. The system will be scalable and maintainable, using object-oriented design and incorporating several software engineering patterns to achieve modularity and extensibility.

Description, Acronyms and Abbreviations

Admin	The system's administrator who manages users and monitors auctions.
Seller	A registered user who can list items for auctioning
Bidder	A user who can view items and place bid.
MVP	Minimum Viable Product
Item	Product or service listed for bidding

Overall description

The system will function as a stand-alone web application and later integrate with a payment gateway or shipping services. It will include models for user management, item listing, bidding logic and reporting.

Product Functions

1. User registration and login for admin , bidder and sellers
2. Adminn control panel
3. Item listing and status tracking
4. Placing, updating and cancelling
5. Bid historyand winner announcement
6. Notification for bidding events (SMS/email)

User classes and charasteristics

Class	Charasteristics
Admin	has a high level access for managing all users and content.
Seller	Can add, edit and monitor items listings and review bidding activity.
Bidder	Can search for items, place bids, and receive notifications.

Design and Implementation

1. The system must follow object-oriented design principles.
2. The system must also support standard security practices like input validation and password hashing.
3. It should also be extensible platform for future phases.

Specific Requirements

Functional Requirements

1. Users must register and login using secure credentials.
2. Sellers must be able to add, update and remove items from the listing.
3. Bidders must be able to place, modify or cancel bids before the auction ends.
4. System administrator must be able to approve listing, ban users, and view system logs.
5. The system must offer automatic highest bid and close and countdown to the auction.
6. The winner should be announced after the auctions closes.

Non-functional Requirements

1. The system should support at least 1000 concurrent users
2. All data must be securely stored and encrypted where necessary.
3. The system should respond to user auctions within 2 seconds
4. The UI must be responsive and mobile-friendly

Participating Actors:

- The buyer registers, looks through auctions, and submits bids.
- Seller: Signs up, keeps track of bids, and lists products
- Admin: Oversees users, goods, and auctions.
- System: Controls timers, bid logic, and validations

Interface Requirements

User Interface	Admin Dashboard	API
Form-based inputs for items and bid creation	Graphs and tables for system monitoring	RESTful endpoints for frontend-backend communication

Conclusion

The system is designed to provide a seamless and efficient platform that enables sellers to auction their items and bidders to participate competitively in a real-time bidding. By leveraging object-oriented principles and proven software engineering patterns, the system will ensure scalability, maintainability, and user interaction, while also addressing design constraints and non-functional requirements, covering strong functions for the development phase.

5. System Requirements for Online Bidding System

System requirements defines the system level requirements necessary for successful development and functionality of the Online Bidding System. With this system, theregisted users are enabled to participate on online auctions by placing bid or items posting.

Overview

The system is said to be a web-based application that allows three distinct role: Seller, Biiders and Admin. It is anticipated to enable listing of items, real-time bidding, validation, auction timing and winner declaration and provide administrative contrl for managing users and auctions.

Hardware Requirements

Components	Min Requirements	Recommended
Server CPU	Quad-Core 2.4GHz	Octa-Core 3.0 GHz
RAM	4 GB	8GB or more
Storage	50GB	100GB
Client Device	<ul style="list-style-type: none">1GHz Proccesor2 GB RAM	<ul style="list-style-type: none">2 GHz proccesor4 GB RAM
Network	5mbps upload/download speed`	20 mbps or higer

Software Requirements

Components	Requirements
OS	Win Server 2012
Web Server	Apache
Database	MySQL
Runtime Environment	Flask
Frontend Framework	HTML, CSS, JavaScript
Browser Support	Chrome, FireFox, Safari, Edge

6. System Design Document For Online Bidding System

This section provide a comprehensive system level design of the named system Online Bidding System. It outlines how the systems' fucntional requirements will be released to the final product.

Scope

Online Bidding System allows the users to register as buyers or sellers abd participate in auction. Selles can list items and buyers can place a bid. The system manages bid evaluation, notifications abd auction closure in the real-time.

With the system design document, there is architectural design, system components, modules interaction, interface design and data design requirements for the implementation and deployment of the Online Bidding System.

System Context

1. The system operates over the internet.
2. It integrates with user device (mobile/web)
3. Interacts with a secure database backend.
4. Interface with a notifications services (email, SMS)

Architecture

The system follows 3-tier Architecture as follows

1. **Presentation layer :**
 - a. Web-based UI for all users (views)
2. **Application layer:**
 - a. Core business logic using python based service (Control).
3. **Data layer :**
 - a. MySQL database managing all persistent data (model)

In general, out the three layers given, the system employees **Model View Control**.

Design Consideration

Assumption

1. User have access to a stable internet connection.
2. System operates on HTTPs for security.

Constraints

- Real-time bid update latency less than 2 seconds
- System should support up 1000 concurrent users.

System Architecture

Components Diagram

1. User management
2. Auction Engine
3. Item catalog
4. Notification Service
5. Reporting Module

6. UML Diagrams

1) Use Case Diagram



Figure 1

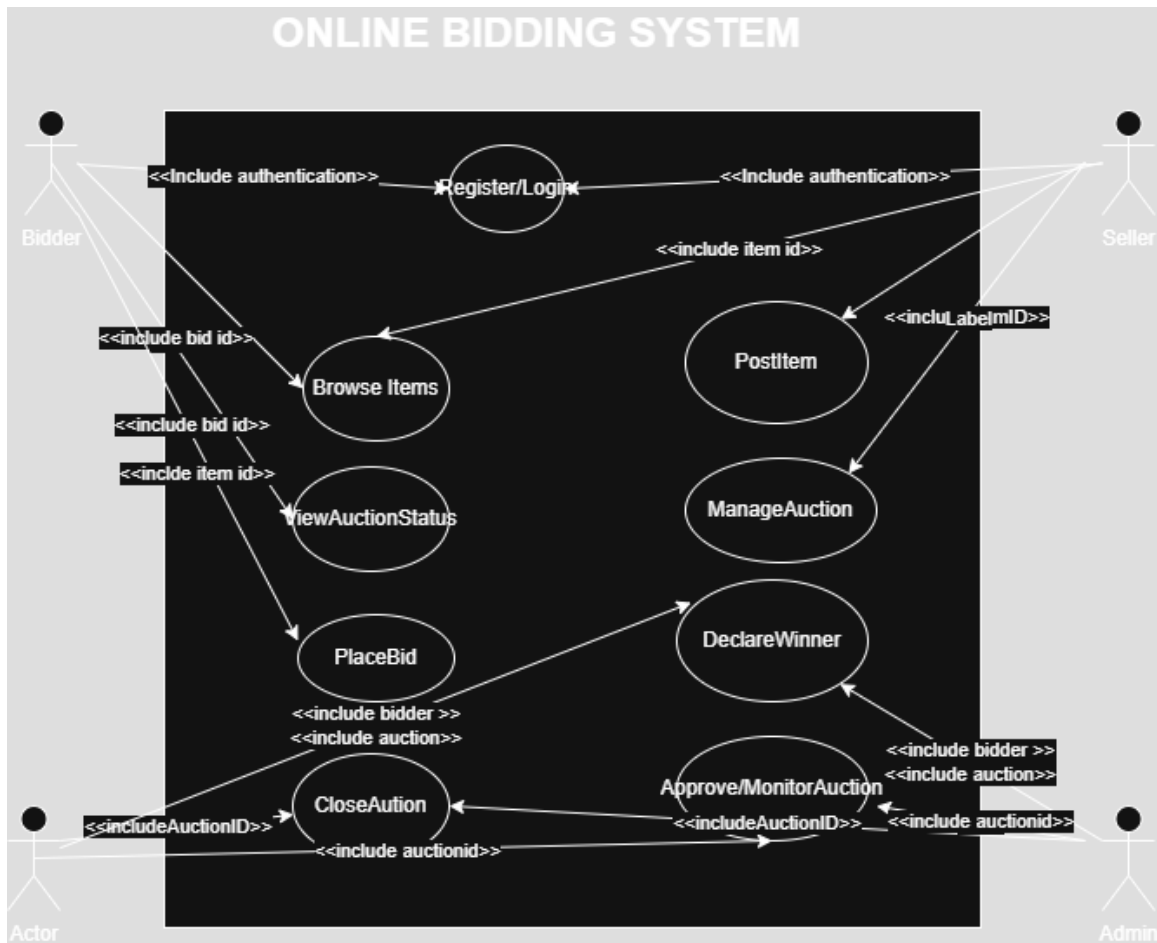


Figure 2

The system's functionality is described in the Use Case Diagram from the viewpoint of its users. Actors like buyers, sellers, and administrators are included, as are their interactions with system functions including product listings, bidding, registration, login, and auction monitoring. This graphic makes user roles and responsibilities clear and aids in visualizing the system's external behavior.

2. Object Diagram

It shows a snapshot of instance (object) and their relationship at a specific moment in time. It is similar to a class diagram but focuses on real objects and their values, not just structure.

It is useful for

1. Debugging
2. Understanding system state at the time
3. Slowing test case scenarios

Scenario Snapshot

Imagine this moment in the system.

1. "A user named "WESI" is bidding on "Iphone 14 Pro" "
2. "The current highest bid is "\$750""
3. "Admin "Bob" is managing items"

Objects and Attributes

1. **User**
 - a. Username = "WESI"
 - b. Email = wezimosiuoa@gmail.com
2. **Admin**
 - a. Username = "TEBA"
 - b. Role = "Admin"
3. **Item**
 - a. Title = "Iphone 14 Pro"
 - b. Starting Price = "\$500"
 - c. Status = "Active"
4. **Bid**
 - a. Amount = "\$750"
 - b. Bidder = "WESI"
 - c. Item = "Iphone 14 Pro"

Instances of the classes specified in the Class Diagram at a specific moment are depicted in the Object Diagram. It illustrates the relationship between items such as a particular buyer, an item up for auction, and the bids that go with it. This makes it easier to see how the system's parts are doing in real time within a particular situation, like an ongoing auction.

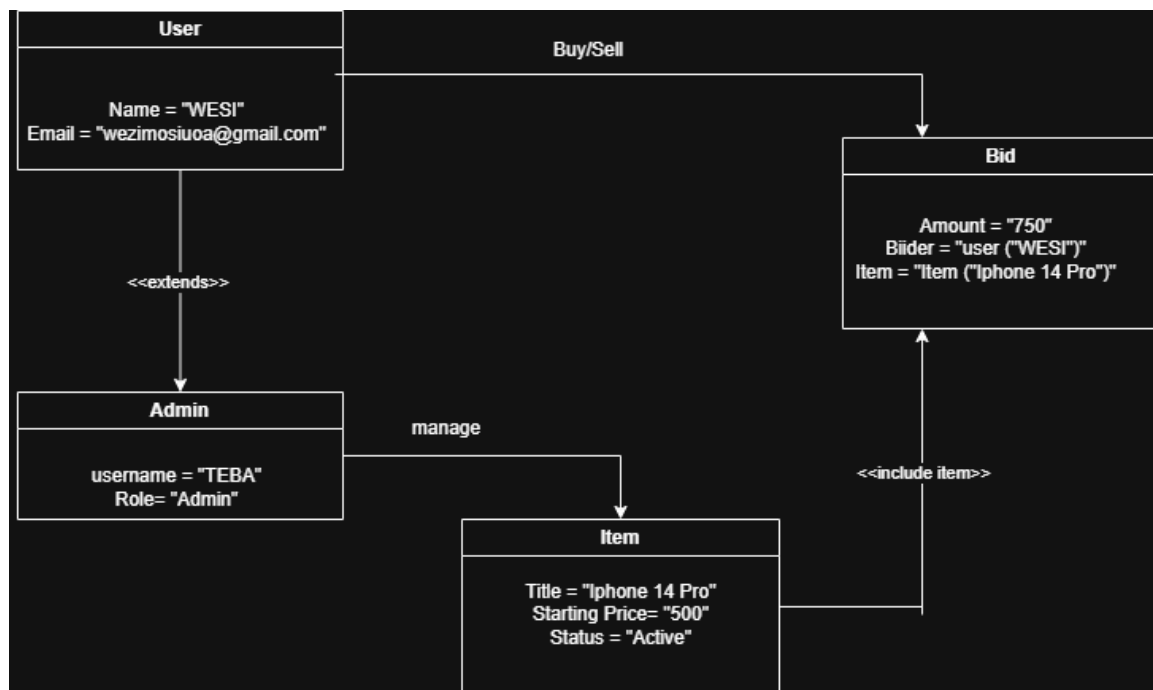


Figure 3

3. Class Diagram

The Online Bidding System's static structure, comprising its classes (User, Product, Bid, Auction, and Admin), is defined by the Class Diagram. In addition to highlighting each class's methods and characteristics, it displays relationships such as inheritance between an administrator and a user, linkages between users and bids, and the aggregation of bids in an auction. This acts as an implementation guide.

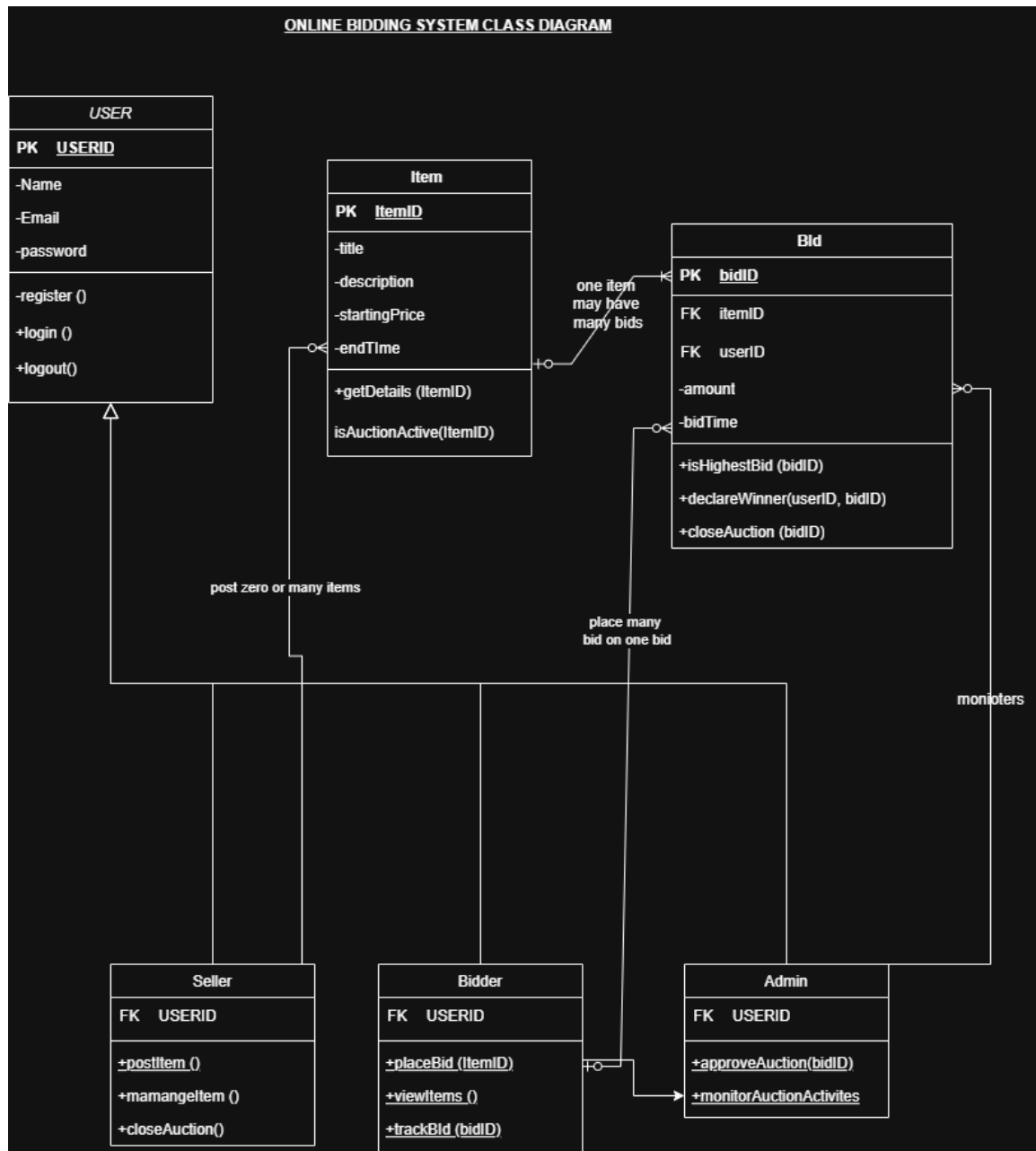


Figure 4

4.1 Sequence Diagram

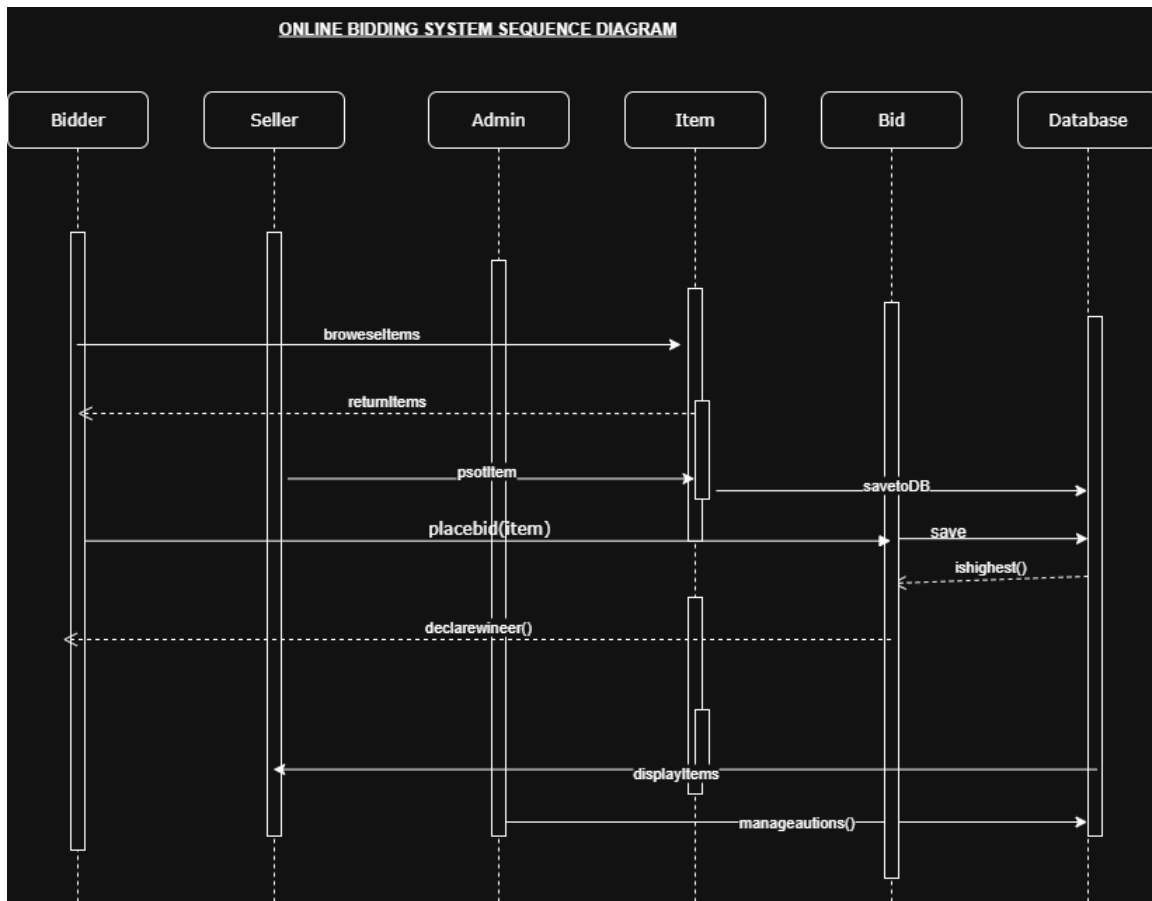


Figure 5

By simulating the interactions between items in a temporal sequence, the sequence diagram shows the system's dynamic behavior. It illustrates, for instance, how a buyer submits a bid: the buyer sends messages to the auction, which updates the bid and notifies the buyer. This is helpful for comprehending how data and control move during operations.

ONLINE BIDDING SYSTEM COLLABORATION DIAGRAM

```

sequenceDiagram
    participant User
    participant Seller
    participant Admin
    participant Bidder
    participant Item
    participant Bid
    participant DB
    participant payment

    User-->>Seller: extends
    Seller-->>Admin: payseller(userid)
    Admin-->>User: extends
    Admin-->>Bidder: 11
    Bidder-->>User: extends()
    Bidder-->>Item: 3 browserItems()
    Item-->>DB: 1 postItem()
    DB-->>Item: 2 saveinDDB
    Bidder-->>Bid: 4 placeBid (items)
    Bid-->>DB: 5 saveBid (bidID)
    DB-->>Admin: 12
    Admin-->>DB: 6 manageBid ()
    Admin-->>DB: 10 getthighestBid (bid)
    Admin-->>payment: 8 retrievePayment()
    payment-->>DB: 9 savePayment()
    Bidder-->>payment: 7 declarWinner(userid)
    payment-->>Item: 1 makePayment ()
    
```

The diagram illustrates the interactions between various components in an online bidding system. The participants are User, Seller, Admin, Bidder, Item, Bid, DB, and payment. The sequence of interactions is as follows:

1. **Item** calls `postItem()` on **DB**.
2. **DB** returns `saveinDDB` to **Item**.
3. **Bidder** calls `browserItems()` on **Item**.
4. **Bidder** calls `placeBid (items)` on **Bid**.
5. **Bid** calls `saveBid (bidID)` on **DB**.
6. **Admin** calls `manageBid ()` on **DB**.
7. **Bidder** calls `declarWinner(userid)` on **payment**.
8. **Admin** calls `retrievePayment()` on **payment**.
9. **payment** calls `savePayment()` on **DB**.
10. **Admin** calls `getthighestBid (bid)` on **DB**.
11. **Admin** calls `payseller(userid)` on **Seller**.
12. **DB** returns data to **Admin**.
- User** extends **Seller**.
- Seller** extends **Admin**.
- Admin** extends **User**.
- User** extends **Bidder**.
- Bidder** extends **payment**.

Although it is optional, the Collaboration Diagram shows how various items cooperate to complete a task. It is less concerned with time sequence and more with object relationships. It illustrates how the Buyer, Auction, Product, and System objects interact and communicate to process a bid in the bidding scenario.

5. Activity Diagram

It represents the main activities and the workflow in Online Bidding System platform. It typically focuses on interactions such as item listing, auction management, and winner declarations.

Activities included

1. User login
2. Role base access:
 - a. Seller list items
 - b. Bidder browsers and places bid
 - c. Auction manager manages auctions
 - d. Admin manages users
3. Bidding process
4. Auction closing
5. Notifications

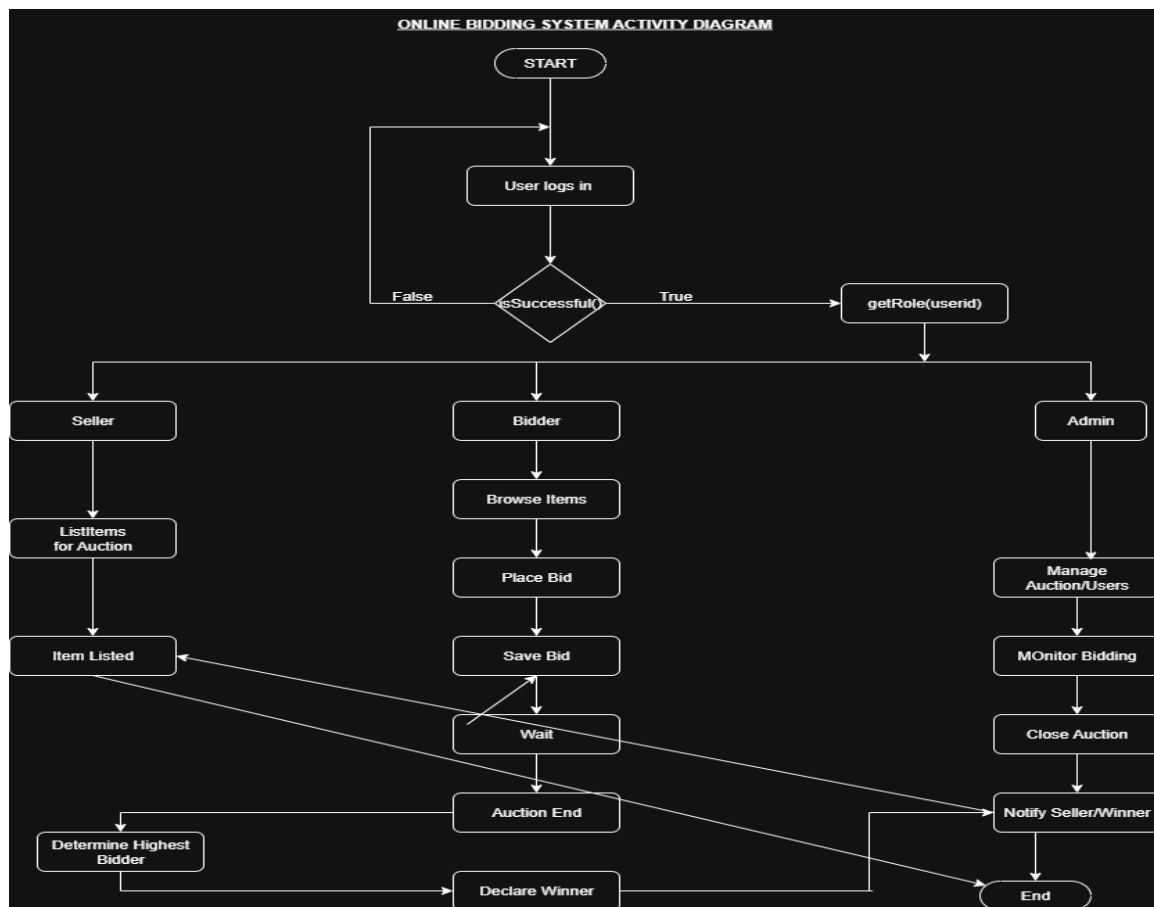


Figure 7

The system's workflow, including the bidding lifecycle for an auction, is modeled by the Activity Diagram. It consists of actions such as listing the product, starting the bidding process, placing several bids, and ending the auction. The system's logical flow is demonstrated by the representation of decisions such as "Is the highest bid?" and "Has the auction expired?"

6. State Chart Diagram

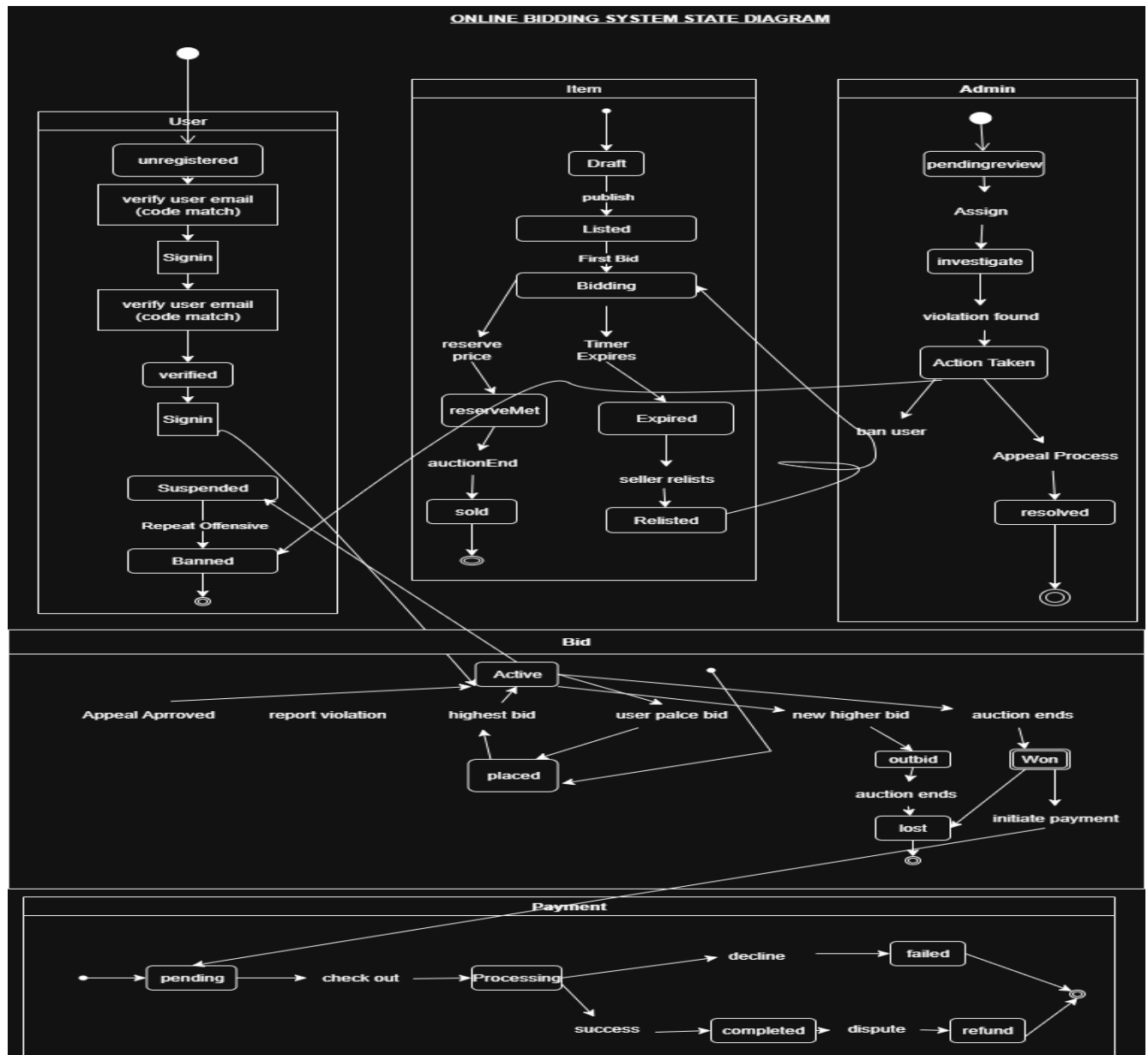


Figure 8

The several states of a crucial object, such as an auction, and the changes between them are depicted in the state chart diagram. For example, depending on time and user activity, an auction may shift from "Scheduled" to "Ongoing" and then to "Ended." Modeling object behavior throughout its lifecycle requires the use of this diagram.

7. Component Diagram

It is a type of UML diagram that focuses on the high-level structure of a system, showing how its components (modules, services, subsystems) are organized and how they interact.

Key concept in the Component System

Elements	Description
Components	A modular: replaceable part of the system (like a package, module, class group or microservice)
Interface	Represents a contract/service provided/required by a component
Dependency	Arrow showing how components rely on each other.
Port	Defines how a component communicates with the outside world.

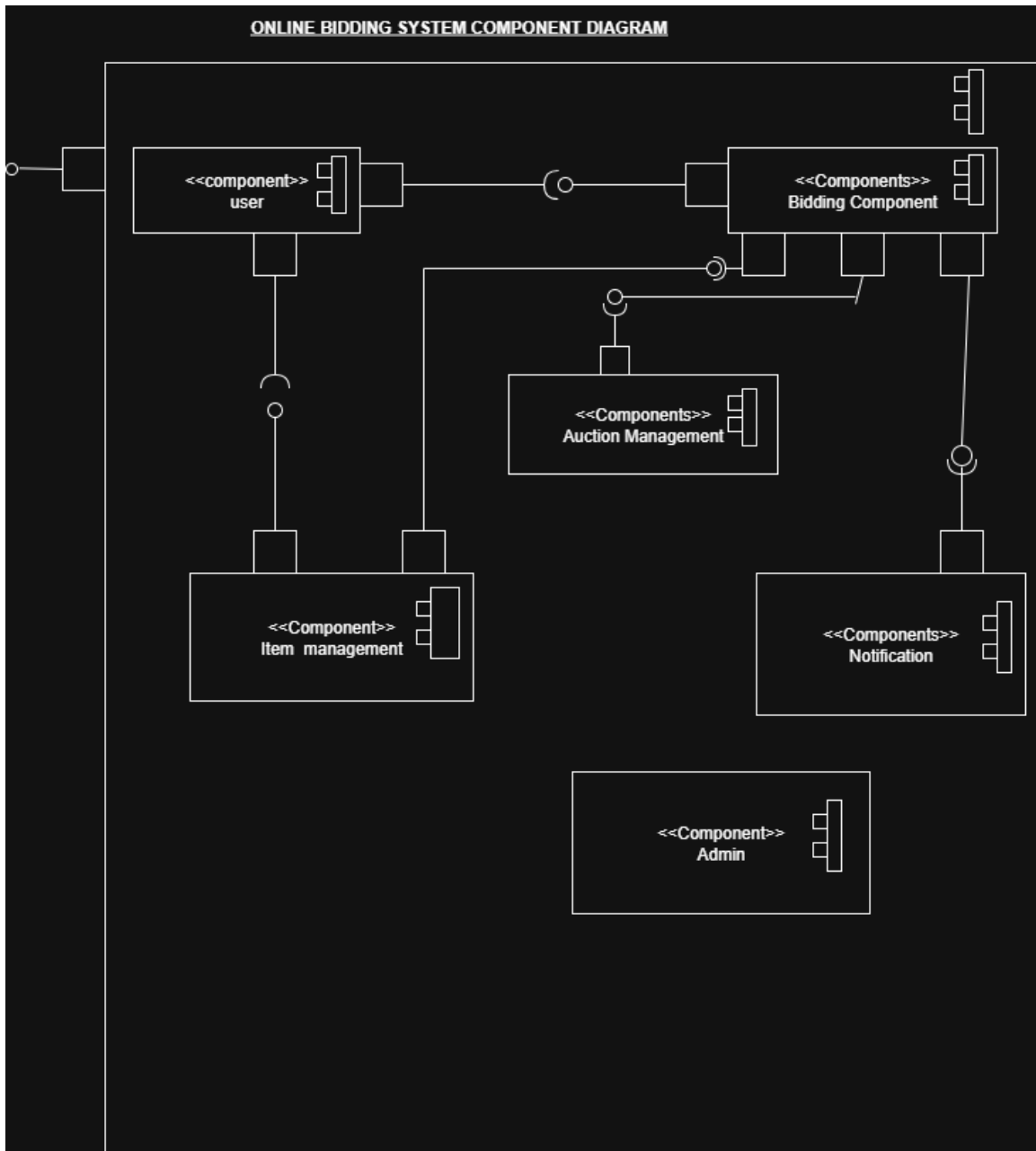


Figure 9

According to the Component Diagram, the software is divided into several parts, such as the User Interface, Auction Management Module, Bid Processor, and Database Connector. During development and implementation, it facilitates a clearer understanding of the software's architecture and the interdependence of its components.

8. Deployment Diagram

It is type of software diagram that shows physical arrangement of hardware (node) and the deployment of the software artifact (like executables, libraries and database).

It is primarily used in system engineering and software engineering to visualize how a system will be deployed in the real-world environment.

1. Nodes
 - a. Represents physical/virtual hardware devices/execution environments.
 - b. Can be
 - i. Devices (e.g Servers, Smartphones, routers)
 - ii. Execution Environments (JVM, Docker Container)
2. Artifacts
 - a. Represents software components/files that are deployed to the nodes (Jar files, .exe files and databases)
3. Communication Paths
 - a. Show how nodes communicate with each other (usually modules/protocols like HTTPs, TCP/IP)
4. Dependencies
 - a. Arrows that show how artifacts depend on each other or on resources.

When to use Deployment Diagram

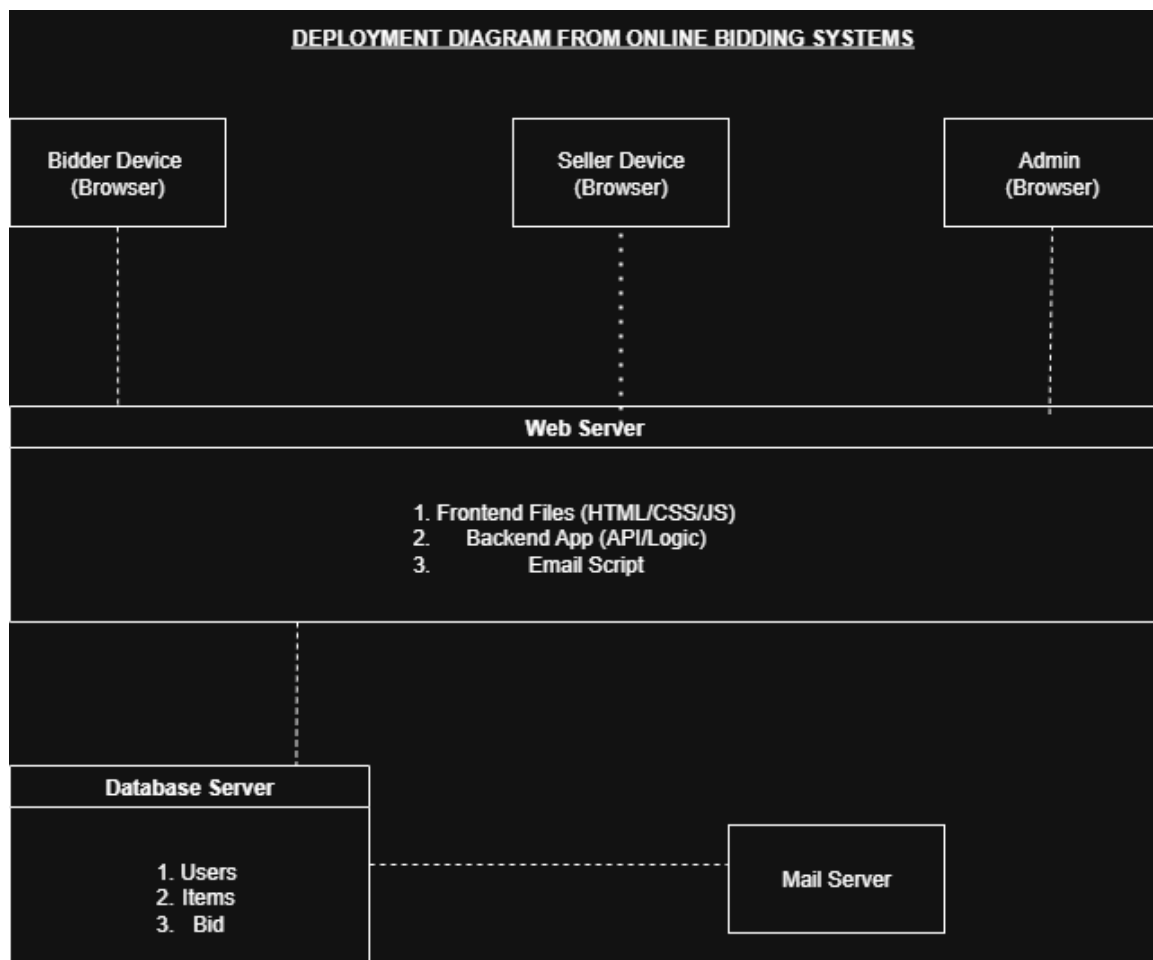
1. To show system topology
2. To model the hardware–software relationship
3. During system design/deployment planning
4. For DevOps Documentation

Key Components

1. Client Devices
2. Web Servers
3. Application Server
4. Database Server
5. Mail Server

Deployment Elements

<u>Nodes</u>	<u>Artifact</u>	<u>Communication Paths</u>
<ol style="list-style-type: none">1. User Device (Browser)2. Admin Devices (Browser)3. Web Server4. Database5. Mail Server	<ol style="list-style-type: none">1. Front-End files2. Backend Application3. Database Schema4. Email Script	<ol style="list-style-type: none">1. HTTPs between devices and web server2. SQL between web server and database.3. SMTP between web server and mail server



The Deployment Diagram shows the physical layout of hardware (nodes) and the software components deployed on each. For this system, the nodes may include a client browser, web server, application server, and a database server. This diagram helps visualize the system's infrastructure and is crucial for planning the deployment environment.

7. Test Specification for Online Bidding System

Test specification defines the test case to verify the functionality and behaviour of the aimed system classes. It aims to ensure the correctness, consistency and integrity performed by Seller, Bidder and Admin, Item and Bid.

Item Testing

7. Seller Class

8. Bidder Class
9. Admin Class
10. Item Class
11. Bid Class

Test Specification

Unit Testing

1. Confirm each class/ methods perform the anticipated output in isolation.

- a. Testing Bid Class

```
def test_bid_amount():
    bid = Bid(100.00, bidder, item)
    assert bid.get_amount() > 0
```

This can be done using **unittest** package in Python

2. Integration Testing

- Test interaction between multiple classes

Case

Testing that a bidder places a bid on an Item and Bid is updated.

py

```
import unittest

class TestBidder(unittest.TestCase):
    def test_bidder_place_bid(self):
        seller = Item("Laptop", 2000.00)
        bidder = Bidder("WESI")
        bid = bidder.place_bid(seller, 2000.00)
        self.assertEqual(300, seller.get_highest_bid().get_amount())

if __name__ == '__main__':
    unittest.main()
```

System Testing

With this we aim to test the entire system behaviour from the beginning to the end.

Case

Seller adds item, bidder places bid, admin closes auction and system announces the winner. This can be executed through flow using real or simulated user.

Acceptance Testing

The mandate is to ensure that the system meets user requirements by leveraging admin login and registration as well as approval.

Approved sellers receive a communication and can list items. This system is often done using a checklist from requirements specifications.

Regression Testing

It is meant to ensure that changes don't affect existing functionality.

Case

After modifying the placeBid methods, rerun all tests to Bidder and items to ensure all they behave correctly like before.

It can be done using tools like

- JUnit + CI tool such as GitHub

Performance Testing

This type of testing is intended to handle expected load. In this case the system is expected to simulate 1000 concurrent bidders placing bids using tools like Apache.

Metrics

- Response time
- CPU usage
- Memory Usage

Security Testing

It is always imperative to make sure that the system is safe and secure from unauthorised access.

Case

1. A bidder access admin?
2. A seller modify another seller's items?

Tools

- OWASP ZAP mainly for penetration testing.
- Role-based access testing in code

Usability Testing

Online Bidding is supposed to be of easy User Interface and User Experience

Case

Bidder placing a bid and creating an item. Feedback is also collected based in the following

- Clarity of buttons
- Navigations
- Error Handling

Compatibility Testing

An online bidding system is expected to ensure that app works across all devices /browsers but specifically on web browser because this is web-based application.

Smoke Testing

In Online Bidding System, we perform the quick tests verifying basic functionality after development.

Case

- nLogin
- Adding Item
- Placing Bid

Exploring Testing

Online Bidding System has gone under testing without formal testing cases to discover unexpected bugs.

Case

- Place a negative bid
- Place a bid after auction ends
- Remove item while bidding is active

12.Manual /Guide for Online Bidding System

1. User Manual

Target Audience

- End User
- Bidder and Seller

2. Programmer Guide

Developers

- Developer maintaining or extending the system

Contents

- System Architecture Overview
- Description of core classes (Bid, Bidder, Seller, Items, Admin)
- Database schema and sample quizzes
- REST API or method level document (if applicable)
- Coding conversations and design patterns used.
- Instructions on running unit/integration tests.
- Deployment notes

3. Administrator Guide (Installation Instruction)

Contents

- System requirements
 - o Client
 - OS : Win/MacOS
 - RAM : 2GB /more
 - Processor : Intel
 - o Server
 - OS : Win Server
 - RAM : 8GB
 - Processor : i7 or more
- Installation and configuration
- DB Setup
- Roles (Admin, Seller, Bidder)

13. Project Plan and Schedule

1. Project Schedule and Milestone

Milestone	Start date	End date	Deliverables
Requirements Gathering	1 Jul 2025	3 Jul 2025	Requirements Specification Document
System Design (UML, Classes Desing)	4 Jul 2025	6 Jul 2025	UML diagram, Class Diagram
Back-end Development (Core Classes)	7 Jul 2025	11 Jul 2025	Seller, Bidder, Items, Bid, Admin
Front-end UI	12 Jul 2025	13 Jul 2025	Basic UI
Unit and integration Testing	14 Jul 2025	18 Jul 2025	Test report
Document and manual;	19 Jul 2025	20 Jul 2025	User, Admin, performance manual
Feedback and submission	21 Jul 2025	21 Jul 2025	Final Submission

2. Resource allocation

Resources	Qauntity	Usage
Developer (students)	4	Coding/testing`
Laptops	4	Development and Testing
IDE (VS Code)	1 per developer	Coding and Debugging
Github	1 repo	Version Control
Internet		Research and Collaboration s

3. Resource allocation

Item	Cost (Rupees)
Development Tools	0 (Open Source)
Internet and Electricity	4000.00 shared
Printing	850.00

14. Quality Plan : Setting Quality Assurance Procedures

This part outlines the processes, standards, and activities required to ensure that a Online Bidding System meets its intended quality objectives. It completely defines how quality will be managed, monitored and measured throughout the lifecycle of the project.

Purpose

- To provide a structure to maintaining quality.
- To ensure deliverables meet stakeholder expectations and comply with relevant standards. To identify responsibilities for quality assurance (QA) and quality control (QC) activities.

Components

Component	Definition
Quality Objectives	Specific, measurable goal related to quality
Standards and Guidelines	Industry standards, company policies or regulatory requirements to follow
Quality Control Measures	Methods for detecting and correcting defects
Roles and Responsibilities	Who will perform QA/QC activities
Documentation Requirements	Reports, checklists and records to maintain evidence of quality.
Review and Audit Plan	How and when quality will be reviewed or audited

Quality plan for online bidding system outlines the quality assurance procedures which includes the core components which Bidder, Seller, Admin, Item, and Bid to ensure that the system is reliable, secure, and performs as expected for all users.

Quality Objective

- Ensure all bidding transactions are accurate and consistent.
- Prevent fraudulent bids for items or unauthorised access.
- Maintain data integrity for items, bids and user accounts
- Ensure the system is user-friendly and responsive.
- Comply with best practices in software development and testing.

Standards and Guidelines

- Follow PEP8 (Python) coding standards for clarity and maintainability
- Implement OWASP guidelines to prevent common vulnerabilities like SQL injection, XSS, CSRF.
- System will handle at least 1000 concurrent users without significant lag.

Quality Assurance Procedures

1. Code Review
 - a. All code involving Bidder, Seller, Admin, Item, and Bid classes must undergo peer review before merging.
2. Unit Testing
 - a. Each class method like placeBid in Bidder will have automated unit tests.
3. Integration Testing
 - a. Validate the flow between classes, like Bidder placing a bid updates the Item's highest bid correctly.
4. System Testing
 - a. End-to-end scenarios tested, including item listing, bidding and admin approval.
5. Performance Testing
 - a. Load testing to ensure response times remain optimal under heavy usage.

Quality Control Measures

1. Use ticketing system like Jira to record, prioritize and track issues.
2. Measure number of bugs found per module.
3. Seller and Bidder workflows were tested by sample group before deployment.

Roles and Responsibilities

1. Quality Assurance Engineers execute test cases and report defects
2. Developers fix defects and follow coding standards
3. Admin classes implement accordance of bids and auctions correctly.
4. Project managers will oversee quality assurance activities.

Documentation Requirements

1. Test case documents for each feature.
2. QA reports for every sprint.
3. User manuals for sellers, bidders, and admins.

Review and Audit Plan

1. **Weekly QA Meetings** to review progress and defects.
2. **Monthly Audits** to ensure compliance with the plan.

15.Conclusion

A cutting-edge digital framework for holding online auctions is offered by the Online Bidding System. A thorough UML-based design process has been used to examine and document the system from a number of angles. Understanding the system's requirements, structure, behavior, and deployment is aided by each UML diagram. The project provides a comprehensive understanding of how software systems can be efficiently represented and conveyed using UML, and it is scalable and safe.