

Darshan University

A Project Report on

**“Auction Management System”**

Under the subject

**Software Engineering (2301CS405)**

B. Tech, Semester – IV

Computer Science & Engineering Department

|  |  |
| --- | --- |
| Submitted By | |
| Student Name: Dhruv R. Pithwa | Enrolment No.: 24010101675 |
| Academic Year  (2024-2025) | |
| Internal Guide  Prof. R. B. Gondaliya  Darshan University | Dean-DIET  Dr. Gopi Sanghani  Darshan University |

|  |  |
| --- | --- |
|  | **Computer Science & Engineering Department**  **Darshan University** |

**DECLARATION**

We hereby declare that the SRS, submitted along with the **Software Engineering** **(2301CS405)** for entitled **“Auction Management System”** submitted in partial fulfilment for the Semester-5 of **Bachelor Technology (B. Tech)** in **Computer Science and Engineering (CSE)** Departmentto Darshan University, Rajkot, is a record of the work carried out at **Darshan University, Rajkot** under the supervision of R. B. Gondaliya and that no part of any of report has been directly copied from any students’ reports, without providing due reference.

Dhruv R. Pithwa

Student’s Signature

Date: \_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
|  | **Computer Science & Engineering Department**  **Darshan University** |

**CERTIFICATE**

This is to certify that the SRS on **“Auction Management System” has** been satisfactorily prepared by **Dhruv R. Pithwa** (**24010101675**) under my guidance in the fulfillment of the course **Software Engineering (2301CS405)** work during the academic year 2024-2025.

|  |  |  |
| --- | --- | --- |
| Internal Guide  Prof. R. B. Gondaliya  Darshan University |  | Dean-DIET  Dr. Gopi Sanghani  Darshan University |

**Acknowledgement**

I wish to express my sincere gratitude to my project guide Prof. R. B. Gondaliyaand all the faculty members for helping me through my project by giving me the necessary suggestions and advices along with their valuable co- ordination in completing this work.

I also thank my parents, friends and all the members of the family for their precious support and encouragement which they had provided in completion of my work. In addition to that, I would also like to mention the Darshan University personals who gave me the permission to use and experience the valuable resources required for the project from the University premises.

Thus, in conclusion to the above said, I once again thank the faculties and members of **Darshan University** for their valuable support in completion of the project.

Thanking You

**Dhruv R. Pithwa**

**ABSTRACT**

This Software Requirements Specification outlines the development of an internet-based Auction Management System designed to streamline online auctions. The system supports multiple user roles—Bidders, Sellers, Guests, Auction Managers, Customer Support, and Drivers—offering functionalities such as auction creation, bidding, real-time updates, dispute resolution, and delivery tracking. Key features include secure account management with two-factor authentication, wallet integration, and intuitive auction browsing with filters. The system ensures usability, accuracy, 24/7 availability, and maintainability, catering to an efficient and transparent online auction marketplace.

**Table of Contents**

[List of Figures I](#_Toc146706805)

[List of Tables II](#_Toc146706806)

[1 Introduction 1](#_Toc146706807)

[1.1 Product perspective 1](#_Toc146706808)

[1.2 Product features 1](#_Toc146706809)

[1.2.1 There are three different users who will be using this product: 1](#_Toc146706810)

[1.2.2 The features that are required for the Librarian are: 1](#_Toc146706811)

[1.2.3 The features that are required for the Member are: 1](#_Toc146706812)

[1.3 Functional Requirement 1](#_Toc146706813)

[1.3.1 Librarian 1](#_Toc146706814)

[1.3.2 Member 2](#_Toc146706815)

[1.4 Non-Functional Requirement 2](#_Toc146706816)

[1.4.1 Usability: 2](#_Toc146706817)

[1.4.2 Accuracy: 2](#_Toc146706818)

[1.4.3 Availability: 2](#_Toc146706819)

[1.4.4 Maintainability: 2](#_Toc146706820)

[2 Design and Implementation Constraints 3](#_Toc146706821)

[2.1 Use case diagram 3](#_Toc146706822)

[2.2 Activity diagram and Swimlane diagram 4](#_Toc146706823)

[2.3 Sequence diagram 5](#_Toc146706824)

[2.4 State diagram 6](#_Toc146706825)

[2.5 Class diagram 7](#_Toc146706826)

[2.6 Data flow diagram 8](#_Toc146706827)

[2.6.1 Context diagram (level-0) 8](#_Toc146706828)

[2.6.2 DFD Level-1 8](#_Toc146706829)

[2.6.3 DFD Level-2 9](#_Toc146706830)

[3 External interface requirement (Screens) 10](#_Toc146706831)

[3.1 Screen-1: Registration Form 10](#_Toc146706832)

[3.2 Screen-2: Login Form 11](#_Toc146706833)

[3.3 Screen-3: Add borrower book 12](#_Toc146706834)

[4 Database design 13](#_Toc146706835)

[4.1 List of Tables 13](#_Toc146706836)

[5 Stories and Scenario 14](#_Toc146706837)

[5.1 Story-1: Add New Book in Library Catalogue 14](#_Toc146706838)

[5.1.1 Scenario# S1.1 14](#_Toc146706839)

[5.1.2 Scenario# S1.2 14](#_Toc146706840)

[5.1.3 Scenario# S1.3 14](#_Toc146706841)

[5.2 Story-2: Search Book 15](#_Toc146706842)

[5.3 Story-3: Manage due date for borrowed book 15](#_Toc146706843)

[5.4 Story-4: Renew book 15](#_Toc146706844)

[5.5 Story-5: Generate a report on book usage and availability 15](#_Toc146706845)

[6 Test cases 16](#_Toc146706846)

[7 References 20](#_Toc146706847)

# List of Figures

[Figure 2.1‑1 Use case diagram for library management system 3](#_Toc146706848)

[Figure 2.2‑1 Activity diagram for Book Issue 4](#_Toc146706849)

[Figure 2.2‑2 Swimlane diagram for Book Issue 4](#_Toc146706850)

[Figure 2.3‑1 Sequence diagram for Book Issue 5](#_Toc146706851)

[Figure 2.4‑1 State diagram of Book 6](#_Toc146706852)

[Figure 2.4‑2 State diagram for Librarian 6](#_Toc146706853)

[Figure 2.5‑1 Class diagram for Library management system 7](#_Toc146706854)

[Figure 2.6‑1 Context diagram for Library management system 8](#_Toc146706855)

[Figure 2.6‑2 DFD level-1 for Library management system 8](#_Toc146706856)

[Figure 2.6‑3 DFD level-2 for Issue book 9](#_Toc146706857)

[Figure 3.1‑1 Screen-1: Registration Form 10](#_Toc146706858)

[Figure 3.2‑1 Screen-2: Login Form 11](#_Toc146706859)

[Figure 3.3‑1 Screen-3: Add borrower book 12](#_Toc146706860)

# List of Tables

[Table 3.1‑1 Screen element of Registration form 10](#_Toc146706861)

[Table 3.2‑1 Screen element of Login form 11](#_Toc146706862)

[Table 3.3‑1 Screen element of Add borrower book 12](#_Toc146706863)

[Table 4.1‑1 Table: Book 13](#_Toc146706864)

[Table 4.1‑2 Table: Borrower 13](#_Toc146706865)

[Table 4.1‑3 Table: Staff 13](#_Toc146706866)

[Table 4.1‑4 Table: Student 13](#_Toc146706867)

# Introduction

## Product perspective

This project aims to develop an internet-based auction management system that replaces traditional manual auction processes with a digital platform. The system enables users to participate in auctions remotely, manage their accounts, and track transactions and deliveries seamlessly. It is a multi-user application designed to support various roles, including Buyers (Bidders), Sellers, Guests, Auction Managers, Customer Support, and Drivers. The system handles core auction-related functions such as auction creation, bidding, approval, dispute resolution, and delivery tracking. It is intended to cater to the needs of an online auction marketplace, ensuring efficiency, transparency, and user satisfaction.

## Product features

### User Roles

### The system supports six distinct user types, each with specific functionalities:

* **Bidder:** Buyers who participate in auctions by placing bids.
* **Member:** who will be accessing the library.
* **Guest:** who will request for membership.
* **Auction Manager:** Administrators who oversee auction operations.
* **Customer Support:** Staff who assist users with issues.
* **Driver:** Personnel responsible for delivering auctioned items.

### Features for bidder

* Browse ongoing auctions with filters (e.g., category, price, rating).
* View auction listings, including item photos, names, descriptions, and current bid prices.
* Place bids higher than the current bid amount.
* Increase bid amounts on active auctions.
* Retrieve (cancel) a bid within 30 seconds of placing it.
* Receive real-time notifications about auction updates and outcomes.
* Bid on multiple auctions simultaneously.
* Manage a wallet for deposits, withdrawals, and transaction history.
* Track delivery status of won items.

### Features for Seller

* Create auction listings with titles, descriptions, and item details.
* Upload photos of items for auction.
* Set initial bid prices for items.
* Define auction duration using a start and end timestamp.
* Publish auctions for bidders to participate.
* Manage a wallet for tracking earnings and transactions.

### Shared Features for Members (Bidders and Sellers)

* Sign up with email or phone, verified via email/SMS.
* Log in with validated credentials; reset passwords via email/SMS.
* Enable two-factor authentication (2FA) for enhanced security.
* Set up profiles with name, profile picture, interests, and role (buyer/seller).

### Features for Guest

* Browse ongoing and past auctions without logging in.
* Search auctions using filters (e.g., category, price, rating).
* Receive prompts to sign up for full access (bidding or selling).

### Features for Auction Manager

* Review and approve seller auction submissions.
* Monitor auctions and pause or stop them if necessary.
* Access a dashboard with auction activity and trends.
* Manage user accounts (e.g., warn or block buyers/sellers).
* Resolve disputes between buyers and sellers.

### Features for Customer Support

* Respond to user help requests and troubleshoot issues.
* Provide real-time assistance via live chat.
* Track and resolve support tickets.
* Maintain a FAQ section with common solutions.

### Features for Driver

* View delivery details (e.g., item, destination).
* Access optimized delivery routes.
* Update delivery status (e.g., picked up, in transit, delivered).
* Contact buyers or sellers directly if needed.

## Functional Requirement

### Bidder

* **Place Bid:** Submit a bid amount exceeding the current highest bid.
* **Increase Bid:** Adjust an existing bid to a higher amount.
* **Retrieve Bid:** Cancel a bid within 30 seconds of placement.
* **Auction Updates:** Receive real-time notifications about bid status and auction results.
* **Multi-Bidding:** Place bids on multiple auctions concurrently.
* **Delivery Tracking:** Monitor the status of item deliveries post-auction.

### Seller

* **Set Auction Details:** Input auction title, description, and item specifics.
* **Upload Item Photo:** Add an image to the auction listing.
* **Set Item Price:** Define the starting bid amount.
* **Set Time:** Specify auction duration with start and end timestamps.
* **Publish Auction:** Launch the auction for bidding.

### Member (Bidder and Seller Shared Functions)

* **Account Creation:** Register using email or phone, confirmed via email/SMS.
* **Login:** Authenticate with credentials, reset password via email/SMS.
* **Two-Factor Authentication:** Activate 2FA for secure access.
* **Profile Setup:** Configure profile with name, photo, interests, and role.
* **Wallet Management:** Link a bank account, create a wallet, deposit/withdraw funds, and view transaction history.
* **Auction Browsing:** Explore auctions with filters (category, price, rating).
* **Auction Listing:** View detailed auction information (photo, name, description, bid price).

### Guest

* **Browse Auctions:** View ongoing and past auctions without authentication.
* **Search & Filter:** Search auctions by category, price, or rating.
* **Sign Up Prompt:** Display registration prompts to encourage account creation.

### Auction Manager

* **Approve Auctions:** Validate and approve seller-submitted auctions.
* **Monitor Auctions:** Pause or terminate auctions if issues arise.
* **Track Data:** Access auction statistics and trends via a dashboard.
* **Manage Users:** Issue warnings or block users for policy violations.
* **Handle Disputes:** Mediate and resolve conflicts between bidders and sellers.

### Customer Support

* **Help Requests:** Address user inquiries and technical issues.
* **Chat Support:** Provide live chat assistance.
* **Track Issues:** Manage support tickets through resolution.
* **FAQs:** Develop and update a knowledge base of common questions.

### Driver

* **Delivery Details:** Retrieve details of items and delivery locations.
* **Routes Made Easy:** Access optimized delivery routes.
* **Update Status:** Log delivery progress (picked up, in transit, delivered).
* **Contact Info:** Obtain contact details for buyers/sellers.

## Non-Functional Requirement

### Usability:

* The user interface must be intuitive and user-friendly, requiring no specialized training.
* Support for multiple languages can be implemented based on user demographics.

### Accuracy:

* Auction data (e.g., bid amounts, transaction records) and delivery statuses must be accurate, consistent, and reliable.
* Real-time updates must reflect the correct auction state without delays.

### Availability:

* The system should be available 24/7 to support online auctions across time zones.
* Downtime recovery should occur within 30 minutes or less.
* Responses to user actions (e.g., bid placement, page loading) must occur within two seconds.

### Maintainability:

* The software must be easily maintainable, allowing for quick updates or feature additions.
* It should be portable, adaptable to various devices (e.g., web, mobile) and platforms with minimal reconfiguration.

# Design and Implementation Constraints

## Use case diagram

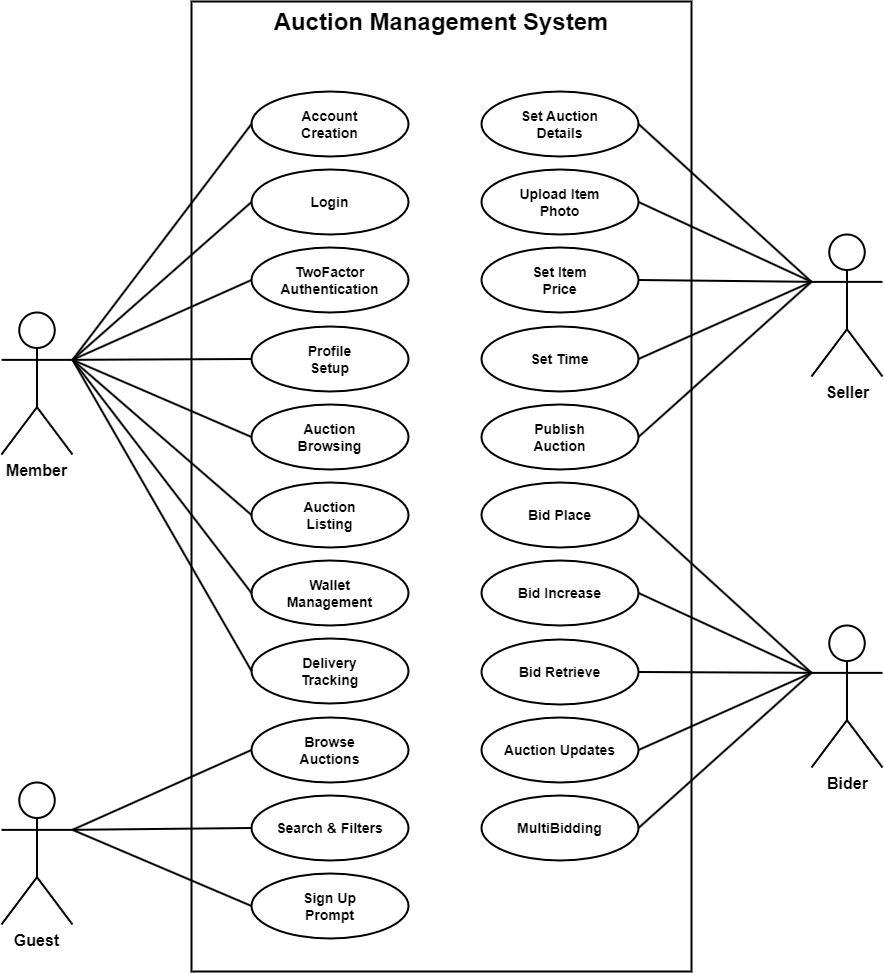


Figure 2.1-1 Use case diagram for Auction Management System

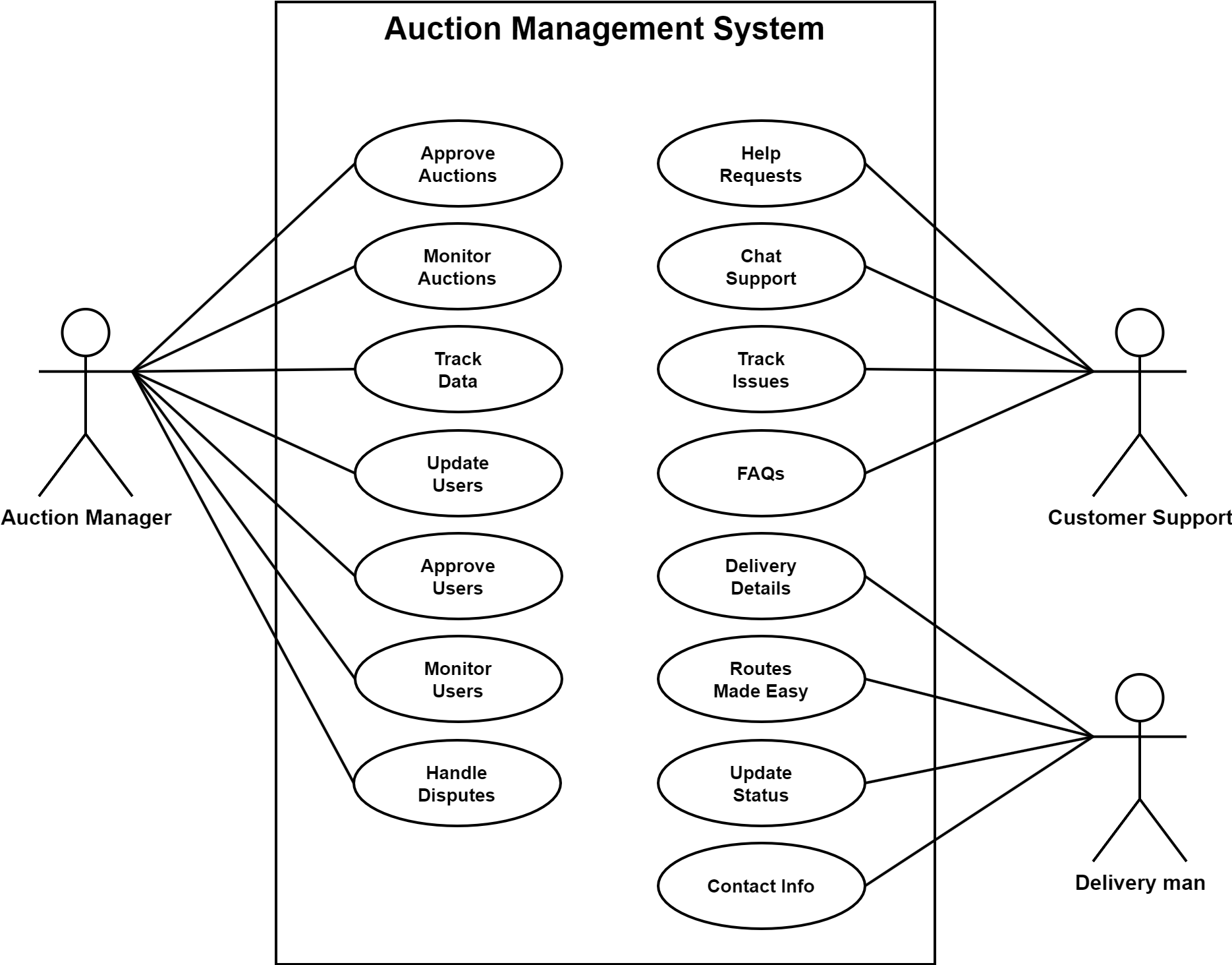


Figure 2.1-2 Use case diagram for Auction Management System

## Activity diagram and Swimlane diagram

### Activity diagram and Swimlane diagram for Creating Auction

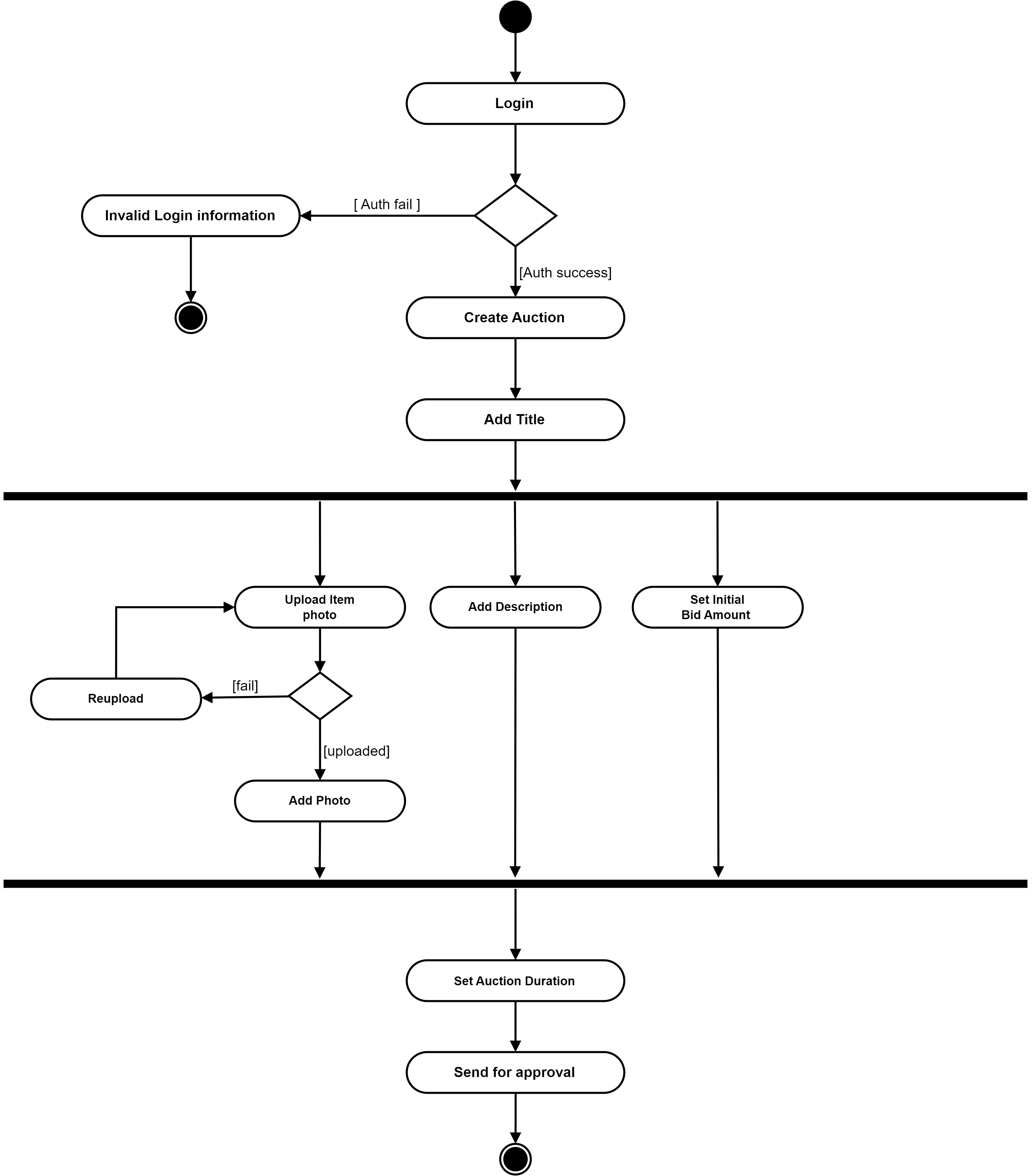


Figure 2.2.1-1 Activity diagram for Creating Auction

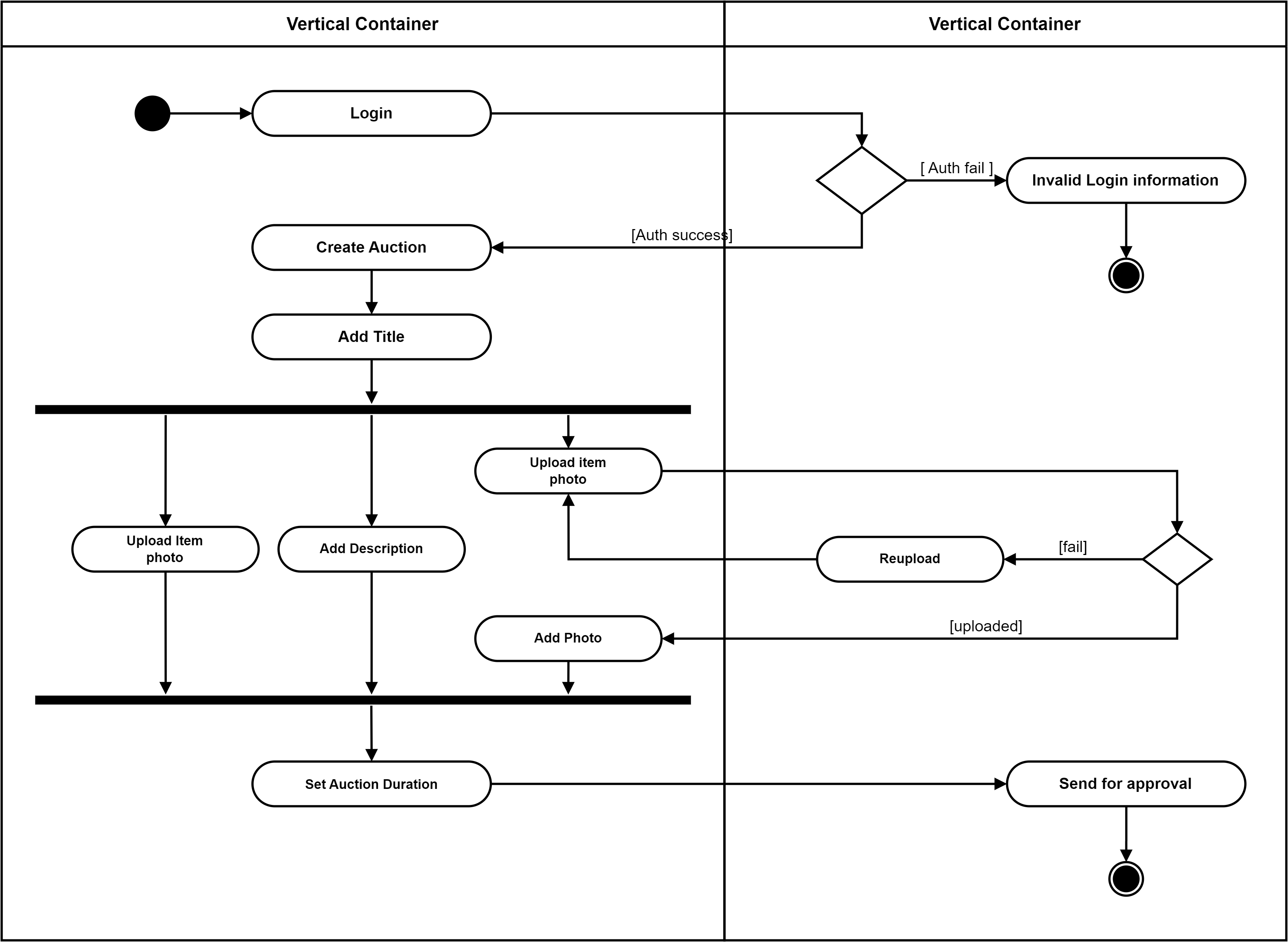


Figure 2.2.1-2 Swimlane diagram for Creating Auction

### Activity diagram for Placing a Bid

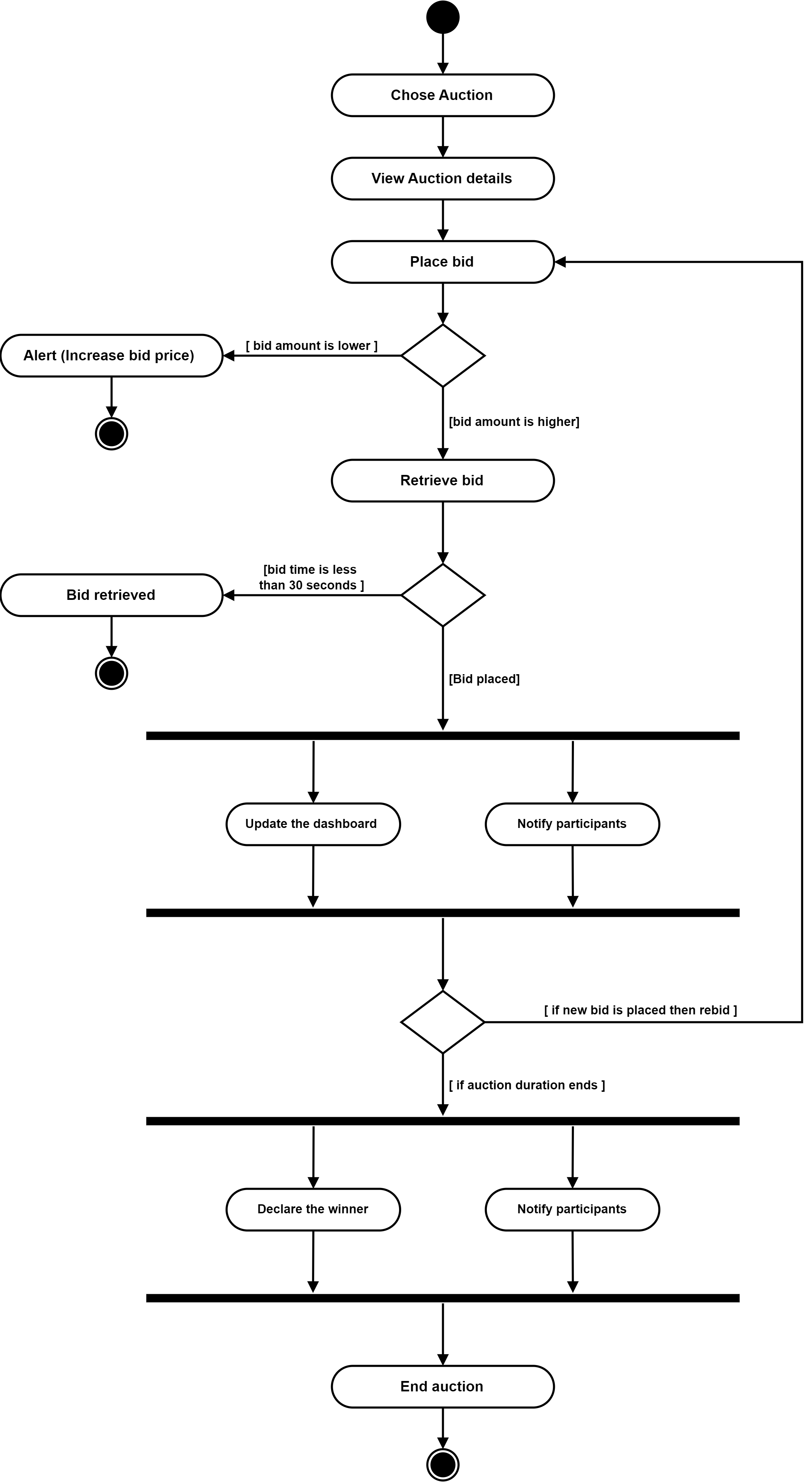


Figure 2.2.2 Activity diagram for Placing a Bid

## Sequence diagram

### Sequence diagram for Creating Auction



Figure 2.3.1 Sequence diagram for Creating Auction

### Sequence diagram for Searching Auctions and Placing a Bid

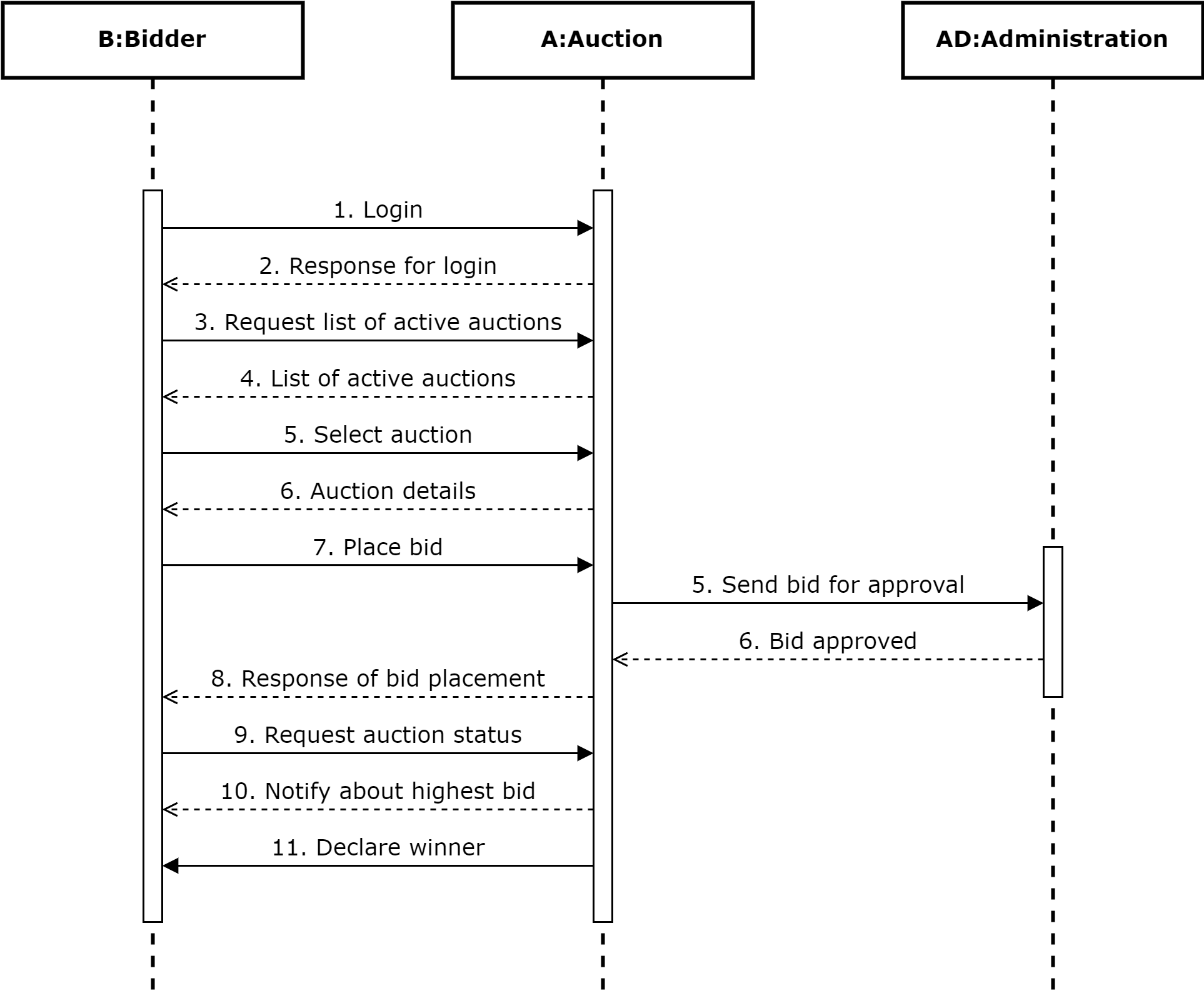


Figure 2.3.2 Searching Auctions and Placing a Bid

## State diagram

### State diagram of Auction

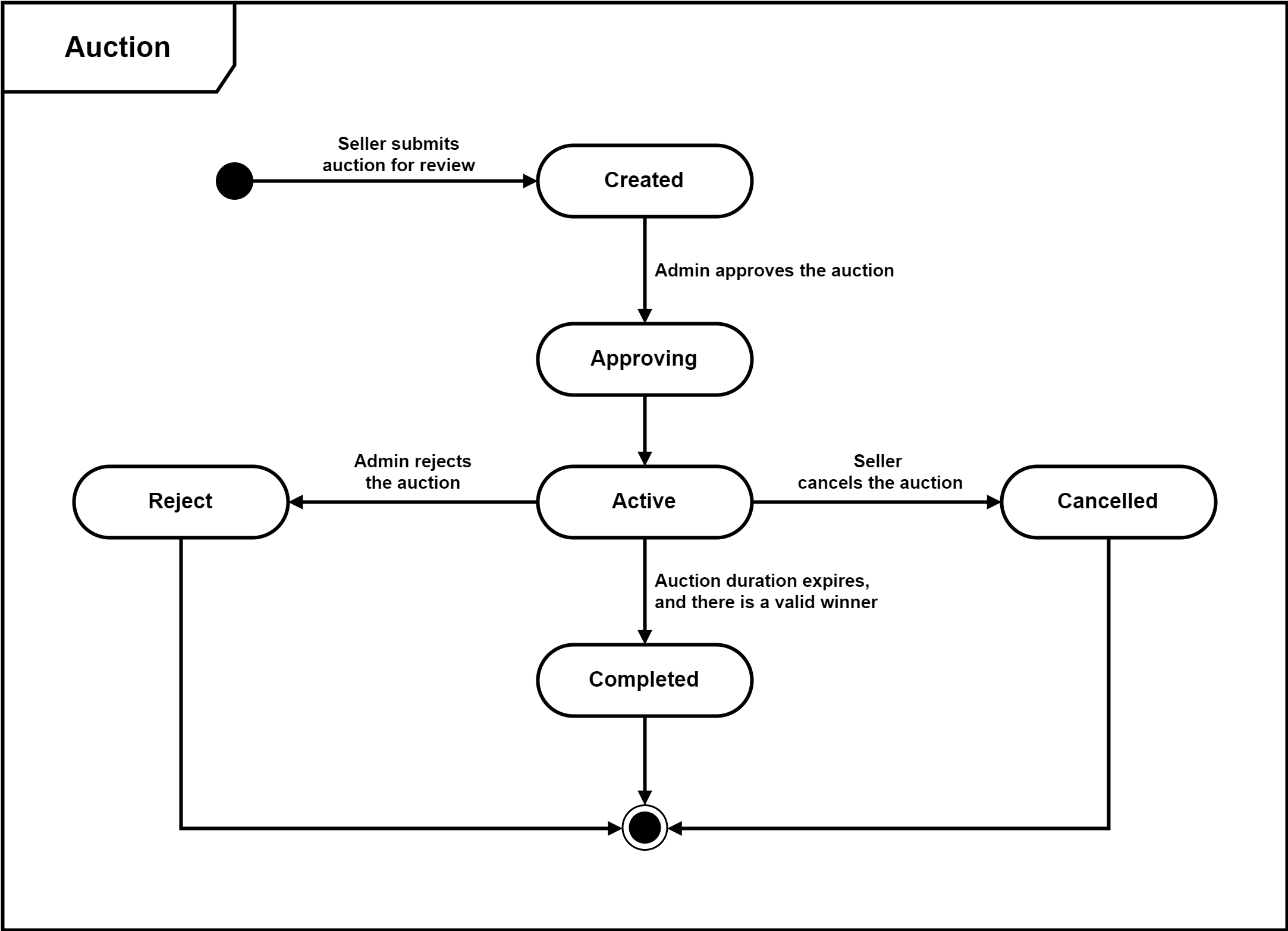


Figure 2.4.1 State diagram of Auction

### State diagram of Bid

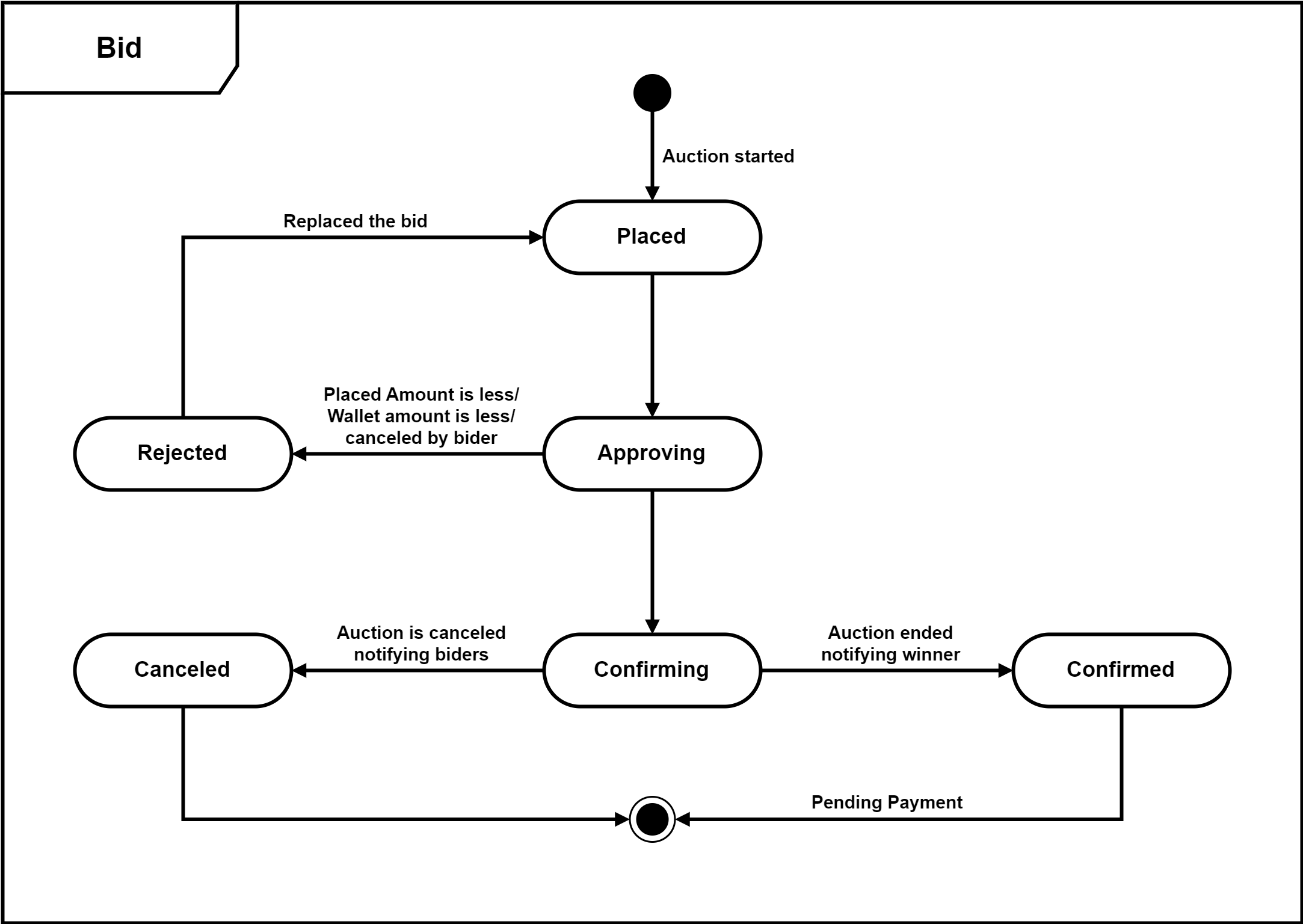


Figure 2.4.2 State diagram of Bid

## Class diagram

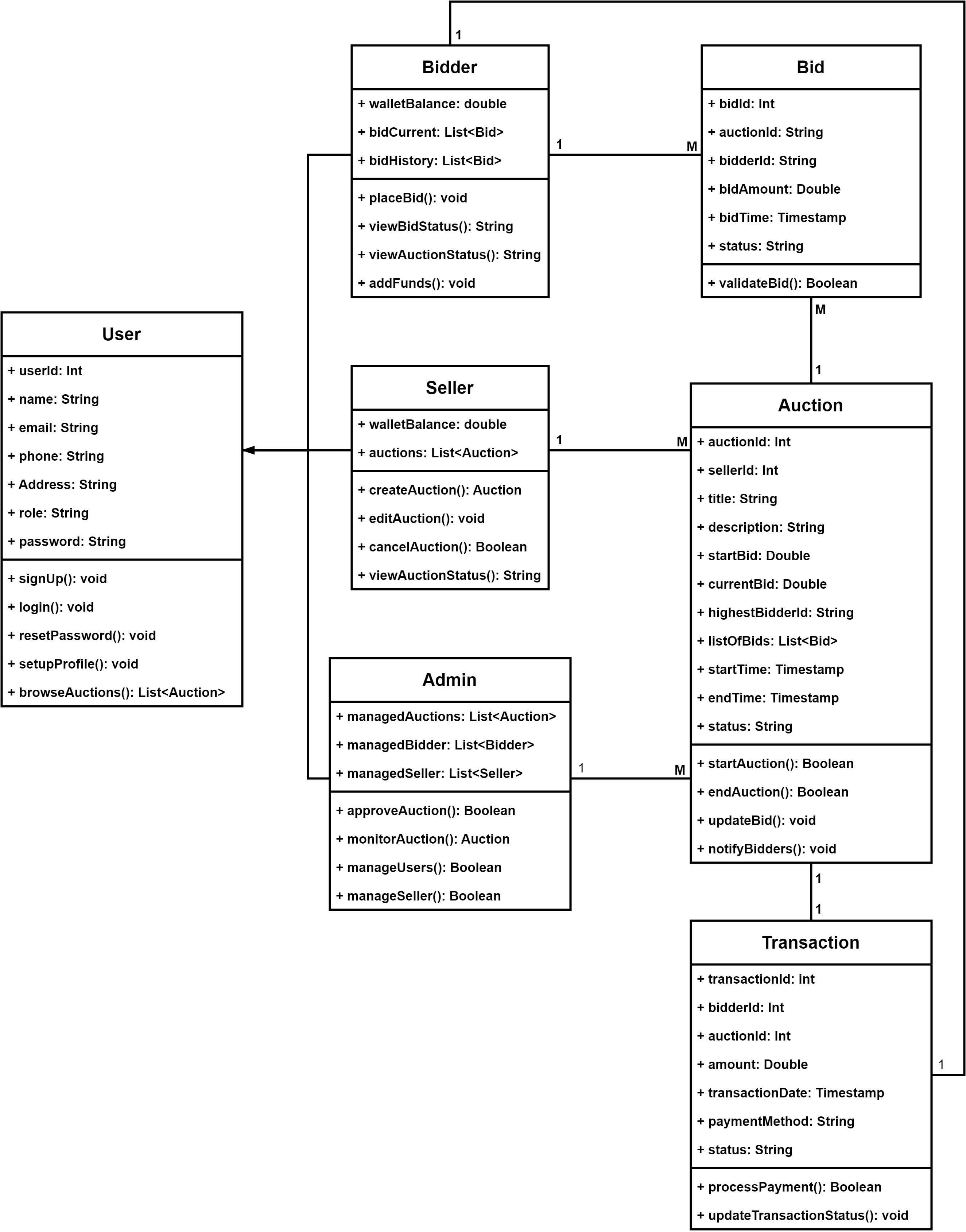


Figure 2.5 Class diagram for Auction Management System

## Data flow diagram

### Context diagram (level-0)

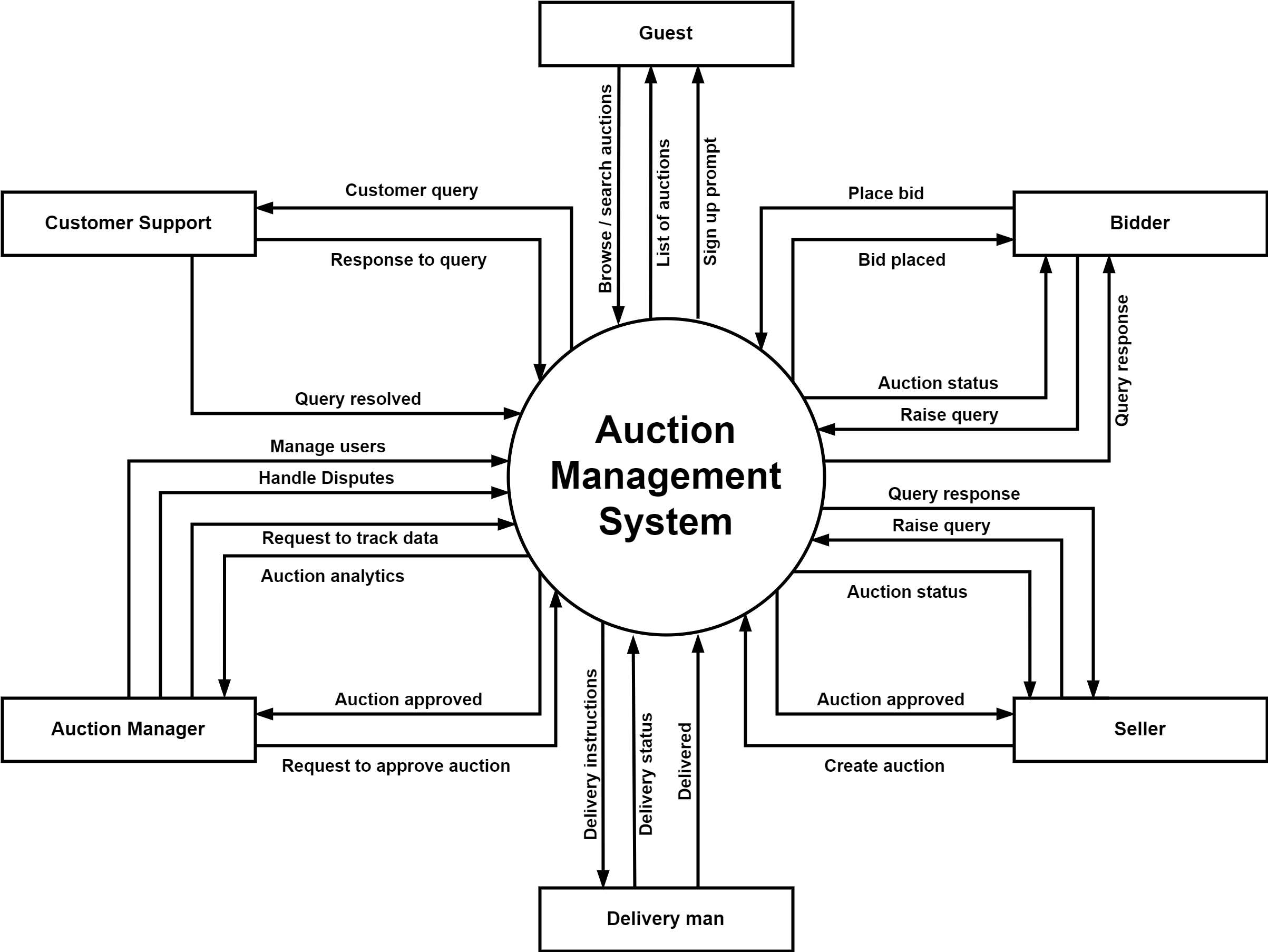


Figure 2.6.1 Context diagram (level-0) for Auction Management System

### DFD Level-1

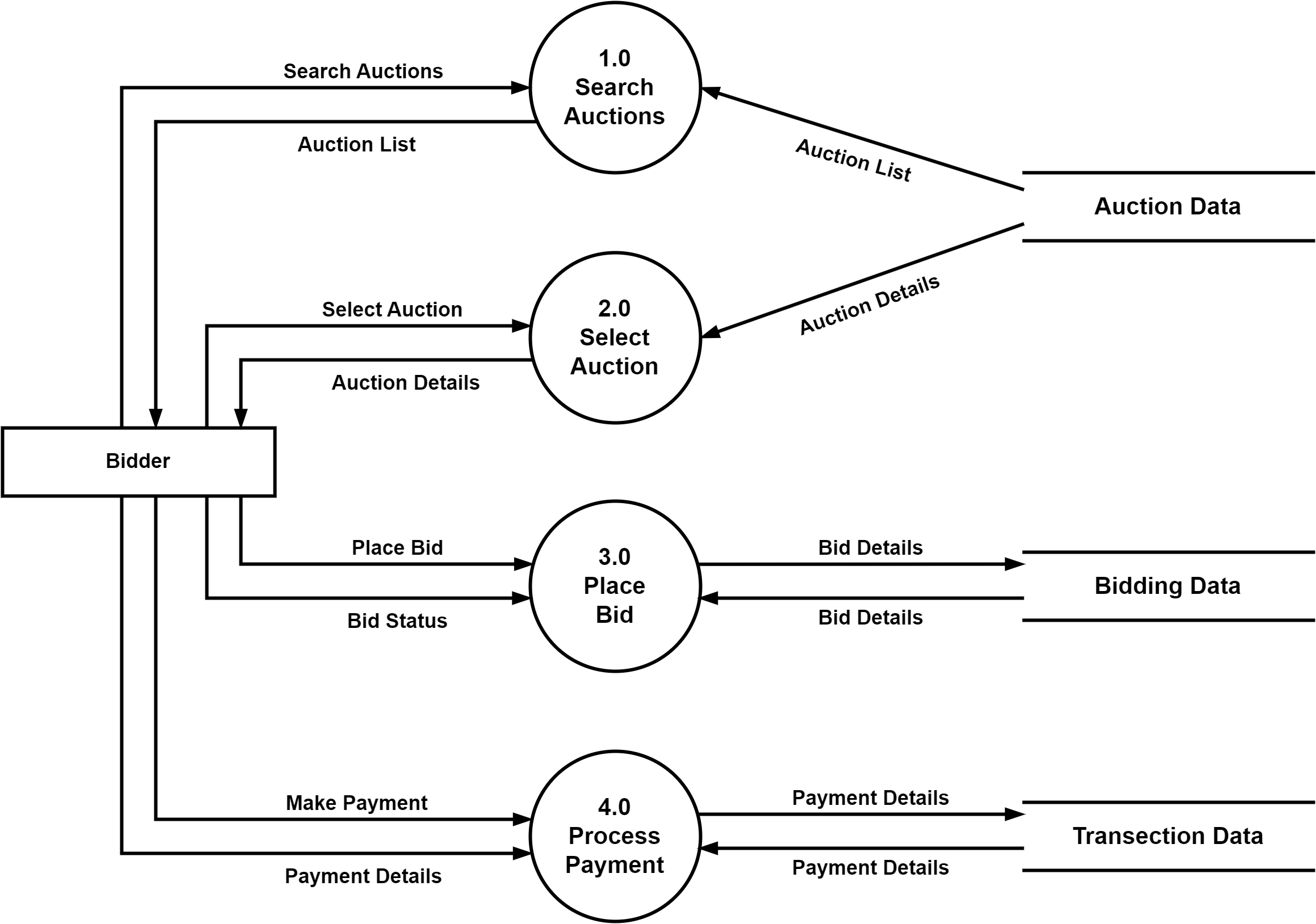


Figure 2.6.2 Data flow diagram level-1 for Auction Management System

### DFD Level-2

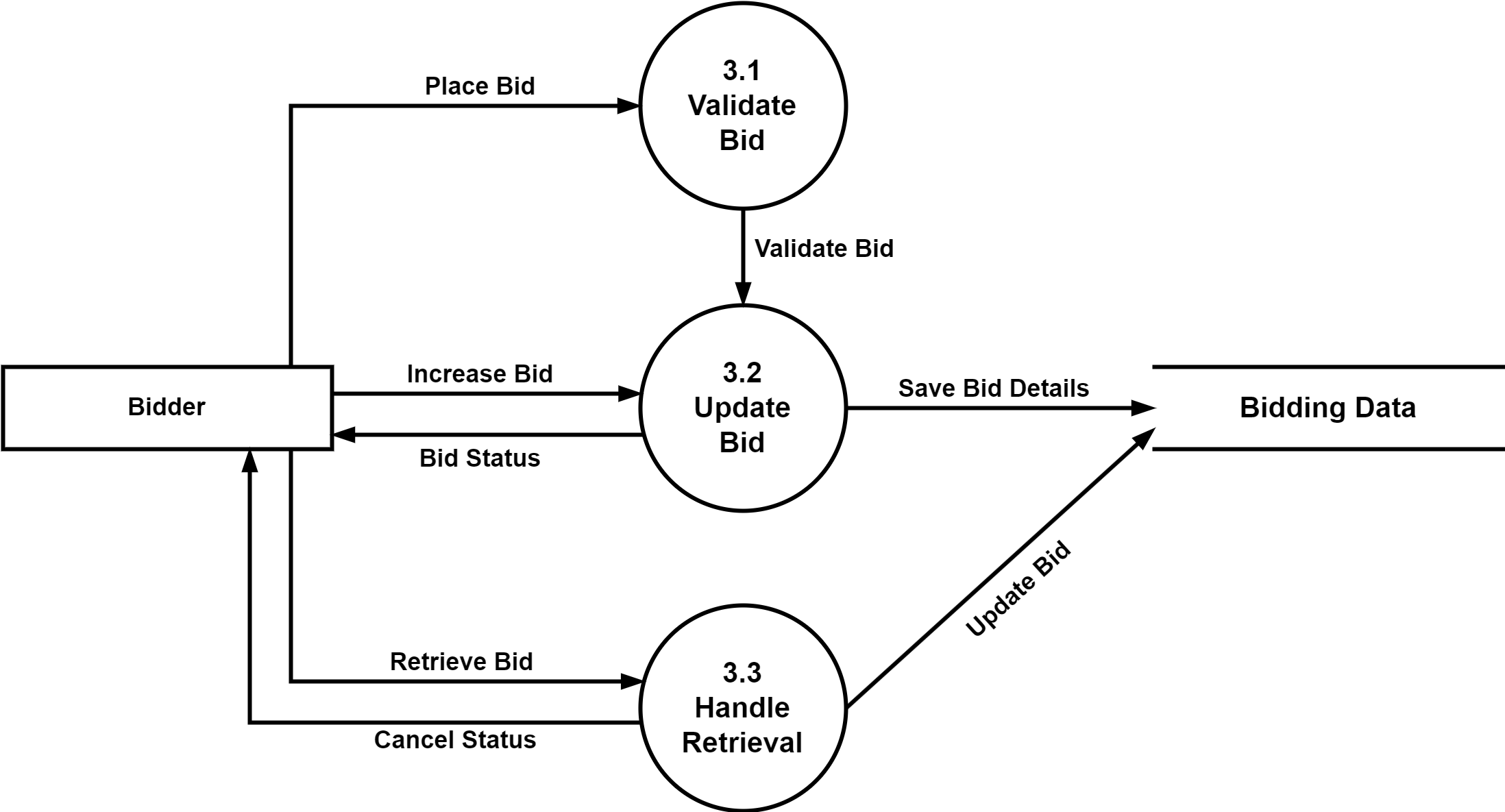


Figure 2.6.3 Data flow diagram level-2 for Auction Management System

# External interface requirement (Screens)

## Screen-1: Registration Form



Figure Screen-1: Registration Form‑1 Screen-1: Registration Form

**Purpose:** This form will allow the target end-users to register in the system. To register , the following information will be encoded in the system.

Table Screen-1: Registration Form‑1 Screen element of Registration form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.** | **Screen Element** | **Input Type** | **O/M** | **1/N** | **Description** |
| **1** | Username | Textbox | M | 1 | Username field should be editable and accept the Username. |
| **2** | Email | Textbox | M | 1 | Email field should be editable and accept the email with proper format. |
| **3** | Password | Password | M | 1 | Password field should be editable and accept the password and display as star or dot. |
| **4** | Retype password | Password | M | 1 | Retype Password field should be editable and accept the password and display as star or dot. |
| **5** | I agree to the terms | Checkbox | M | 1 | Checkbox for accepting terms and condition. |
| **6** | I already have a membership | Link | ------ | ------ | Link for navigate to login page for registered user. |
| **7** | Register | Button | ------ | ------ | Register is a button for store the entered data into database. |

## Screen-2: Login Form



Figure Screen-2: Login Form‑2 Screen-2: Login Form

**Purpose:** This form will be used by the system’s users to access records and features of the system. The users will input the correct combination of their username and password to be able to login to the system.

Table Screen-2: Login Form‑2 Screen element of Login form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.** | **Screen Element** | **Input Type** | **O/M** | **1/N** | **Description** |
| **1** | Username | Textbox | M | 1 | Username field should be editable and accept the Username. |
| **2** | Password | Password | M | 1 | Password field should be editable and accept the password and display as star or dot. |
| **3** | Remember Me | Checkbox | M | 1 | Saving login credentials through remember me checkbox |
| **4** | I forgot my password | Link | ------ | ------ | Link for navigate to Forgot password page for allows users to recover password. |
| **5** | Register a new membership | Link | ------ | ------ | Link for navigate to membership registration. |
| **6** | Sign in | Button | ------ | ------ | Login button navigates to another page even if valid login credentials. |

## Screen-3: Add borrower book



Figure Screen-3: Add borrower book‑3 Screen-3: Add borrower book

**Purpose:** This module will allow the system administrator to add, edit, update or delete borrowers of book. The admin can add borrower information and manage it.

Table Screen-3: Add borrower book‑3 Screen element of Add borrower book

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.** | **Screen Element** | **Input Type** | **O/M** | **1/N** | **Description** |
| **1** | Borrower ID | Textbox | M | 1 | Borrower ID field should be editable and fill autogenerated Borrower ID. |
| **2** | Student ID | Textbox | M | 1 | Student ID field should be editable and accept Student barcode id. |
| **3** | First Name | Textbox | M | 1 | First Name field should be editable and accept only string as a borrower first name. |
| **4** | Middle Name | Textbox | O | 1 | Middle Name field should be editable and accept only string as a borrower middle name. |
| **5** | Last Name | Textbox | M | 1 | Last Name field should be editable and accept only string as a borrower last name. |
| **6** | Contact | Textbox | M | 1 | Contact field should be editable and accept only 10-digit mobile number. |
| **7** | Grade Id | Textbox | O | 1 | Auto fetch detail from the database if proper student id is entered. |
| **8** | Submit | Button | ----- | ----- | Submit is a button for store the entered data into database. |
| **9** | Cancel | Button | ----- | ----- | Cancel is a button for close add borrower dialog. |

# Database design

## List of Tables

* User
* Bidder
* Transaction
* Auction
* Bid

Table List of Tables‑4 Table: User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Null** | **Keys & Constrains** | **Default Value & Description** |
| **UserID** | int | NN | PK (Auto Increment) |  |
| **UserName** | varchar(100) | NN |  |  |
| **Email** | varchar(100) | NN | Unique |  |
| **Phone** | varchar(20) | NN | Unique |  |
| **Address** | varchar(200) | NN |  |  |
| **Role** | varchar(50) | NN | Check(‘User’,’Bidder’,  Seller) |  |
| **Password** | varchar(100) | NN |  |  |

Table List of Tables‑5 Table: Bidder

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Null** | **Keys & Constrains** | **Default Value & Description** |
| **BidderID** | int | NN | PK (Auto Increment) |  |
| **UserID** | int | NN | FK | Reference of Users Table |
| **WalletBalance** | double | NN | Check(WalletBalance ≥ 0) |  |

Table List of Tables‑6 Table: Transaction

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Null** | **Keys & Constrains** | **Default Value & Description** |
| **TransactionID** | int | NN | PK (Auto Increment) |  |
| **BidderID** | int | AN | FK | Reference of Seller Table |
| **AuctionID** | int | NN | FK | Reference of Auction Table |
| **Amount** | double | NN | Check(Amount ≥ 0) |  |
| **TransactionDate** | timestamp | NN |  |  |
| **PaymentMethod** | varchar(50) | NN | Check(‘Cod’,’Net-banking’,’UPI’,’Card’,’Cheque’) |  |

Table List of Tables‑7 Table: Auction

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Null** | **Keys & Constrains** | **Default Value & Description** |
| **AuctionID** | int | NN | PK (Auto Increment) |  |
| **SellerID** | int | NN | FK | Reference of Seller Table |
| **Title** | varchar(100) | NN |  |  |
| **Description** | varchar(800) | NN |  |  |
| **StartBid** | double | NN | Check(StartBid ≥ 0) |  |
| **CurrentBid** | double | NN | Check(CurrentBid ≥ 0) |  |
| **HighestBidderId** | int | AN | FK | Reference of Bidder Table |
| **StartTime** | Timestamp | NN |  |  |
| **EndTime** | Timestamp | NN |  |  |
| **Status** | varchar(50) | NN | Check(‘Active’,’Inactive’,’Ended’) |  |

Table List of Tables‑8 Table: Bid

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Null** | **Keys & Constrains** | **Default Value & Description** |
| **BidID** | int | NN | PK (Auto Increment) |  |
| **AuctionID** | int | NN | FK | Reference of Auction Table |
| **BidderId** | int | NN | FK | Reference of Bidder Table |
| **BidAmount** | double | NN | Check(StartBid ≥ 0) |  |
| **BidTime** | Timestamp | NN |  |  |
| **Status** | varchar(50) | NN | Check(‘Placed’,’Highest’,’Retried’) |  |

# Stories and Scenario

## Story-1: Place a Bid on an Auction

|  |  |  |
| --- | --- | --- |
| *Story #* ***S1*** | **:** | **As a** Bidder,  **I want** **to** place a bid on an auctioned item,  **so that** I can participate in the auction and attempt to purchase the item. |
| **Priority** | **:** | High |
| **Estimate** | **:** | L |
| **Reason** | **:** | Bidders must be able to place bids easily for the auction system to function  effectively. |

### Scenario# S1.1

|  |  |  |
| --- | --- | --- |
| *Scenario#* ***S1.1*** | **:** | Placing a Bid with Valid Information |
| **Prerequisite** | **:** | The bidder is logged into the system and has a valid payment method. |
| **Acceptance Criteria** | **:** | **Given:**  The bidder is on the auction listing page.  **When:** The bidder selects an auction and enters a valid bid amount (higher than the current bid).  **Then:** The system records the bid, updates the auction page, and sends a confirmation notification. |

### Scenario# S1.2

|  |  |  |
| --- | --- | --- |
| *Scenario#* ***S1.2*** | **:** | Placing a Bid with an Invalid Amount |
| **Prerequisite** | **:** | The bidder is logged in. |
| **Acceptance Criteria** | **:** | **Given:** The bidder is on the auction listing page.  **When:** The bidder enters a bid lower than the current highest bid.  **Then:** The system displays an error message stating that the bid must be higher, and the bid is not placed. |

## Story-2: Create an Auction Listing

|  |  |  |
| --- | --- | --- |
| *Story #* ***S2*** | : | **As a** Seller,  **I want to** list an item for auction,  **So that** potential buyers can bid on it. |
| **Priority** | **:** | High |
| **Estimate** | **:** | L |
| **Reason** | **:** | Auctions are the core feature of the platform, and sellers must be able to list items smoothly. |

### Scenario# S2.1

|  |  |  |
| --- | --- | --- |
| *Scenario#* ***S2.1*** | **:** | Successfully Creating an Auction |
| **Prerequisite** | **:** | The seller is logged in. |
| **Acceptance Criteria** | **:** | **Given:**  The seller is on the "Create Auction" page.  **When:** The seller enters valid auction details and uploads item photos.  **Then:** The system publishes the auction, making it available for bidders. |

### Scenario# S2.2

|  |  |  |
| --- | --- | --- |
| *Scenario#* ***S2.2*** | **:** | Notifying Seller of the Auction Winner While Logged Out |
| **Prerequisite** | **:** | The seller has an active auction that has ended. |
| **Acceptance Criteria** | **:** | **Given:** The seller is logged out of the system.  **When:** The auction ends and a winner is determined.  **Then:** The system sends a notification to the seller informing them of the auction winner. |

## Story-3: Approve an Auction Listing

|  |  |  |
| --- | --- | --- |
| *Story #* ***S3*** | **:** | **As an** Auction Manager,  **I want** **to** review and approve auction listings,  **so that** only valid auctions are made public. |
| **Priority** | **:** | Medium |
| **Estimate** | **:** | M |
| **Reason** | **:** | The system should prevent invalid or fraudulent listings from being published. |

### Scenario# S3.1

|  |  |  |
| --- | --- | --- |
| *Scenario#* ***S3.1*** | **:** | Approving a Valid Auction |
| **Prerequisite** | **:** | The auction manager is logged in. |
| **Acceptance Criteria** | **:** | **Given:** The manager is on the "Pending Auctions" dashboard.  **When:** The manager reviews an auction and approves it.  **Then:** The system marks the auction as approved, and the seller is notified. |

### Scenario# S3.2

|  |  |  |
| --- | --- | --- |
| *Scenario#* ***S3.2*** | **:** | Rejecting an Auction with Invalid Details |
| **Prerequisite** | **:** | The auction manager is logged in. |
| **Acceptance Criteria** | **:** | **Given:** The manager is on the "Pending Auctions" dashboard.  **When:** The manager finds missing or inappropriate auction details and rejects the listing.  **Then:** The system notifies the seller about rejection. |

# Test cases

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Name:** | **Auction Management System** | **Test Designed by:** | **Dhruv R. Pithwa** |
| **Module Name:** | **Login** | **Test Designed date:** | 07-03-2025 |
| **Release Version:** | **1.0** | **Test Executed by:** | **R. B. Gondaliya** |
|  |  | **Test Execution date:** | 09-03-2025 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pre-condition**: **Web application should be accessible** | | | | |
| **Test Case ID** | **Test Title** | **Test Type** | **Description** | **Test Case ID** |
| **TC\_001** | Login to web application with valid credential | Functional | Login to Library management system web application through valid credential | TC\_001 |
| **TC\_002** | Login to web application with invalid credential | Functional | Login to Library management system web application through invalid credential | TC\_002 |
| **TC\_003** | Varify login page elements | GUI | varify that all elements are availabe on login page | TC\_003 |

|  |  |
| --- | --- |
| **Test Case Title** | Login to web application with valid credential |
| **Test Type** | Functional |
| **Test Priority** | High |
| **Pre-condition** | Web application should be accessible |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **BUG ID** |
| 1 | Access Web application URL | The site launched properly | Site launched successfully | Pass |  | <https://accounts.google.com/ServiceLogin> |  |
| 2 | Enter valid Username in username field | Username field should be editable and accept the Username | Username input accepted | Pass |  | Username:  Rbgondaliya@gmail.com |  |
| 3 | Enter valid Password in Password field | Password field should be editable and accept the password and display as star or dot | Password input displayed in dot and accepted | pass |  | Password: rbgondaliya |  |
| 4 | Enter valid captcha code in captch field | Captch field should editable and accept captcha and captcha is case sensitive | Captcha input accepted | Pass | Step required when human action validation perform | get captcha from image which is near by captcha field |  |
| 5 | Click on login button | User should login into site and navigated to dashboard | User navigated to dashboard and username should br display in top of the right side. | pass |  |  |  |

|  |  |
| --- | --- |
| **Test Case Title** | Login to web application with invalid credential |
| **Test Type** | Functional |
| **Test Priority** | Medium |
| **Pre-condition** | Web application should be accessible |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **Bug ID** |
| 1 | Verify that User is not able to Login with invalid Username and invalid Password | Should be display an error message enter wrong username or password | Display an error of wrong username and password | Pass |  |  |  |
| 2 | Verify that User is not able to Login with Valid Username and invalid Password | Should be display an error message enter wrong password | Display an error of wrong password | Pass |  |  |  |
| 3 | Verify that User is not able to Login with invalid Username and Valid Password | Should be display an error message User not found | Display an error Username not found | Pass |  |  |  |
| 4 | Verify that User is not able to Login with blank Username or Password | Set required field validation message for Username and Password | Display an error of wrong username and password | Fail | Not performa a validation function fix it |  | Bug\_002 |

|  |  |
| --- | --- |
| **Test Case Title** | Varify login page elements |
| **Test Type** | GUI |
| **Test Priority** | Medium |
| **Pre-condition** | Web application should be accessible |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **Bug ID** |
| 1 | Launch application with the given url | The site launched properly | Site launched successfully | Pass |  | <https://accounts.google.com/ServiceLogin> |  |
| 2 | Verify that the login screen contains elements such as Username, Password, Sign in button, Remember password check box, Forgot password link, and Create an account link. | All listed control displayed properly on the page | Login page loaded successfully | Pass |  |  |  |
| 3 | Verify that cursor is focused on “Username” text box on the page load | Cursor is focused in Username textbox | Cursor focus in Username textbox | Pass |  |  |  |
| 4 | Verify that tab functionality is working properly or not | When tab pressed cursor move in next control | Cursor moving in next control | Pass |  |  |  |
| 5 | Verify that all the fields such as Username, Password has a valid placeholder | All text fields have proper placeholder | All text fields have proper placeholder | Pass |  |  |  |
| 6 | Verify that the labels float upward when the text field is in focus or filled (In case of floating label) | When field is focused or filled, label display on top of the filled | When field is focus or filled, label display on top of the filled | Pass | step required when fields with floating label |  |  |
| 7 | verify that forgot password link working properly | when click on forgot password load forgot passworg page | forgot password link not working | Fail |  |  |  |

# References

* http://www.w3schools.com/html/html\_intro.asp
* https://www.w3schools.com/php/default.asp
* https://www.javatpoint.com/uml