

Online Auction App Final Year Project Report

Submitted by M.Umair(491-2021)

Bismah Imran (1797-2021)

Maham Mirza(1922-2021) Supervisor

Osama Ahmed Khan

Inpartialfullfillmentoftherequirementsforthedegreeof Bachelor of Science in Computer Science

2025

**FacultyofEngineeringSciencesandTechnology** Hamdard Institute of Engineering and Technology Hamdard University, Main Campus, Karachi, Pakistan

# CertificateofApproval

****

**FacultyofEngineeringSciencesandTechnology**

HamdardInstituteofEngineeringandTechnology Hamdard University, Karachi, Pakistan

This project “Online Auction App” is presented by Bismah Imran, M.Umair, Maham Mirza under the supervision of their project advisor and approved by the project examination committee, and acknowledged by the Hamdard Institute of Engineering and Technology, in the fulfillment of the requirements for the Bachelor degree of Computer Science.

Mr.OsamaAhmed In-chargeFYP-Committee

(ProjectSupervisor)

Mr.XYZ Chairman

(ProjectCo-Supervisor) (DepartmentofComputing)

(Dean,FEST)

# Authors’Declaration

Wedeclarethatthisprojectreportwascarriedoutinaccordancewiththerulesandregulations of Hamdard University. The work is original except where indicated by special references in the text and no part of the report has been submitted for any other degree. The report has not been presented to any other University for examination.

Dated:

AuthorsSignatures:

M.Umair

Bismah Imran

MahamMirza

# PlagiarismUndertaking

We, M.Umair ,Bismah Imran, and Maham Mirza solemnlydeclare that the work presented in the Final Year Project Report titled Online Auction App has been carried out solely by ourselves with no significant help from any other person except few of those which are duly acknowledged.Weconfirmthatnoportionofourreporthasbeenplagiarizedandanymaterial used in the report from other sources is properly referenced.

Dated:

AuthorsSignatures:

M.Umair

Bismah Imran

MahamMirza

# Acknowledgments

**DocumentInformation**

Table1:Document Information

|  |  |
| --- | --- |
|  |  |
| Customer |  |
| ProjectTitle | OnlineAuctionApp |
| Document | FinalYearProjectReport |
| Document Version | 1.0 |
| Identifier | FYP-026/FL24FinalReport |
| Status | Final |
| Author(s) | BismahImran |
| Approver(s) | OsamaAhmedKhan |
| IssueDate |  |

#### DefinitionofTerms,Acronyms,andAbbreviations

* **Auction**:Aprocessofsellingitemstothehighestbidderthroughbidding,replicated digitally in the app.
* **Real-timeBidding**:Instantupdatesandbidplacements,enablingadynamicauction environment.
* **Firebase**:Acloud-basedplatformforappbackendoperations,including authentication and real-time data.
* **Flutter**:A UItoolkitused forbuildingresponsiveand cross-platform applications.
* **Dart**:ProgramminglanguageusedwithFlutterforappdevelopment.
* **MobileCompatibility**:Theabilityoftheapptofunctionsmoothlyonsmartphones and tablets.
* **ChatSystem**:Anin-appmessagingfeatureenablingdirectcommunicationbetween buyers and sellers.
* **Encryption**:Asecurityfeatureconvertingdataintoasecureformattoprevent unauthorized access.
* **Notifications**:Alertssenttousersaboutauctionupdates,bids,or messages.
* **AdminModule**:Abackendfeatureforadministratorstomanageauctions,users,and disputes.

# Abstract

The **Online Auction App** is a modern platform that makes buying and selling items more engagingandflexible.Unliketraditionalshoppingwebsiteswithfixedprices,thisappallows userstobidforitemsinreal-time,ensuringcompetitivedealsforbuyersandbetterprofitsfor sellers. Built with mobile users in mind, the app provides a user-friendly experience with features like instant notifications, secure logins, and a chat system for direct communication. Backed by reliable technologies like Firebase and Flutter, it ensures smooth and secure operations. Whether you're looking to buy unique items or sell to a wider audience, the app offers an exciting and interactive auction environment.

# TableofContents

|  |  |
| --- | --- |
| [CertificateofApproval](#_bookmark0) | [2](#_bookmark0) |
| [Authors’Declaration](#_bookmark1) | [3](#_bookmark1) |
| Acknowledgment | [5](#_bookmark2) |
| [Document Information](#_bookmark3) | [6](#_bookmark3) |
| [Abstract](#_bookmark4) | [7](#_bookmark4) |
| [Chapter1INTRODUCTION](#_bookmark5) | [10](#_bookmark5) |
| [Descriptionabout Project](#_bookmark6) | [10](#_bookmark6) |
| [DetailsabouttheDomain](#_bookmark7) | [10](#_bookmark7) |
| [RelevantBackground](#_bookmark8) | [10](#_bookmark8) |
| [Chapter2RELEVANT BACKGROUND& DEFINITIONS](#_bookmark9) | [11](#_bookmark9) |
| [Chapter3 LITERATUREREVIEW&RELATEDWORK](#_bookmark10) | [12](#_bookmark10) |
| [LiteratureReview](#_bookmark11) | [12](#_bookmark11) |
| [RelatedWork](#_bookmark12) | [12](#_bookmark12) |
| [Gap Analysis](#_bookmark13) | [12](#_bookmark13) |
| [Chapter4METHODOLOGY](#_bookmark14) | [13](#_bookmark14) |
| [SoftwareEngineeringMethodology](#_bookmark15) | [13](#_bookmark15) |
| [ProjectMethodology](#_bookmark16) | **Error!Bookmarknotdefined.** |
| [Chapter5EXPERIMENTALEVALUATIONS&RESULTS](#_bookmark16) | [15](#_bookmark16) |
| [Evaluation Testbed](#_bookmark17) | [15](#_bookmark17) |
| ResultsandDiscussion | 15 |
| [Chapter6CONCLUSIONAND DISCUSSION](#_bookmark17) | [16](#_bookmark17) |
| [LimitationsandFutureWork](#_bookmark18) | [16](#_bookmark18) |
| ReasonsforFailure–If Any | 16 |
| [REFERENCES](#_bookmark19) | [17](#_bookmark19) |

[APPENDICES 18](#_bookmark20)

[A0.CopyofProject Registration Form 19](#_bookmark21)

[A1a.ProjectProposalandVisionDocument 20](#_bookmark22)

[A1b.Copyof ProposalEvaluation Comments by Jury 21](#_bookmark23)

[A2.RequirementSpecifications 22](#_bookmark24)

[A3.DesignSpecifications 23](#_bookmark25)

[A4.OtherTechnicalDetail Documents 24](#_bookmark26)

[TestCasesDocument 24](#_bookmark27)

[UI/UXDetail Document 24](#_bookmark28)

[CodingStandardsDocument 24](#_bookmark29)

[ProjectPolicyDocument 24](#_bookmark30)

UserManua1Document 24

[A5.Flyer &PosterDesign 25](#_bookmark31)

[A6.CopyofEvaluationComments 26](#_bookmark32)

[Copyof Evaluation Comments bySupervisorfor Project–IMid SemesterEvaluation 26](#_bookmark33)

[Copyof Evaluation Comments bySupervisorfor Project–IEnd SemesterEvaluation 27](#_bookmark34)

[Copyof EvaluationComments byJuryfor Project–IEnd Semester Evaluation 28](#_bookmark34)

[Copyof Evaluation Comments bySupervisorfor Project–IIMid Semester Evaluation 29](#_bookmark35)

Copyof Evaluation CommentsbyJuryfor Project–IIEnd Semester Evaluation 31

[A7.Meetings’ Minutes 32](#_bookmark36)

[A8.DocumentChangeRecord 33](#_TOC_250001)

[A9.ProjectProgress 34](#_TOC_250000)

A10.ResearchPaper **Error!Bookmarknotdefined.**

#### ListofFigures

**Figure No Description Page No.**

Figure1.1:ScopeofProject 2

Figure2.1:FaceDetection 4

Figure2.2:FaceRecognitionProcess 5

Figure4.1:SystemflowforFaceRecognition 9

Figure4.2:UseCase Model 10

Figure4.3:UseCaseDiagramforsetting 12

Figure5.1:Designofa System 13

Figure6.1:Step1downloadsOpenCVmanager 18

Figure6.2:Step2installsapplication 19

Figure6.3:Step3Open Application 20

Figure6.4:Step4openinbox 21

Figure6.5:Step5selectmessage 22

Figure6.6:Step6textvariations 23

Figure6.7:Step7readcontacts 24

Figure6.8:Step8forwritemessage 25

Figure6.10:Step10settings 26

|  |  |  |
| --- | --- | --- |
|  | **ListofTables** |  |
| **Table No.**  Table2.1:ComparisonTable | **Description** | **PageNo.**  6 |
| Table3.1:PhasesofProject |  | 7 |
| Table5.1:TestCase1 |  | 16 |
| Table5.2:TestCase2 |  | 16 |
| Table5.3:TestCase3 |  | 16 |
| Table5.4:TestCase4 |  | 17 |
| Table5.5:TestCase5 |  | 17 |
| Table5.6:TestCase6 |  | 17 |

### CHAPTER 1 INTRODUCTION

The Online Auction App is a cutting-edge platform designed to make buying and selling items easier and more exciting. Unlike traditional e-commerce websites where items have fixedprices,thisappallowsuserstoparticipateinliveauctions,givingbuyersthechanceto snag great deals and sellers the opportunity to get the best possible price for their items.

Whether you're looking to auction off collectibles, electronics, or even vehicles, this app offersaseamlessandfunwaytoconnectwithpotentialbuyersandsellersfromanywhere.

One of the key features of the app is its real-time bidding system. Users can see the current bids and place their own offers in a matter of seconds. This creates a dynamic marketplace wherecompetitiondrivesbetterpricesforeveryone.Tokeepthingssecureandorganized,the app also has user profiles, where buyers and sellers can manage their information, track their bidding history, and communicate directly through a built-in chat system. Notifications keep users updated on the status of their auctions, ensuring they never miss out on important updates.

Designed for mobile devices, the app is accessible and easy to use, even for those who are alwaysonthego. Itisbuiltusingmoderntechnologytoensurefastandreliableperformance, with a focus on keeping user data safe and secure. Whether you're a seasoned bidder or just getting started, the Online Auction App provides a simple, secure, and engaging way to explore the world of online auctions.

#### Motivation

Themotivation behind developingthe Online Auction App stems from the need to create a more interactive and flexible marketplace for buying and selling goods. Traditional e- commerce platforms often lack the dynamic nature of auctions, where buyers can bid for itemsandpotentiallygetthematalowerprice,whilesellerscanbenefitfromcompetitive bidding. The app aims to bridge this gap by providing a platform that offers real-time bidding, enhancing the shopping experience for both buyers and sellers. It encourages a fair marketplace where users can engage more actively and enjoy the thrill of auctions from the comfort of their mobile devices.

#### ProblemStatement

Current e-commerce applications typically operate on fixed pricing models, which do not allow for price negotiation or bidding. This static pricing approach limits both buyers and sellers, as buyers cannot compete for better deals, and sellers cannot maximize their profits throughcompetitivebidding.Theabsenceofareal-timeauctionsysteminmoste-commerce apps creates a gap in the market, leaving users seeking more dynamic and flexible options.

TheOnlineAuctionAppaddressesthisissuebyintroducingaplatformthatfacilitatesreal- time bidding, thereby creating a more engaging and profitable environment for all users.

#### GoalsandObjectives

TheprimarygoaloftheOnlineAuctionAppistoprovideasecure,user-friendlyplatformfor online auctions, where users can buy or sell goods dynamically through bidding. The objectives include:

* + - Enhancingthebuyingexperiencebyofferingcompetitivedealsthroughreal-time auctions.
    - Empoweringsellerstomaximizetheirprofitsbyreachingabroaderaudienceand receiving competitive bids.
    - Facilitatingreal-timeinteractionsbetweenbuyersandsellersfornegotiationsand updates.
    - Ensuringtheappisoptimizedformobiledevices,providingaseamlessuser experience on the go.

#### ProjectScope

Theprojectscopeencompassesseveralkeyfeaturesandfunctionalitiestoensurea comprehensive online auction platform:

* + - **UserManagement**:Userscanregister,login,andmanagetheirprofiles securely.
    - **AuctionListings**:Userscanlistitemsforauction,includingdetailslikestartandend times, and minimum bids.
    - **Bidding**:Theappsupportsreal-timebidding,allowinguserstoplaceandtrackbids dynamically.
    - **ChatSystem**:Buyersandsellerscancommunicatethroughachatsystemtonegotiate and discuss terms.
    - **Administration**:Anadminmoduletomanageuseraccounts,auctions,andresolve disputes.
    - **Notifications**:Real-timenotificationsforupdates onbidsandauctionstatuses.
    - **MobileCompatibility**:Theappisdesignedtoberesponsiveandaccessibleonall mobile devices, ensuring a broad user base and ease of use.

**CHAPTER2**

**RELEVANTBACKGROUND&DEFINITIONS**

# RelevantBackground& Definitions

TheOnline Auction App represents a significant advancement in the waygoods and services are bought and sold in the digital marketplace. To fully understand its importance and functionality, it is essential to delve into thebackground of onlineauctions, the technological infrastructuresupportingtheapp,andthedefinitionsofkeytermsandconceptsrelatedtothis domain.

##### HistoricalContextofOnlineAuctions

Onlineauctionshave evolvedfromtraditionalauctionhouses,wherebiddersgatherinperson to compete for items, to digital platforms that allow participants to engage in the auction process from anywhere in the world. The first significant foray into online auctions began in the mid-1990s with platforms like eBay, which revolutionized the way consumers could buy and sell items. These platforms introduced the concept of bidding on items over the internet, openingupnewopportunitiesforbothbuyersandsellers.Theconvenienceofparticipatingin auctions from the comfort of one's home, coupled with the potential for finding rare items or getting good deals, led to the rapid growth of online auction platforms.

As technology advanced, so did the features of online auction systems. Today, these platforms support real-time bidding, mobile compatibility, and secure payment systems, making them more accessible and user-friendly than ever before. The development of the OnlineAuctionAppbuildsonthislegacy,incorporatingmoderntechnologicaladvancements to provide a seamless, interactive, and secure auction experience.

##### TechnologicalInfrastructure

TheOnline Auction App is built usingFlutterand Dart, acombination that allows for the development of highly responsive, cross-platform mobile applications. Flutter, an open- source UIsoftware development kit byGoogle, enables the creation of nativelycompiled applicationsformobile,web,anddesktopfromasinglecodebase. Itprovidesarichsetof

pre-designedwidgetsandtools,makingiteasierfordeveloperstobuildaestheticallypleasing and functional user interfaces.

**Dart**, the programming language used with Flutter, is designed for client development, such asforwebandmobileapps.Itisoptimizedforfastdevelopment,offeringasmoothand quick development cycle, which is crucial for dynamic applications like the Online Auction App.

TheuseofFlutterandDartensuresthattheappis notonlyvisuallyappealingbutalso efficient and fast.

ThebackendoftheappispoweredbyGoogleFirebase,acomprehensiveplatformforapp development. Firebase provides various services such as real-time database management,

user authentication, and cloud messaging. These features are integral to the functionality of theOnlineAuctionApp,allowingittohandlereal-timedataupdates,secureuserlogins,and instant notifications effectively.

##### DefinitionsandKeyConcepts

To fully grasp the workings and benefits of the Online Auction App, it is important to understandseveralkeyconceptsandtermsassociatedwithonlineauctionsandtheapp’s features:

1. **Auction**: A public sale in which goods or property are sold to the highest bidder. Onlineauctionsreplicatethisprocessdigitally,allowinguserstobidonitemsthrough an internet platform.
2. **Real-time Bidding**: A process in which auction bids are placed, updated, and displayedinstantaneously.Thisfeatureiscrucialforonlineauctions,asitprovidesa dynamic and engaging user experience.
3. **User Authentication**: A security process that ensures only authorized users can accesscertainareasoftheapporperformspecificactions.TheOnlineAuctionApp uses Firebase Authentication to manage user sign-ins, sign-ups, and email verification.
4. **Firebase Real-time Database**: A cloud-hosted NoSQLdatabase that allows data to bestoredandsynchronizedinreal-timeacrossallconnectedclients.Thisisessential for keeping auction listings and bid updates current and consistent for all users.
5. **Mobile Compatibility**: The ability of an app to function seamlessly across various mobiledevices,ensuringthatuserscanaccessandusetheappfromtheirsmartphones or tablets without any issues.
6. **Chat System**: A built-in messaging platform that enables users to communicate directlywitheachother.InthecontextoftheOnlineAuctionApp,thisallowsbuyers and sellers to discuss items, negotiate prices, and finalize transactions more effectively.
7. **Admin Module**: A part of the app designed for administrators to manage auctions, overseeuseraccounts,andresolvedisputes.Thisensuresthesmoothoperationofthe platform and maintains the integrity of the auction process.
8. **Notifications**:Alertssenttouserstoinformthemofimportantupdates,suchasbeing outbid on an item or an auction ending soon. Notifications are crucial for keeping users engaged and informed about their activities on the platform.
9. **Encryption**:Theprocessofconvertingdataintoasecureformatthatcannotberead by unauthorized users. This is an important feature for protecting sensitive user information and ensuring the security of transactions.
10. **User Interface (UI)**: The part of the app that users interact with. A well-designed UI iscrucialforprovidingapositiveuserexperience,makingiteasyforuserstonavigate the app and participate in auctions.

##### TheRoleofOnlineAuctionAppsinModernCommerce

Onlineauctionappsliketheonedescribedinthisdocumentplayapivotalroleinmodern commerce by offering a flexible and dynamic alternative to traditional e-commerce platforms. Theycaterto a varietyof users, includingindividual sellers lookingto reach a

broaderaudience,collectorsseekingrareitems,andbusinessesaimingtomaximizetheir revenue streams.

Theseappsbenefitusersby:

* + **Providing Competitive Prices**: Buyers can potentially get items at lower prices throughcompetitivebidding,whilesellerscanachievehigherprofitsbyattracting multiple bidders.
  + **ExpandingMarketReach**:Sellerscanreachaglobalaudience,increasingthe chances of finding the right buyer for their items.
  + **EnhancingUserEngagement**:Theinteractivenatureofauctions,withreal-time updates and competitive bidding, creates a more engaging shopping experience compared to fixed-price models.

##### ChallengesandConsiderations

Whileonlineauctionappsoffernumerousbenefits,theyalsopresentcertainchallengesthat need to be addressed:

* + **Security Concerns**: Protecting user data and ensuring secure transactions is paramount.Implementingrobustauthenticationmethodsanddataencryptionis essential to gaining user trust.
  + **UserExperience**:Theappmustbeeasytouseandnavigate,evenforthosewhomay not be tech-savvy. A cluttered or confusing interface can deter users from participating in auctions.
  + **Real-timePerformance**:Theappmusthandlereal-timedataefficientlytoprovidea seamless experience. Any delays or lags can frustrate users and impact the overall success of the auctions.

### CHAPTER3

**LITERATUREREVIEW&RELATEDWORK**

#### LiteratureReview

Online auctions have been a popular method of buying and selling goods for years, revolutionizedbyplatformslikeeBayandsimilarwebsites.Theseplatformsintroducedthe concept of bidding on items online, allowing users to participate in auctions without being physically present. Over time, this model has evolved with advancements in technology, leading to the development of more sophisticated auction systems that are faster, more secure, and accessible through mobile devices.

Several studies and developments have focused on enhancing online auction platforms. Research has been conducted on improving user experience, ensuring data security, and optimizing real-time bidding processes. Various papers have highlighted the importance of userengagementandtheroleofinstantnotificationsinkeepingusersinformedandinvolved. There has also been significant work on using mobile technology to make auction platforms moreaccessible, catering to thegrowingnumber of users who prefer conductingtransactions via their smart phones

#### RelatedWork

1. **eBay and Other Online Platforms**: eBay was one of the first major online auction platforms, setting the standard for online auctions. It allowed users to bid on a wide range of items, from collectibles to electronics, and introduced features like auction timers, feedback systems, and secure payment methods. Other platforms, such as AmazonAuctionsandYahoo!Auctions,followedsuit,offeringsimilarserviceswith varying degrees of success.
2. **MobileAuction Apps**:Withtheriseofmobiletechnology,severalauctionplatforms developed apps to cater to mobile users. These apps focused on providing a seamless experienceonsmallerscreens,withfeatureslikeswipenavigation,pushnotifications, and mobile payment options. Research in this area has shown that mobile compatibility significantly increases user engagement and bidding activity.
3. **SecurityEnhancements**:Ensuringthesecurityofuserdataandtransactionshasbeen a critical area of focus. Studies have explored the use of encryption, secure login systems, and fraud detection algorithms to protect users and maintain the integrity of the auction process. Platforms that fail to implement robust security measures often face user distrust and potential legal issues.
4. **User Engagement and Gamification**: Some platforms have experimented with gamification elements to enhance user engagement. Features like reward points for participation,leaderboards,andinteractivetutorialshavebeenshowntoincreaseuser activity and retention. Research indicates that creating a fun and engaging environment can lead to higher bidding rates and user satisfaction.

#### GapAnalysis

Despitetheprogressmadeinonlineauctionplatforms,therearestillseveralgapsandareas for improvement:

1. **LackofReal-timeInteraction**:Manytraditionalauctionplatforms,whileeffective, lack real-time interaction capabilities. This gap is particularly noticeable in mobile apps, where users expect instant feedback and updates. The Online Auction App addresses this gap by integrating real-time bidding and notifications, ensuring users are always up-to-date with the latest auction status.
2. **Limited Mobile Optimization**: While many platforms offer mobile apps, not all are fully optimized for mobile use. Issues like slow loading times, difficult navigation, andlimitedfunctionalitycandeterusersfromparticipatinginauctionsontheirmobile devices. The Online Auction App is designed with mobile-first principles, ensuring a smooth and responsive experience on all mobile devices.
3. **Security Concerns**: Despite advances in security technology, some platforms still struggle with protectinguser data and ensuringsafetransactions. TheOnlineAuction Appfillsthisgapbyusingsecureauthenticationmethodsandencryptiontosafeguard user information, providing a secure environment for both buyers and sellers.
4. **User Experience and Accessibility**: Many existing platforms can be overwhelming or confusing for new users, with complex interfaces and jargon-filled instructions. This limits the platform’s reach and user base. The Online Auction App focuses on providingauser-friendlyinterfacewithclearinstructionsandeasynavigation,making it accessible to users of all technical abilities.
5. **Limited Communication Features**: Communication between buyers and sellers is oftenlimitedtopost-saleinteractionsorexternalmessagingplatforms.Thiscanlead to misunderstandings and disputes. The Online Auction App includes an integrated chat system, allowing users to communicate directly within the app, fostering transparency and reducing the likelihood of disputes.

**CHAPTER4**

## PROJECT DISCUSSION

#### SoftwareEngineeringMethodology

To develop the Online Auction App, we adopted the **Agile Software Development Methodology**. This methodologyis known for its flexibility, iterative development, and continuous delivery approach. Agile was chosen because it fits perfectly with the requirementsofamodernmobileapplicationwhereuserfeedbackandquickupdatesare important.

In Agile, the work is broken down into small time-bound cycles known as **sprints**. Each sprint focuses on developing specific features and includes planning, design, development, testing,andfeedback.Thisallowedustograduallybuildandimprovetheappwhileensuring high-quality results. Regular meetings and reviews were conducted to track progress and make changes where necessary.

#### ProjectMethodology

Ourprojectmethodologyfollowsa**step-by-stepanduser-focusedapproach**,structuredas follows:

1. **RequirementAnalysis**:Understandingwhatfeaturesusersneed,suchasbidding, chat, and secure login.
2. **Planning**:Dividingtheprojectintophasesandassigningtaskstoteammembers.
3. **Design**:CreatingwireframesandUIdesignsusingFluttertoensureacleanand responsive interface.
4. **Development**:Writingcodeforfrontendandbackend,integratingfeatureslike Firebase authentication, real-time bidding, and push notifications.
5. **Testing**:Checkingeachmoduleforbugsandperformanceissuesafter everysprint.
6. **Deployment**:Finalizingtheappandpreparingitforreleaseonmobile platforms.
7. **Maintenance**:Makingimprovementsbasedonuserfeedbackandfixingpost- deployment issues.

Thismethodologyhelpedusstayorganized andfocused throughout theproject.

#### PhasesofProject

Theproject was divided intothefollowingmajorphases:

#### PlanningandResearch

* + - * Identifiedtheproject goalsandmainfeatures.
      * Researchedexistingauctionplatformstogatherideasand inspiration.

#### Design

* + - * CreatedUIwireframes anddefineduserflow.
      * Designedtheoveralllayout andexperienceof theappusingFlutter.

#### Development

* + - * SetupFirebaseforauthenticationandreal-timedatabase.
      * Developedfeatureslikelogin/signup,biddingsystem,notifications,andchat.

#### Phase4–Testing

* + - * Conductedmanualtestingforeachfeature.
      * Fixedbugsandimproved appspeedanduser experience.

#### Phase5–Deployment

* + - * Finalizedandpublishedtheapplication.
      * Preparedprojectdocumentationandpresentationmaterials.

#### Software/ToolsUsedinProject

Belowarethe mainsoftware and toolsused duringdevelopment:

|  |  |
| --- | --- |
| **Tool/Software** | **Purpose** |
| **Flutter** | Usedtodevelop themobileapp (frontend)withcross-platform support. |
| **Firebase** | Backendservicesincludingauthentication,real-timedatabase,and cloud storage. |
| **VisualStudio Code** | Codeeditorusedfor writingandmanagingtheapp's codebase. |
| **Figma** | ForUI/UXdesignand wireframing. |
| **GitHub** | Versioncontrolandprojectcollaboration. |
| **AndroidStudio** | Emulatorand testingenvironment formobile app. |

#### HardwareUsedinProject

Thedevelopmentofthe projectrequired basic andwidelyavailable hardware:

|  |  |
| --- | --- |
| **Hardware** | **Specification/Purpose** |
| **Laptop/PC** | Corei5 or higher, 8GBRAM – used fordevelopment and testing |
| **Smartphone** | Androiddevice forreal-timeapp testing |
| **InternetConnection** | RequiredforFirebaseservices,GitHub,and collaboration |
| **StorageDevices** | USBdrives and harddrives forbackupand data transfer |

**Chapter5**

## IMPLEMENTATION

* 1. **ProposedSystemArchitecture/Design**

TheOnlineAuctionAppisdesignedwitha **client-serverarchitecture**using**Flutter**forthe frontend and **Firebase** for the backend. The architecture ensures real-time communication, secure authentication, and seamless user experience across devices.

###### ClientSide(MobileApp)

* + - * BuiltwithFluttertosupportAndroidandiOS platforms.
      * Handlesuserinteractions likebidding,login, chat,andviewingauctions.
      * CommunicateswithFirebasefordatastorageandupdates.

###### Backend(FirebaseServices)

* + - * **FirebaseAuthentication**:Managessecureuserloginand registration.
      * **CloudFirestore**:Storesauctionitemdata,userprofiles,andbidsinreal-time.
      * **FirebaseCloudMessaging(FCM)**:Sendsinstantnotificationsforbidupdates.
      * **FirebaseStorage**:Storesitemimagesanduseruploadssecurely.

###### SystemFlow

1. Userlogsinorregistersviatheapp.
2. Firebaseverifiescredentialsandprovidesaccess.
3. Userscanlistitems,placebids,orcommunicatevia chat.
4. Updates(e.g.,newbidplaced)arereflectedinstantlyusingreal-timedatabase features.

## FunctionalSpecifications

Thesearethefeatures andoperationsthesystem isexpectedto perform:

* **UserAuthentication**:Signup,login,andlogoutfunctionality.
* **ItemListing**:Sellerscanpostproductsforauctionwithimages,description,and starting bid.
* **BiddingSystem**:Buyers canplacebidsonitemsinreal-time.
* **Notifications**:Instantalertsforbidupdatesand auctionresults.
* **ChatSystem**:Directcommunicationbetweenbuyer andsellerwithintheapp.
* **ProfileManagement**:Userscanmanagepersonalinformationandviewpast auctions/bids.

## Non-FunctionalSpecifications

Theserefer tothequalityattributesof thesystem:

* **Performance**:Theappprovidesfastresponsetimeswithminimal delays.
* **Scalability**:Firebasesupportsmultipleconcurrentuserswithoutperformance degradation.
* **Security**:UserdataandauthenticationareprotectedusingFirebaseAuthentication and secure storage.
* **Usability**:Theapphasaclean,intuitiveuserinterfacedesignedforeaseofuseon mobile devices.
* **Reliability**:Real-timeupdatesandconsistentdatasynchronizationensuresmooth operation.
* **Portability**:Theappruns on bothAndroid and iOSplatforms using Flutter.

## Testing

Theapplicationwastestedthoroughlytoensureallfeaturesworkasintendedandthatthere are no major bugs or issues. Manual testing was conducted during and after each development sprint.

###### TypesofTestingUsed

* + - * **UnitTesting**:Individualcomponentslikelogin,bidding,anditemlistingweretested separately.
      * **IntegrationTesting**:Combinedmodulessuchasbiddingwithnotificationsanditem display with chat were tested together.
      * **UITesting**:Ensuredthattheuserinterfaceworkssmoothlyondifferentscreensizes and devices.
      * **RealDeviceTesting**:TheappwastestedonactualAndroidsmartphonestoevaluate performance and usability.

## Purpose ofTesting

Themainobjectives oftestingtheOnlineAuctionApp were:

* Toverifythatallfeatureswork correctly.
* Toidentifyand fix bugsduringdevelopment.
* Toensuretheapp is user-friendlyand providesasmooth experience.
* Tomakesuretheapp meetsall functionaland non-functional requirements.
* Tocheckthe real-timeperformanceoffeatureslikebiddingandmessaging.

## Test Cases:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TestCase** | **Description** | **ExpectedResult** | **Actual Result** | **Status** |
| Loginwithvalid credentials | Userenterscorrect email & password | Userlogsin successfully | Worksas expected | Pass |
| Loginwithwrong password | Userentersincorrect password | Showerror message | Showserror message | Pass |
| Addnewauction item | Sellerfillsformand uploads image | Itemislisted successfully | Workscorrectly | Pass |
| Placeabid | Userplacesahigherbid than current | Bidisacceptedand updated | Bidupdatesin real-time | Pass |
| Notificationon new bid | Newbidisplacedon item | Notifysellerand other bidders | Notifications received | Pass |
| Startchatwith seller | Buyerclicksonchat button | Chatopenswith correct user | Chatfunctions properly | Pass |

**Chapter5**

**EXPERIMENTALEVALUATIONS&RESULTS**

## EvaluationTestBed

Toevaluatetheperformanceandreliabilityofthe **OnlineAuctionApp**,weconductedtests usingrealdevicesandaliveFirebasebackend environment.Theevaluationwasfocusedon responsiveness, usability, real-time data handling, and system behavior under various conditions.

###### HardwareUsed

* + - * **Smartphone**:Android11+, 4GBRAM (foruser testing)
      * **Laptop/PC**:Corei5,8GBRAM(fordevelopmentandmonitoringbackend operations)
      * **InternetConnection**:Stable10MbpsWi-Ficonnection

###### SoftwareEnvironment

* + - * **FlutterSDK**: Formobiledevelopment
      * **FirebaseConsole**:Real-timedatabase,authentication,andcloudfunctions
      * **VisualStudioCode**:Sourcecodeeditingand debugging
      * **AndroidStudioEmulator**:Forinitialtestinganddebugging

###### TestingTools

* + - * ManualtestingacrossrealAndroiddevices
      * Firebasedebugtoolsandlogsforbackendbehavior monitoring
      * In-apploggingtotrackerrorsanddata flow

###### EvaluationMetrics

Thesystemwasevaluated basedonthefollowingcriteria:

* + - * **ResponseTime**:Timetakentoloadcontentandplacebids
      * **Real-TimePerformance**:Accuracyandspeedof real-timebidupdates
      * **UserExperience**: Easeofnavigation, UIresponsiveness
      * **ErrorHandling**: Howtheapp behavesin errorornetworkfailure scenarios

## Resultsand Discussion

###### FunctionalPerformance

Theapp successfullyperformed all corefunctions:

* + - * Userswereableto **sign up, login,andupdatetheir profile**.
      * Sellerscould**upload items**withimages andsetstartingbidprices.
      * Buyerswereableto**placebids**thatupdatedinstantlyacrossotherusers'screensusing Firebase real-time database.
      * The**chatsystem**workedwithoutdelays,enablingsmoothbuyer-seller communication.
      * **Notifications**triggeredcorrectlywhenanewbidwasplacedoramessagewas received.

###### Real-TimeBiddingEvaluation

Real-timeperformancewastestedwith **multipleusersbiddingsimultaneously**.The Firebase backend ensured:

* + - * Bidsupdatedalmost**instantly**(within0.5–1seconddelay).
      * Nodatalossoccurred duringsimultaneousbid placements.
      * Consistentsynchronizationacross devices.

###### User Feedback

Testusersfoundthe app:

* + - * Easyto use
      * Visuallyappealingandmodern
      * Convenientforbuyingandsellingitemsinanauctionformat Minor UI improvements were noted during testing, such as:
      * Addinga countdown timerforauction end
      * Improvingbutton placementsforsmall screens

###### PerformanceUnderLoad

* + - * Theapp handled**5–10concurrentusers**withoutanyslowdown or crash.
      * Firebase’sreal-timeservicesscaledautomaticallywithoutconfiguration changes.
      * Nomajorperformancebottleneckswereobserved duringevaluation.

**CHAPTER 6 CONCLUSIONANDDISCUSSION**

## StrengthsofThisProject

The**OnlineAuctionApp**successfullyachievesitsgoalofprovidingamodern,interactive platform for buying and selling through real-time bidding. Several strong points of this project include:

* **User-FriendlyInterface**:Theapphasaclean,intuitivedesignusingFlutter,making it easy to use for both buyers and sellers.
* **Real-TimeFunctionality**:Firebaseensuresthatbidding,chatting,andnotifications work in real time, offering a responsive and engaging user experience.
* **Cross-PlatformSupport**:DevelopedwithFlutter,theappcanrunonbothAndroid and iOS devices without the need for separate codebases.
* **SecureAuthentication**:FirebaseAuthenticationprovidessafeandreliableloginand registration for all users.
* **InteractiveFeatures**:In-appchatandpushnotificationsincreaseuserinteractionand improve communication between buyers and sellers.
* **Scalability**:Firebaseallowstheapptohandlemultipleusersandcanbescaledfor larger usage in the future.

Overall,theprojectshowcasesawell-balancedcombinationofdesign,functionality,and performance.

## LimitationsandFutureWork

Althoughtheprojectmeetsmostofitsgoals,therearesomelimitationsthatcouldbe addressed in future versions:

###### Limitations

* + - * **NoPaymentIntegration**:Currently,theappdoesnotincludeabuilt-inpayment system, so transactions must be handled externally.
      * **AdminDashboardMissing**:Thereisnoadminpaneltomanageusers,monitor activity, or handle complaints and reports.
      * **LimitedAuctionControls**:Sellerscannotcurrentlysetspecificauctiontimelimitsor extend auctions.
      * **NoRatingorReviewSystem**:Buyersandsellerscannotleavefeedback,which could help build trust and reputation.

###### FutureWork

Toimproveandexpand theproject,the following featurescanbeadded:

* + - * **SecurePaymentGatewayIntegration**(e.g.,Stripe, PayPal)
      * **AdminPanel**forsystem monitoringandmanagement
      * **Auction Timer**andbiddingcountdownforeach item
      * **ReviewandRatingSystem**forusers
      * **DarkMode**and accessibilityenhancements
      * **WebVersion** oftheplatformforwider reach

Theseenhancementswillmaketheappmoreprofessionalandincreaseitsvalueinareal- world market.

## Reasonsfor Failure–IfAny

Duringtheproject,therewerenocompletefailures.However, wedidface afew**challenges**:

* **TimeManagement**:Managingtimebetweendifferentmodules(UIdesign,backend, testing) was difficult, especially in early stages.
* **LearningCurve**:GainingfamiliaritywithFirebaseandreal-timedatabasestooktime and effort.
* **LimitedTestingDevices**:WewereonlyabletotestonafewAndroiddevices,so cross-device performance might vary slightly.

### REFERENCES

1. D.Gupta,HowtoBuildanOnlineAuctionApplication.Benefits,Features,Costs,2024.
2. G.Patil,OnlineAuctionAppDevelopment:ACompleteGuide,India(Delhi),2022.
3. J.Tiwary,AuctionAppDevelopment:UnlockingtheFutureofOnlineBidding,India (Mumbai), 30 October, 2023.
4. M.Butt,HowtoCreateanOnlineReal-TimeBidding/AuctionApp,Mumbai(bendra), 2022.
5. Ajay,HowtoDevelopanApplicationfor OnlineAuctions?Features,Advantages,and Costs, Chennai, 2024.

### APPENDICES

ListofAppendices

A0.CopyofProject Registration Form

A1a.ProjectProposalandVisionDocument

A1b.CopyofProposalEvaluationCommentsbyJury A2. Requirement Specifications

A3. Design Specifications A4.OtherTechnicalDetails Test cases

UI/UX Details CodingStandards Project Policy

A5.Flyer &PosterDesign

A6.CopyofEvaluationComments

CopyofEvaluationCommentsbySupervisorforProject –IMidSemesterEvaluation Copy of Evaluation Comments by Jury for Project – I End Semester Evaluation

CopyofEvaluationCommentsbySupervisorforProject –IIMidSemesterEvaluation Copy of Evaluation Comments by Jury for Project – II Mid Semester Evaluation

CopyofEvaluationCommentsbyJuryforProject – IIEndSemesterEvaluation A7. Meetings’ Minutes

A8.ResearchPaper A10. Any other

### A0.COPYOFPROJECTREGISTRATIONFORM

APhotostatorscannedcopyshouldbeplacedwhensubmittingadocumentto Project Coordinator.(**Note**: Please remove this line when attach copy that is required)

### A1A.PROJECTPROPOSALANDVISIONDOCUMENT

1. **Introduction:**

The Online Auction System is built with the primary goal of easing the process of conducting auctions for users, which raises the need for constructive non-standard approaches. This application has also enhanced the bidding mechanism, so instead of offering items at fixed prices, bidders can participate in an item’s bidding process and buy it for a more competitive price. Another enhancement is the implementation of a user profile management system that allows users to register and manage their profiles securely, an auction module building which encompasses the description of items and the conditions of the bids, and a time-dependent bidding module where users can track the history of the bids and update their status. The ability to communicate directly for price negotiations is facilitated via a chatting system, while the admin side allows for a streamlined management of the auctions, user accounts, and user disputes. Users are also instantly notified about the changes in the auction status and bids, in addition to the fact that the application is optimized for any mobile device which means that it is easy for users to use the application when they are busy doing other tasks.

* 1. **Problem Statement:**
  2. Current e-commerce platforms limit interactions with fixed pricing, reducing opportunitiesforbuyersandsellers.TheOnlineAuctionApptransformsthisby offering a dynamic, real-time bidding system, enhancing engagement and maximizing value for both parties
  3. **Project Milestones:**
  4. Here are the project milestones for the Online Auction App in simple terms:
  5. Initial Setup and Planning: Set up the development environment and outline the project’s goals and requirements.
  6. User Registration and Login: Build the basic functionality allowing users to create accounts, log in, and manage their profiles.
  7. Auction Listing Feature: Enable users to post items• for auction with details like start/end times and minimum bids.
  8. Bidding System: Implement real-time bidding,• where users can place and track bids on auction items.
  9. Chat System: Develop a feature allowing buyers and sellers to communicate directly through the app.
  10. Admin Panel Development: Create an admin interface for managing users, auctions, and handling disputes.
  11. Notification System: Add real-time alerts to inform users about bid updates and auction status changes.
  12. Testing and Quality Assurance: Test the app thoroughly to ensure it works well and fixes any issues found.
  13. Launch and Deployment: Release the app to users, ensuring it’s accessible and functioning as expected.
  14. Feedback and Improvements: Collect user feedback and make necessary updates to enhance the app’s functionality and user experience.

**Literature Review:**

The Online Auction App seeks to innovate the traditional auction process by leveraging modern technology to enhance user experience, security, and engagement. Below is a review of related literature, existing products, and services in the field.

* Literature Review :

Evolution of Online Auctions: Research has shown that online auction platforms, such as eBay, have significantly transformed traditional auctions by allowing global participation. Studies emphasize the importance of user experience and real-time bidding as key factors in user satisfaction.

User Engagement: Literature highlights the role of gamification and real-time notifications in keeping users engaged. Features like bidding updates and personalized messages can improve user retention and activity.

Security Concerns: Numerous studies have focused on the critical need for secure transactions and data protection in online platforms. Strong authentication, encryption, and fraud detection mechanisms are crucial for maintaining user trust.

* Product Review

eBay: One of the pioneers in online auctions, eBay offers a vast marketplace with auction and fixed-price options. Its strengths include a wide user base and advanced bidding features, but it can be complex for new users.

Amazon Auctions: This platform provides auction services but lacks the comprehensive auction-specific features found in dedicated auction platforms. Its integration with Amazon's vast marketplace is a significant advantage.

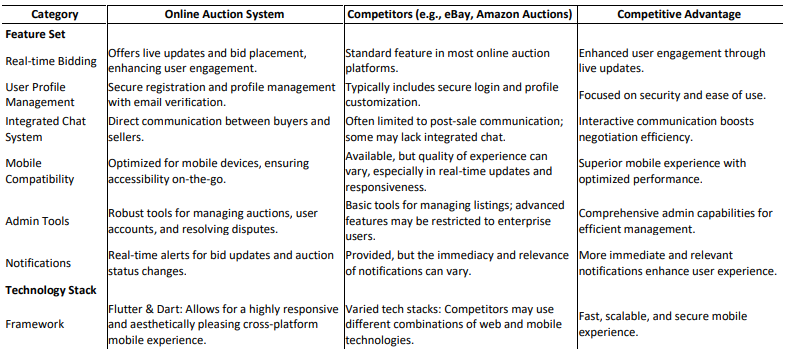
BidSpotter: Specializing in industrial auctions, BidSpotter offers live auction streaming and bidding. Its niche focus is its• strength, though it may not cater to general consumer needs.

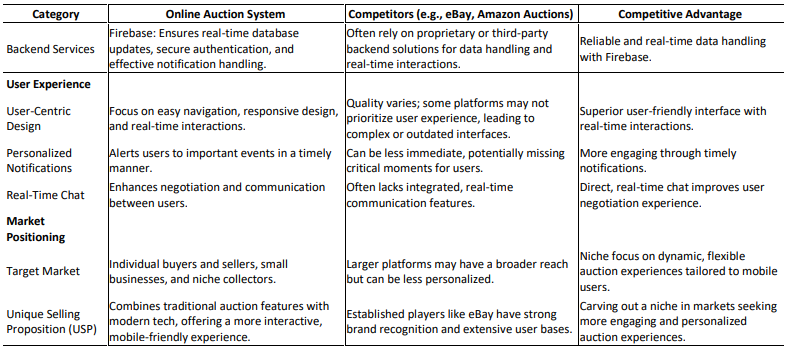
* Service Review Mobile Auction Apps:

Many platforms now offer mobile apps to facilitate on-the-go bidding. These apps are often praised for their convenience but can suffer from performance issues if not well-optimized.

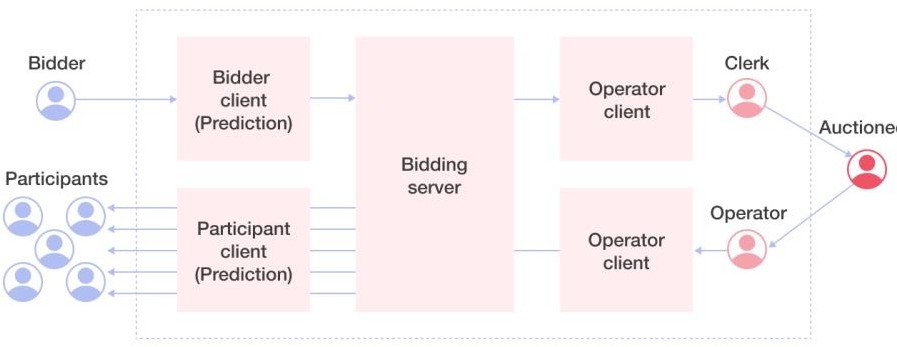
Real-time Notifications and Chat: Services that provide real-time updates and communication between buyers and sellers have been well-received. They enhance transparency and user satisfaction by facilitating direct negotiations and timely updates.

**Gap Analysis:**

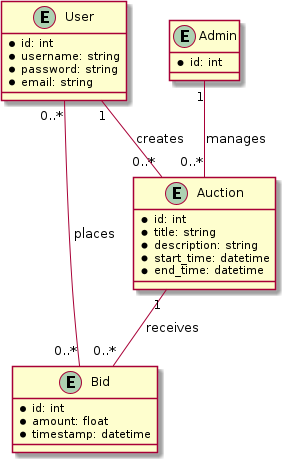
****

****

**Big Picture:**



**ER Diagram:**



**Algorithm and Model Findings:**

**Algorithm:**

1. Auction Lifecycle Management

* his alogrithm manages the creation, running, and conclusion of auctions.

**Steps:**

* Initialize auction with start and end times.
* Validate item details(name, category, starting price)
* Continuously monitor for incoming bids during the auction.
* Automatically close auctions after the end time and determine the winner.

1. Bid Validation Algorithm

* his ensures that bids are valid

**Steps:**

* Check if the auction is active.
* Compare the new bid to the current higher bid.
* If valid, update the current bid and notify participants.

1. Winner Selection Algorithm

* At the end of an auction, this determines the highest bidder as the winner.

**Steps:**

* Sort of all bids for the auction by bid amount in descending order.
* Select the highest bid.
* Notify the winner and seller.

**Model:**

The system can be module using the MVC pattern for clarity and scalability.

* **Model:**

Handles the business logic and data storage, including auctions, users and bids.

* **Veiw:**

User interfaces for bidders and sellers to interact with the system(e.g.,bidding pages, dashboards).

* **Controller:**

Coordinates user actions.

**Findings:**

Using above algorithm and model structure,the following findings can be observed:

* **User Engagement:**

Real-time updates on bids create a competitive environment, increasing user activity.

Notifications for outbids and auction results enhance user experience.

* **Efficiency:**

Bid validation ensures data integrity, preventing errors like lower bids being accepted. Automated auction management reduces manual intervention.

* **Scalability:**

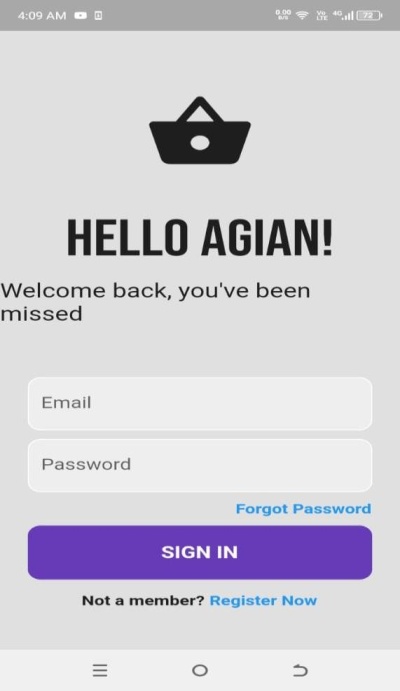
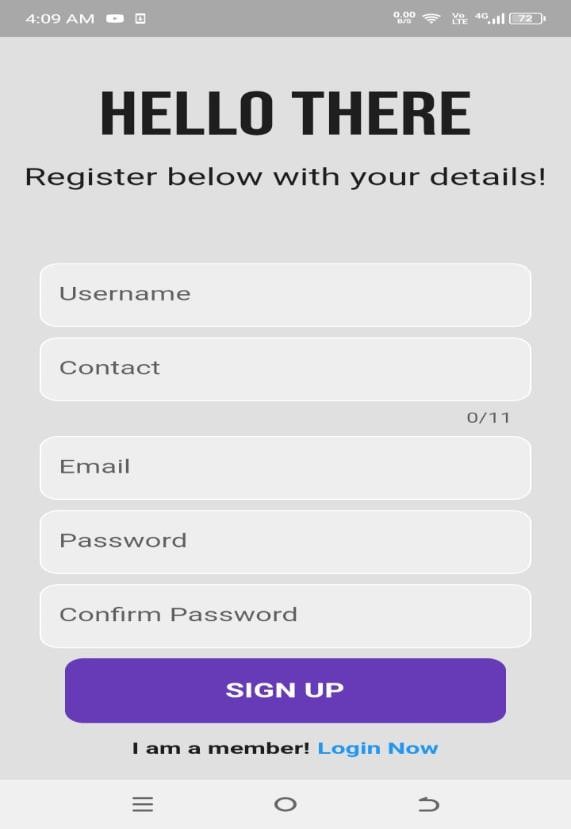
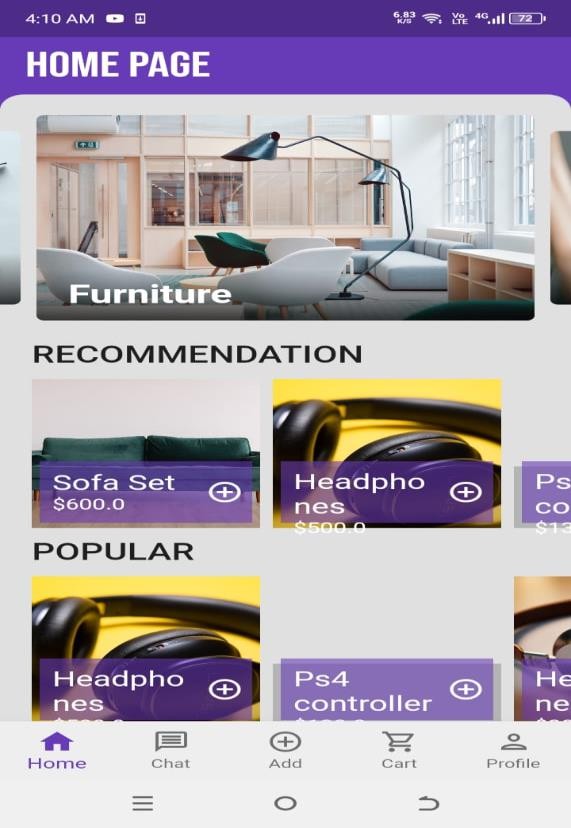
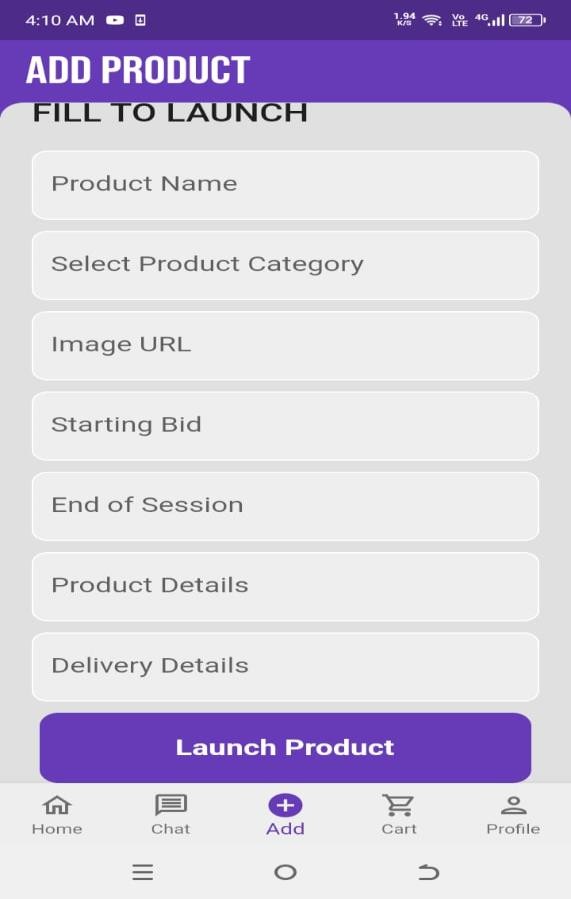
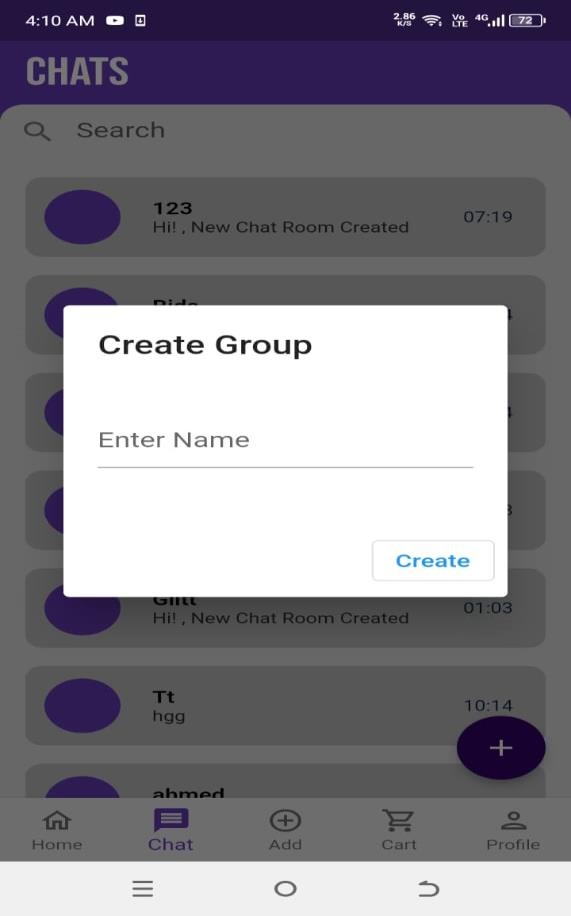
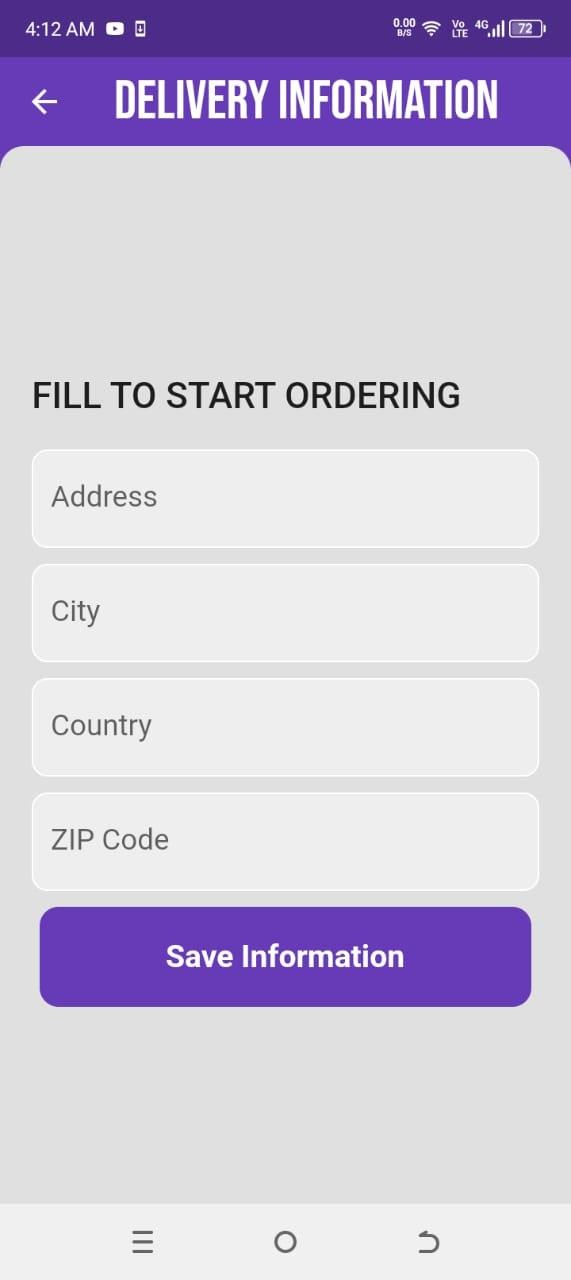
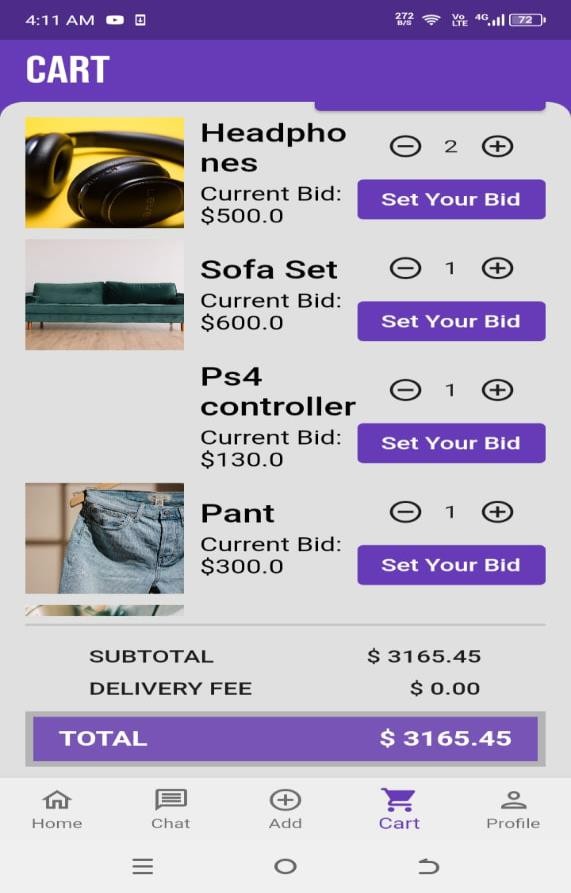
The modular design allows adding features like payment processing or user reviews without major changes to the system.

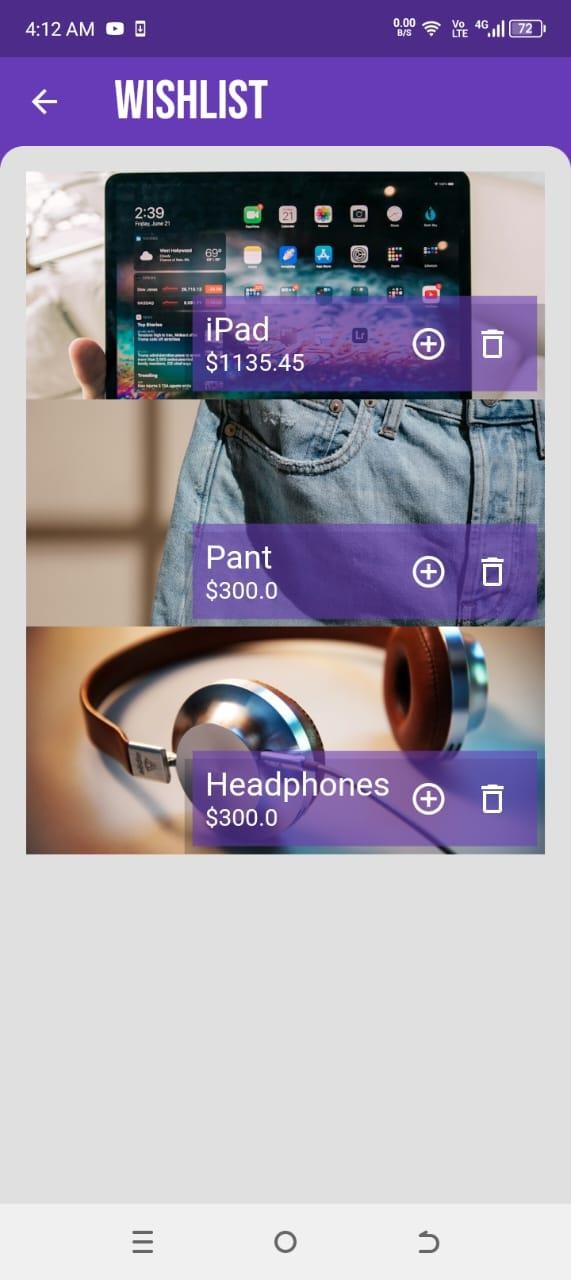
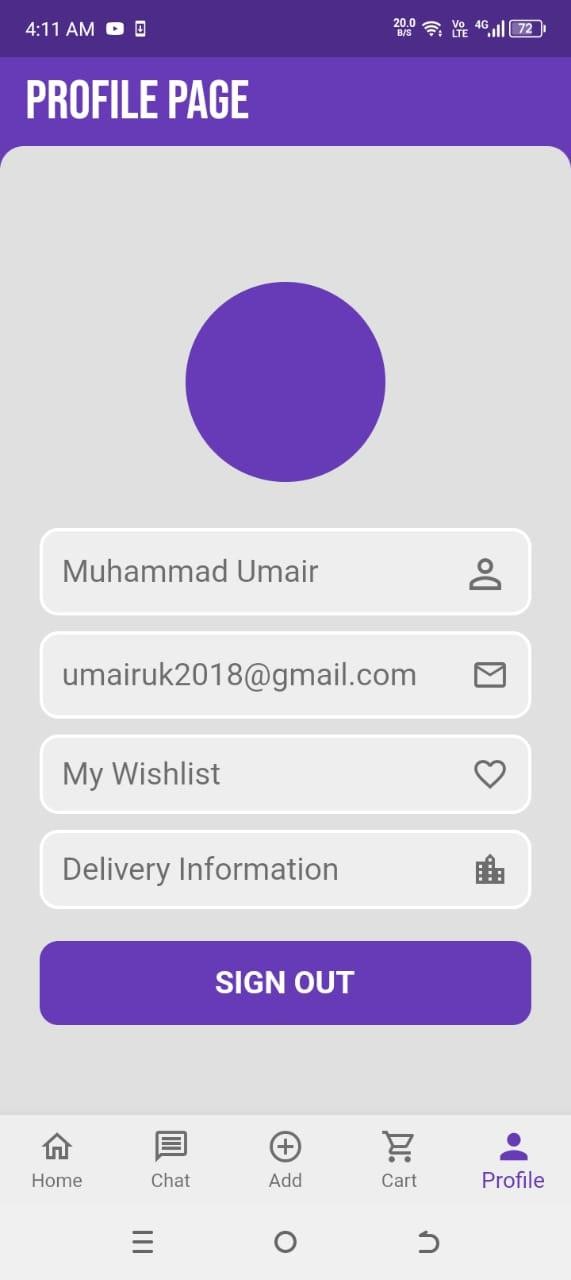
* **Potential Challenges:**

Handling high traffic during popular auctions may require server optimization

. Ensuring fair play and detecting fraudulent activities (e.g., fake bids).

**UI Designs:**



**References:**

* D. Gupta, How to Build an Online Auction Application.Benefits, Features, Costs, 2024.
* G. Patil, Online Auction App Development: A Complete Guide, India (Delhi), 2022.
* J. Tiwary, Auction App Development: Unlocking the Future of Online Bidding, India (Mumbai), 30 October, 2023.
* M. Butt, How to Create an Online Real-Time Bidding/Auction App, Mumbai (bendra), 2022.
* Ajay, How to Develop an Application for Online Auctions? Features, Advantages, and Costs, Chennai, 2024.

### A1B.COPYOFPROPOSALEVALUATIONCOMMENTSBY JURY

APhotostatorscannedcopyshouldbeplacedwhensubmittingadocumenttoProject Coordinator.(**Note**: Please remove this line when attach copy that is required)

### A2.REQUIREMENT SPECIFICATIONS

# Table of Contents

[**1.** Introduction 8](#_Toc4677804)

[1.1 Purpose of Document 8](#_Toc4677805)

[1.2 Intended Audience 8](#_Toc4677806)

[1.3 Abbreviations 8](#_Toc4677807)

[**2.** Overall System Description 9](#_Toc4677808)

[2.1 Project Background 9](#_Toc4677809)

[2.2 Problem Statement 9](#_Toc4677810)

[2.3 Project Scope 9](#_Toc4677811)

[2.4 Not In Scope 9](#_Toc4677812)

[2.5 Project Objectives 9](#_Toc4677813)

[2.6 Stakeholders & Affected Groups 9](#_Toc4677814)

[2.7 Operating Environment 9](#_Toc4677815)

[2.8 System Constraints 9](#_Toc4677816)

[2.9 Assumptions & Dependencies 9](#_Toc4677817)

[**3.** External Interface Requirements 10](#_Toc4677818)

[3.1 Hardware Interfaces 10](#_Toc4677819)

[3.2 Software Interfaces 10](#_Toc4677820)

[3.3 Communications Interfaces 10](#_Toc4677821)

[**4.** System Functions / Functional Requirements 11](#_Toc4677822)

[4.1 System Functions 11](#_Toc4677823)

[4.2 Use Cases 12](#_Toc4677824)

[4.2.1 List of Actors 12](#_Toc4677825)

[4.2.2 List of Use Cases 12](#_Toc4677826)

[4.2.3 Use Case Diagram 13](#_Toc4677827)

[4.2.4 Description of Use Cases 13](#_Toc4677828)

[**5.** Non - Functional Requirements 16](#_Toc4677829)

[5.1 Performance Requirements 16](#_Toc4677830)

[5.2 Safety Requirements 16](#_Toc4677831)

[5.3 Security Requirements 16](#_Toc4677832)

[5.4 Reliability Requirements 16](#_Toc4677833)

[5.5 Usability Requirements 16](#_Toc4677834)

[5.6 Supportability Requirements 16](#_Toc4677835)

[5.7 User Documentation 16](#_Toc4677836)

[**6.** References 17](#_Toc4677837)

# Introduction

The Online Auction System is built with the primary goal of easing the process of conducting auctions for users, which raises the need for constructive non-standard approaches. This application has also enhanced the bidding mechanism, so instead of offering items at fixed prices, bidders can participate in an item’s bidding process and buy it for a more competitive price. Another enhancement is the implementation of a user profile management system that allows users to register and manage their profiles securely, an auction module building which encompasses the description of items and the conditions of the bids, and a time-dependent bidding module where users can track the history of the bids and update their status. The ability to communicate directly for price negotiations is facilitated via a chatting system, while the admin side allows for a streamlined management of the auctions, user accounts, and user disputes. Users are also instantly notified about the changes in the auction status and bids, in addition to the fact that the application is optimized for any mobile device which means that it is easy for users to use the application when they are busy doing other tasks.

## Purpose of Document

The purpose of this document is to provide a detailed Software Requirements Specification (SRS) for the Online Auction App. This document outlines the system's functionality, performance, and operational requirements to ensure the successful implementation and deployment of the application.

## Intended Audience

## The Online Auction App is designed for a range of end-users, including: Individual sellers wanting to auction off their items – this could be anything from a car to a collectible Individuals looking to bid on items they are interested in Businesses seeking to expand their revenue stream by offering products for auction Businesses using the app to source products that they may want to integrate into their existing inventory. The app provides value for buyers who are looking for auction items, competitive deals, rare or hard-to-find products and the thrill of bidding. Just as sellers, whether they are small business owners, collectors, or individuals, can benefit from the app by accessing a broader market, garnering competitive bids that can increase their profit margin, and simplifying their selling process.

**1.3 Abbreviations**

**SRS:**  Software Requirements Specification

**FYP:** Final Year Project

**AI:** Artificial Intelligence

**SDK:** Software Development Kit

**IDE:** Integrated Development Environment

**UI:** User Interface

**UAT:** User Acceptance Testing

**CRUD** Create, Read, Update, Delete

**LAN:** Local Area Network

**IP:** Internet Protocol

**MAC:** Media Access Control

# Overall System Description

## Project Background

The online shopping platform is designed to overcome the limitations of traditional e-commerce platforms, which basically offer fixed pricing with no flexibility for customers to negotiate or bid on products role. This application provides a dynamic and interactive environment in which users can engage in real-time trading, giving sellers the opportunity to maximize profits and provide buyers with competitive deals.

To accomplish this, the app uses modern technologies such as Flutter and Dart to create a functional and efficient interface. Google Firebase acts as a database, ensuring data stability, real-time updates, and secure storage. The app has advanced features such as secure email authentication, real-time notifications of new announcements, and chat features, which empower buyers and sellers communicate directly, creating transparency and trust

The app is designed to meet both practical and non-functional needs. Functional requirements include things like store names, checkout options, and user profile management, while non-functional requirements focus on better performance, reliability, and advanced security measures, including password protection and including low latency.

During testing, the app takes several steps to ensure quality and compliance. Unit testing validates individual components, integration testing monitors the interactions between these components, and system testing examines all application functionality. Finally, user acceptance testing (UAT) ensures that the app is global personal expectations, making it user-friendly and effective.

The significance of this project lies in its ability to provide a platform for buyers and sellers to engage in business dynamically. Features such as product categorization, wishlist creation, and personalized messages make it a comprehensive solution for modern auction needs. The Online Auction App bridges the gap between traditional e-commerce and live auction systems, making it an innovative tool for enhancing online trade.

The importance of this project lies in its ability to offer a platform for customers and dealers to interact in enterprise dynamically. Features inclusive of product categorization, wishlist introduction, and customized messages make it a complete solution for current auction wishes. The Online Auction App bridges the distance among conventional e-trade and live auction systems, making it an progressive device for enhancing on line alternate.

## Problem Statement

The problem statement highlights the limitations of present e-trade apps, that specialize in their reliance on fixed pricing fashions. These apps do now not allow buyers to negotiate or bid on merchandise, creating a gap that the Online Auction System pursuits to cope with via introducing real-time bidding.

## Project Scope

The scope of the application describes the main features and functions of the Online Auction App:

* User management: Managing registration, access, and user information.
* Auction List: To include items for auction with information such as start/end times and minimum bids.
* Bidding : Enable real-time bidding and update users on the current bid status.
* Chat system: facilitates communication between buyers and sellers during negotiations.
* Chat bot: Assist users by providing automated responses and enhancing the overall user experience.
* Administration: Managing user accounts, selling and resolving disputes.
* Reports: Sends real-time updates on auctions and bids.
* Mobile Compatibility: Make sure the application is responsive and accessible on all devices.

## Not in Scope

While not explicitly stated, potential Not in Scope items might include:

* Integration with external e-commerce platforms.
* Handling physical delivery of auctioned items.
* Advanced AI-driven price prediction or bidding suggestions.
* Support for desktop platforms.

## Project Objectives

The objectives of the Online Auction System include:

* Providing a secure, user-friendly platform for online auctions.
* Empowering users to buy or sell goods dynamically through bidding.
* Enhancing business opportunities for sellers and offering competitive deals for buyers.
* Facilitating real-time interactions between users for negotiations.
* Ensuring seamless, responsive operation on mobile devices.

## Stakeholders& Affected Groups

Key stakeholders and affected groups include:

* Buyers: Individuals looking for competitive prices and unique items.
* Sellers: Individuals or businesses aiming to reach a broader audience and maximize profits.
* Administrators: Responsible for managing auctions, user disputes, and overall system integrity.
* Developers: Maintaining and updating the app's functionalities.
* Educational Institutions: Using the project as a model for future academic work.

## Operating Environment

The app operates on:

* Mobile Devices: Designed primarily for Android, with potential expansion to iOS.
* Firebase Database: Ensures real-time data updates, secure storage, and robust performance.
* Dart & Flutter Frameworks: Providing a cross-platform development environment optimized for mobile applications.

## System Constraints

* Performance Limitations: Excessive traffic may lead to lag or increased firebase costs.
* Database Dependency : Full reliance on firebase for real-time functionality and storage.
* Platform Restriction: Initially limited to mobile platforms, without desktop support.
* Financial Constraints: Limited resources for scalability and premium hosting services.

## Assumptions & Dependencies

Assumptions:

* Users will have access to stable internet connections for real-time bidding.
* Firebase services will remain affordable and reliable for project needs.
* Buyers and sellers will adhere to platform rules to minimize disputes.

Dependencies:

* Reliance on Firebase for backend operations.
* Compatibility with Android devices during initial implementation.

# External Interface Requirements

## Software Interfaces

The application interfaces with the following software:

* Firebase Backend Services:
  + Firebase Real time Database: For storing auction listings, user profiles, and bid updates.
  + Firebase Authentication: For user sign-in, sign-up, and email verification.
  + Firebase Messaging: For sending notifications to users regarding auctions and bids.
* External Owners: Firebase, owned by Google, manages the backend operations, including database hosting, authentication services, and notification delivery.
* Interface Details:
  + Firebase APIs are used to integrate these services, ensuring seamless communication between the app and the backend.

## Communications Interfaces

The app relies on the following communication interfaces:

* Network Communication:
  + Wi-Fi/Cellular Network: The app requires an active internet connection to interact with the Firebase services for real-time data sync.
  + REST APIs: Used to interact with Firebase for CRUD (Create, Read, Update, Delete) operations.
* Communication with Devices:
  + Push Notifications: Delivered through Firebase Cloud Messaging, ensuring users receive updates on auctions and bids.
* Local Area Network (LAN):
  + Although not explicitly stated, the app can be used over a secure local network if configured, but its primary reliance is on cloud-based communication.

# System Functions / Functional Requirements

## System Functions

### 1. ****User Functionalities****

#### 1.1 User Registration:

* Users can create an account by registering with their personal details, including name, email, password, and contact information.
* Email verification is required to activate the account.

#### 1.2 User Login:

* Registered users can log in using their email and password.
* A password reset feature is available for users who forget their credentials.

#### 1.3 Homepage Access:

* After successful login, users are directed to a personalized homepage displaying auction highlights, popular items, and ongoing bids.

#### 1.4 Add Bids:

* Users can view auction items and place bids.
* The system displays the current highest bid and allows users to input their own bid.
* Users receive notifications when they are outbid.

#### 1.5 View Items by Category:

* Users can browse items organized into categories and subcategories for easier navigation.

#### 1.6 Post an Auction:

* Users can post their own auction by providing item details, images, starting bid, and auction duration.
* The auction will be visible to all users once approved.

#### 1.7 Search Product Category:

* Users can search for products by category or keywords using a dynamic search bar.

#### 1.8 Chat System:

* A chat system enables users to communicate with one another in real-time.
* Public chat rooms and private messaging are supported.

#### 1.9 Send Personal Messages:

* Users can send direct messages to other users to discuss items or negotiate terms.
* Messages are accessible through the user’s inbox.

### 2. ****Admin Functionalities****

#### 2.1 Manage Users:

* Admin can view and manage user accounts.
* Admin can edit user details and track user activity.

#### 2.2 Manage Items:

* Admin can view, edit, or delete auction items as necessary.
* Items can be approved or rejected based on compliance with platform policies.

#### 2.3 Manage Categories:

* Admin can create, edit, and delete product categories and subcategories.
* Categories are structured to improve item discoverability.

#### 2.4 Block User:

* Admin can block users who violate platform rules.
* Blocked users cannot log in or participate in auctions.

#### 2.5 Administrative Functions:

* Admin have access to system analytics and reports, including user engagement, auction performance, and revenue statistics.
* Admin can configure system-wide settings, such as notifications and auction rules.

### 3. ****Additional Features****

#### 3.1 Email Notifications:

* Users and admin receive email notifications for important updates, such as bid status, auction deadlines, and account activities.

#### 3.2 Pop-Up Notifications:

* Real-time pop-up notifications alert users to outbids, auction status changes, or new messages.

#### 3.3 Delivery Information:

* Sellers can provide delivery details for auctioned items.
* Buyers can track the delivery status through the app.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Functions | Category | Attribute | Details & Boundary Constraints |
| R1.1 | Record the underway sale – the items purchased | Evident | System Response time | Price listing within 3 seconds  Availability agreement in less than 10 sec |
| R1.2 | Reduce inventory quantities when a sale is committed | Hidden | Concurrent user load | Supports up to 100 concurrent users without performance degradation. |
| … | Manage user profiles securely | Hidden | Security | Enforce password complexity and encryption.  - Multi-factor authentication. |
| R1.3 | Facilitate competitive bidding for items b | Evident | User Interaction | Real-time updates on the bidding status with sub-second latency. |
| R1.4 | Notify users about changes in auction status or bids | Evident | Real-Time Notification | Notification delivery within 2 seconds of change. |
| R1.5 | Track bid history and update bid status | Evident | Data Accuracy | Bid history is accurate and updated within 1 second of bid placement. |
| R1.6 | Enable direct communication for price negotiations | Evident | Communication response | Chat messages delivered within 1 second. |
| R1.8 | Streamline admin management of auctions and user disputes | Hidden | Admin Efficiency | Admin actions reflected within 5 seconds of operation. |
| R1.9 | Optimize application for mobile devices | Frill | Compatibility | Application responsive and operational on devices with screens as small as 4 inches. |

**5. System** **Attributes/ Nonfunctional Requirements**

#### 4.1 Performance Requirements

* The application will perform with minimal lag.
* Real time database ensures instant updates and synchronization.

#### 4.2 Reliability

* Ensure data consistency across all users and sessions.

#### 4.3 Security

* Password protection and encryption are implemented to safeguard user data.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Details and Boundary Constraints** | **Category** |
| Response Time | The system should complete important tasks, like placing bids or viewing updates, within 5 seconds. | Optional |
| Concurrent User Load | The app must handle at least 10 users at the same time without slowing down. | Mandatory |
| Data Availability | The system should be up and running 99.9% of the time to ensure users can access it whenever needed. | Mandatory |
| User Interface | The app should be easy to use and work well on different devices. | Optional |
| Security Measures | Use passwords, encryption, and secure communication to keep user data safe. | Mandatory |
| Expansion Flexibility | The system should allow easy addition of new features and handle more users as needed. | Optional |

## Use Cases

### List of Use Cases

* Enable Location:

The user grants location permissions to the system for the location-based features.

* User Registration:

The user creates an account by entering their details and initiating the registration process.

* Email Verification:

After registration, the system sends a verification email, which the user must validate to activate their account.

* Login:

The user provides credentials (email and password) to authenticate and access their account.

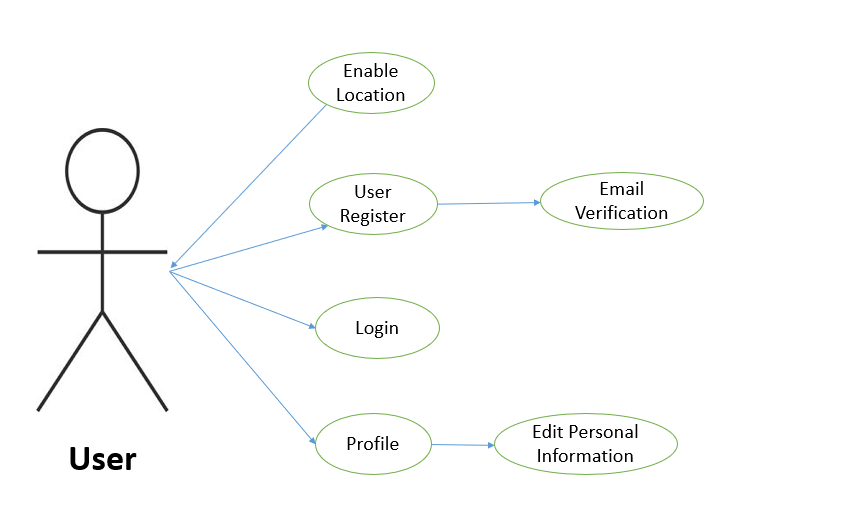
* Profile:

The user views their account details, including personal information and other settings.

* Edit Personal Information:

The user updates their account details, such as name, contact information, or email.

### Use Case Diagram



### Description of Use Cases

**1. Enable User**

|  |  |
| --- | --- |
| Name | Enable User |
| Actors | User |
| Purpose | To allow the user to enable location services to enhance the functionality of application. |
| Description | The user can enable location services to receive location specific features, such as displaying nearby auction items or arranging local deliveries. |
| Cross References | None |
| Pre-Conditions | The user must have a registered account and be logged in. |
| Successful Post-Conditions | The location services are enabled, and the application can use the user’s location to provide relevant features. |
| Failure Post-Conditions | Location services remain disabled, and the user cannot access location specific features. |

**2. User Register**

|  |  |
| --- | --- |
| Name | User register |
| Actors | User |
| Purpose | To allow a new user to create an account in the system. |
| Description | The user enters personal information, such as name, email, and password, to register on the platform. |
| Cross References | Email verification |
| Pre-Conditions | The user must not have an existing account. |
| Successful Post-Conditions | A new account is created, and the user receives an email for verification. |
| Failure Post-Conditions | The account is not created, and the user must try registered again. |

**3. Email Verification**

|  |  |
| --- | --- |
| Name | Email verification |
| Actors | User |
| Purpose | To confirm the user’s email address to activate the account. |
| Description | After registration, the user receives a verification email. Clicking the link in the email confirms the user’s email address. |
| Cross References | User Register |
| Pre-Conditions | The user must have completed the registration process. |
| Successful Post-Conditions | The user’s email is verified, and the account is activated. |
| Failure Post-Conditions | The email remains unverified, and the user cannot fully access the platform. |

**4. Login**

|  |  |
| --- | --- |
| Name | Login |
| Actors | User |
| Purpose | To allow the user to access their account. |
| Description | The user provides their email and password to log into their account. |
| Cross References | None |
| Pre-Conditions | The user must have a registered and verified account. |
| Successful Post-Conditions | The user is logged in and can access their account. |
| Failure Post-Conditions | The user remains logged out, and they must retry logging in. |

**5. Profile**

|  |  |
| --- | --- |
| Name | Profile |
| Actors | User |
| Purpose | To allow the user to view and manage their personal information. |
| Description | The user can access their profile to see personal details, such as name, email, and account settings. |
| Cross References | Edit Personal Information |
| Pre-Conditions | The user must be logged in. |
| Successful Post-Conditions | The user can view their profile details. |
| Failure Post-Conditions | The user cannot view their profile, and they must retry accessing it. |

**6. Edit Personal Information**

|  |  |
| --- | --- |
| Name | Edit Personal Information |
| Actors | User |
| Purpose | To allow the user to update their personal details. |
| Description | The user can edit personal information, such as their name, email, or password, through their profile. |
| Cross References | Profile |
| Pre-Conditions | The user can edit personal information, such as their name, email, or password, through their profile. |
| Successful Post-Conditions | The user’s personal information is updated successfully. |
| Failure Post-Conditions | The changes are not saved, and the user must retry editing their information. |

**4.2.5 Typical Course Events**

| **Actor Action** | **System Response** |
| --- | --- |
| **User registers for a new account** | The system validates the input, sends an email verification, and upon successful validation, creates a new user account. |
|  |  |
| **User logs in with credentials** | The system verifies the credentials and grants access to the user’s dashboard if the credentials are correct. |
|  |  |
| **User views available auctions** | The system retrieves and displays a list of active auctions categorized by type. |
| **User places a bid on an item** | The system checks if the bid is higher than the current highest bid, updates the bid status, and notifies the user of the bid status. |
|  |  |
| **User posts a new auction** | The system validates the auction details, posts the auction in the relevant category, and notifies the user of a successful post. |
|  |  |
| **Admin manages user accounts** | The system allows the admin to view, edit, block, or delete user accounts based on administrative actions. |
|  |  |
| **User sends a message in the chat system** | The system delivers the message to the intended recipient and notifies both parties of new messages. |
|  |  |
| **User receives a notification on bid update** | The system sends a real-time notification to the user regarding updates on bids they are interested in. |
| **Admin manages auction listings** | The system enables the admin to edit, approve, or remove auction listings and resolves any disputes arising from auctions. |
|  |  |
| **User logs out** | The system logs the user out, ensuring session termination and data security. |

**References**

|  |
| --- |
| * D. Gupta, How to Build an Online Auction Application Benefits, Features, Costs, 2024. |
| * G. Patil, Online Auction App Development: A Complete Guide, India (Delhi), 2022. |
| * J. Tiwary, Auction App Development: Unlocking the Future of Online Bidding, India (Mumbai), 30 October, 2023. |
| * M. Butt, How to Create an Online Real-Time Bidding/Auction App, Mumbai (bendra), 2022. |
| * Ajay, How to Develop an Application for Online Auctions Features, Advantages, and Costs, Chennai, 2024. |

### A3.DESIGNSPECIFICATIONS

**Definition of Terms, Acronyms, and Abbreviations**

| **Term** | **Description** |
| --- | --- |
| **Online Auction** | A digital platform where users can bid on items, with the highest bidder winning the item at the close. |
| **Auction Engine** | The core system managing auction functionalities such as item listings, bidding processes, and timers. |
| **User Roles** |  |
| Bidder | Individuals participating in auctions by placing bids. |
| Seller/Auctioneer | Users listing items for auction. |
| Admin | System managers handling user management, disputes, and auction oversight. |
| **Timed Auction** | An auction type where bids are accepted for a fixed duration. |
| **Reserve Auction** | An auction type where the item is sold only if bidding meets a preset reserve price. |
| **"Buy Now" Option** | A feature allowing buyers to purchase items at a fixed price without bidding. |
| **Real-Time Bidding** | A system enabling instantaneous bid placement and updates, ensuring seamless interactions. |
| **Chat Module** | An integrated feature allowing direct communication between buyers and sellers for negotiations. |
| **User Authentication** | A secure login mechanism, including encrypted passwords and potentially two-factor authentication. |
| **Notification System** | Alerts users about bid status, auction closures, and system updates in real-time. |
| **Data Encryption** | A method of safeguarding user data and transactions against unauthorized access. |
| **Caching Mechanisms** | Technology like Redis used to speed up data retrieval for active auctions and frequent queries. |
| **Fraud Detection** | Systems and algorithms to identify and mitigate fraudulent activities, such as fake listings or shill bidding. |
| **COD (Cash on Delivery)** | A payment option where buyers pay upon receiving the product, often integrated with logistics systems. |
| **Payment Gateway** | Third-party services facilitating secure online transactions, including COD and escrow payments. |
| **Accessibility Standards** | Compliance with guidelines like WCAG to ensure the app is usable by individuals with disabilities. |
| **Regulatory Compliance** | Adherence to laws and regulations governing online auctions, including data protection and e-commerce. |
| **Scalability** | The app's ability to handle increased traffic and user growth without performance issues. |
| **Dispute Resolution** | Mechanisms to address and resolve conflicts between buyers and sellers effectively. |

# Table of Contents

[1. Introduction 8](#_Toc4677804)

[1.1 Purpose of Document 8](#_Toc4677805)

[1.2 Intended Audience 8](#_Toc4677806)

[1.3 Abbreviations 8](#_Toc4677807)

[2. Overall System Description 9](#_Toc4677808)

[2.1 Project Background 9](#_Toc4677809)

[2.2 Problem Statement 9](#_Toc4677810)

[2.3 Project Scope 9](#_Toc4677811)

[2.4 Not In Scope 9](#_Toc4677812)

[2.5 Project Objectives 9](#_Toc4677813)

[2.6 Stakeholders & Affected Groups 9](#_Toc4677814)

[2.7 Operating Environment 9](#_Toc4677815)

[2.8 System Constraints 9](#_Toc4677816)

[2.9 Assumptions & Dependencies 9](#_Toc4677817)

[3. External Interface Requirements 10](#_Toc4677818)

[3.1 Hardware Interfaces 10](#_Toc4677819)

[3.2 Software Interfaces 10](#_Toc4677820)

[3.3 Communications Interfaces 10](#_Toc4677821)

[4. System Functions / Functional Requirements 11](#_Toc4677822)

[4.1 System Functions 11](#_Toc4677823)

[4.2 Use Cases 12](#_Toc4677824)

[4.2.1 List of Actors 12](#_Toc4677825)

[4.2.2 List of Use Cases 12](#_Toc4677826)

[4.2.3 Use Case Diagram 13](#_Toc4677827)

[4.2.4 Description of Use Cases 13](#_Toc4677828)

[5. Non - Functional Requirements 16](#_Toc4677829)

[5.1 Performance Requirements 16](#_Toc4677830)

[5.2 Safety Requirements 16](#_Toc4677831)

[5.3 Security Requirements 16](#_Toc4677832)

[5.4 Reliability Requirements 16](#_Toc4677833)

[5.5 Usability Requirements 16](#_Toc4677834)

[5.6 Supportability Requirements 16](#_Toc4677835)

[5.7 User Documentation 16](#_Toc4677836)

[6. References 17](#_Toc4677837)

# Introduction

The Online Auction System is built with the primary goal of easing the process of conducting auctions for users, which raises the need for constructive non-standard approaches. This application has also enhanced the bidding mechanism, so instead of offering items at fixed prices, bidders can participate in an item’s bidding process and buy it for a more competitive price. Another enhancement is the implementation of a user profile management system that allows users to register and manage their profiles securely, an auction module building which encompasses the description of items and the conditions of the bids, and a time-dependent bidding module where users can track the history of the bids and update their status. The ability to communicate directly for price negotiations is facilitated via a chatting system, while the admin side allows for a streamlined management of the auctions, user accounts, and user disputes. Users are also instantly notified about the changes in the auction status and bids, in addition to the fact that the application is optimized for any mobile device which means that it is easy for users to use the application when they are busy doing other tasks.

## Purpose of Document

The purpose of this document is to provide comprehensive Software Design Specifications (SDS) for the Online Auction App. Outlining the design methodology, architecture and detailed design considerations. This document is intended to guide the development team and serve as a reference for stakeholders.

## Intended Audience

This document is intended for

* Development Team members
* Project managers
* System architectures
* QA testers
* Stakeholders and supervisors

## Document Convention

The font: Arial

Font size: 12

## Project Overview

The Online Auction App is a digital platform designed to make buying and selling items easier and more dynamic. Unlike regular e-commerce apps with fixed prices, this app lets users bid on products, giving buyers a chance to get better deals and sellers an opportunity to earn more through competitive bidding.

It includes features like real-time updates on bids, user profiles to manage activities, and a chat system for direct communication between buyers and sellers. The app is secure, ensuring user data and transactions are safe, and works smoothly on mobile devices so users can participate in auctions anytime, anywhere. Administrators help manage the system, handle disputes, and ensure everything runs smoothly.

Overall, this app bridges the gap between traditional e-commerce and live auction systems, providing a modern and efficient solution for online trading. Whether you're a buyer looking for bargains or a seller wanting to reach a wider audience, this app offers everything you need for a seamless auction experience

## Scope

* The Online Auction App is designed to create a modern, interactive platform for buying and selling items through real-time bidding. It allows users to register, create profiles, and post items for auction while providing a secure and user-friendly environment. Buyers can browse auctions, place bids, and receive instant updates about their bid status. Sellers benefit from reaching a wider audience and maximizing profits through competitive bidding.
* The app includes built-in chat features, enabling buyers and sellers to communicate directly, making negotiations and transactions smoother. It is optimized for mobile devices, ensuring that users can participate in auctions anytime and anywhere. Additionally, the app incorporates strong security measures to protect user data and transactions, such as encrypted communication and secure login processes. Powered by Firebase, the app ensures real-time updates and scalability, allowing it to handle a growing number of users and auctions seamlessly.
* The scope of this project focuses on delivering a reliable, accessible, and engaging online auction experience, bridging the gaps in traditional e-commerce platforms and empowering both buyers and sellers with innovative tools and features.

.

# Design Considerations

Here’s a tailored version of the text for your project on an **online auction app**:

### 4.1.1 Problem Identification and Preliminary Considerations

This section outlines the critical issues and requirements that need to be addressed before creating a comprehensive design solution for the online auction app. These considerations serve as the foundation for a robust, user-centric system design.

#### **User Accessibility and Experience**

* Ensure the app is intuitive and user-friendly for both buyers and sellers.
* Provide seamless navigation, from account creation to bidding and selling processes.
* Optimize the app for different devices (mobile, tablet, and desktop) and platforms (iOS, Android, and web browsers).

#### **Security and Trust**

* Implement secure authentication (e.g., two-factor authentication).
* Safeguard user data and financial transactions through encryption and compliance with data protection laws.
* Include features like verified user badges, feedback systems, and dispute resolution mechanisms to build trust.

#### **Bidding and Auction Processes**

* Support multiple auction types, such as timed auctions, reserve auctions, and "Buy Now" options.
* Real-time bidding updates with notifications for outbids and auction endings.
* Prevent fraudulent practices, such as shill bidding or fake listings, with automated monitoring.

#### **Payment and Transaction Management**

* Provide an option for sellers to enable or disable COD for their listings based on their preferences.
* Clearly indicate to buyers during checkout whether COD is available for their selected items.
* Specify delivery regions or conditions where COD is applicable.

#### **Scalability and Performance**

* Design the system to handle a growing user base and large volumes of concurrent bids without performance degradation.
* Employ efficient database management for real-time data updates.
* Optimize load balancing and caching mechanisms to reduce latency.

#### **Regulatory and Compliance Requirements**

* Adhere to regional and international e-commerce and auction laws.
* Ensure taxation rules are applied correctly based on user locations.
* Address legal requirements for selling restricted or regulated items.

#### **Customer Support and Feedback**

* Provide multiple channels for customer support (chat, email, FAQs, etc.).
* Enable users to report issues directly through the app.
* Regularly collect user feedback to drive continuous improvement.

#### **Marketing and Growth**

* Include features for promotional offers, such as discounted listing fees or special bidding events.
* Integrate social media sharing to increase user engagement and app visibility.
* Use analytics to track user behavior and improve marketing strategies.

## Assumptions and Dependencies

### 4.2.1 Design-Specific Assumptions and Dependencies

#### **Dynamic Real-Time Updates**

* **Assumption:** The app must support real-time updates for auction events (e.g., bidding activity, auction expiration).
* **Dependency:** The design relies on WebSocket technology or similar real-time communication protocols to ensure low-latency updates. The system must scale to handle concurrent users during high-traffic auctions.

#### **Scalable Modular Architecture**

* **Assumption:** The system will need to support modular features like different auction types (e.g., timed, reserve) and payment methods, including COD and escrow services.
* **Dependency:** The design assumes a microservices architecture, where components such as the bidding engine, payment processing, and notifications are independently scalable.

#### **Customizable User Interfaces**

* **Assumption:** The app will offer different user roles (buyers, sellers, and admins), each with tailored dashboards and workflows.
* **Dependency:** The design depends on frameworks (e.g., React, Angular) that allow responsive, role-specific, and customizable UI elements.

#### **Geographical and Regional Constraints**

* **Assumption:** The system must accommodate region-specific features, such as language, currency, and tax rules.
* **Dependency:** The design incorporates APIs for localization, exchange rates, and tax calculations. Additionally, the system must dynamically adapt based on the user's location.

#### **Secure Payment and Delivery Integration**

* **Assumption:** For payments, COD will require integration with logistics providers to track deliveries and confirm transactions securely.
* **Dependency:** The design assumes APIs for delivery tracking, payment reconciliation, and secure data transfer between the app and third-party services.

#### **Fraud and Dispute Management**

* **Assumption:** Users will expect mechanisms to detect fraudulent activities and resolve disputes efficiently.
* **Dependency:** The design assumes the integration of AI-driven fraud detection tools and a scalable dispute resolution module for automated and manual review.

#### **Flexibility in Feature Addition**

* **Assumption:** New features (e.g., social sharing, in-app chat, or AI recommendations) may need to be added post-launch without disrupting existing workflows.
* **Dependency:** The system must employ a flexible plugin or API-based structure to allow feature expansion with minimal downtime.

#### **Third-Party Services**

* **Assumption:** Critical features like notifications, analytics, and performance monitoring will rely on third-party services (e.g., Firebase, Mixpanel, Twilio).
* **Dependency:** The design assumes stable APIs and continued support from these services, with fallback mechanisms in place for potential service interruptions.

#### **Compliance with Accessibility Standards**

* **Assumption:** The app must comply with accessibility guidelines to cater to users with disabilities.
* **Dependency:** The design depends on accessible design principles (e.g., WCAG) and tools for testing accessibility compliance during the development process.

#### **Support for High Availability and Disaster Recovery**

* **Assumption:** The app should remain operational during high traffic and recover quickly from unexpected downtimes.
* **Dependency:** The design assumes the use of cloud services with built-in redundancy, auto-scaling, and disaster recovery mechanisms.

## Risks and Volatile Areas

#### **Emerging User Requirements**

* **Source of Change:** Users may demand additional features like advanced filtering, new auction types, or expanded payment options (e.g., cryptocurrency, BNPL).
* **Impact:** Requires iterative development and system modularity to integrate new features without disrupting core functionalities.
* **Mitigation:**
  + Design the system with a modular architecture to allow independent addition of new features.
  + Use Agile development methodologies to respond to changing user requirements in short cycles.

#### **Technological Advancements**

* **Source of Change:** Adoption of new technologies (e.g., AI for fraud detection, AR for virtual item previews, blockchain for transparent bidding).
* **Impact:** Upgrades or overhauls in existing systems may be necessary to stay competitive.
* **Mitigation:**
  + Leverage APIs and cloud-based services that allow easy integration with cutting-edge technologies.
  + Regularly review technological trends to proactively plan for updates.

#### **Regulatory and Compliance Changes**

* **Source of Change:** Updates to e-commerce, auction, or data protection laws across regions.
* **Impact:** Non-compliance could lead to legal penalties or service restrictions in specific regions.
* **Mitigation:**
  + Maintain a legal and compliance team to monitor regulatory updates.
  + Design the system to support region-specific configurations for tax laws, restricted items, and data handling.

#### **Scalability Demands**

* **Source of Change:** A sudden increase in users or high-profile auctions leading to system overload.
* **Impact:** Performance degradation, slow bidding updates, or system crashes could harm user trust.
* **Mitigation:**
  + Utilize cloud infrastructure with auto-scaling capabilities to handle peak loads.
  + Employ load testing during development to ensure system stability under high traffic.

#### **Fraud and Security Threats**

* **Source of Change:** Increasing sophistication in fraud techniques (e.g., fake accounts, shill bidding, item misrepresentation).
* **Impact:** Loss of user trust, financial losses, and reputational damage.
* **Mitigation:**
  + Integrate AI-driven fraud detection and monitoring tools.
  + Regularly update security protocols and conduct penetration testing.

#### **Competition and Market Dynamics**

* **Source of Change:** Competitors introducing innovative features or offering lower transaction fees.
* **Impact:** Risk of user attrition and reduced market share.
* **Mitigation:**
  + Monitor competitors and user feedback to stay ahead in innovation.
  + Implement loyalty programs and promotional campaigns to retain users.

#### **Logistics and Payment Ecosystem Changes**

* **Source of Change:** Dependency on third-party logistics providers and payment gateways that may revise APIs or terms.
* **Impact:** Disruptions in payment processing or COD fulfillment.
* **Mitigation:**
  + Partner with multiple logistics providers and payment gateways to reduce single points of failure.
  + Periodically update integrations to align with third-party changes.

#### **Contingency Paths for Changes**

* Maintain a flexible microservices architecture that allows isolated updates without affecting the entire system.
* Use feature toggles to enable or disable new features in response to unexpected challenges.
* Develop a rollback plan for major updates to revert to stable versions if unforeseen issues arise.

# System Architecture

This section provides a high-level overview of the **online auction app**, describing how its functionality and responsibilities are divided among various components and subsystems. The architecture ensures scalability, maintainability, and efficient collaboration between components to deliver a seamless user experience.

#### **High-Level Overview**

The system follows a **modular and micro services-based architecture**, ensuring that each component operates independently while interacting with others through defined interfaces. It is divided into the following layers:

1. **Presentation Layer**
   * **Purpose:** User interaction via mobile and web applications.
   * **Components:**
     + Mobile App (iOS/Android)
     + Web Interface (React/Angular-based)
   * **Responsibilities:**
     + Provide an intuitive UI for buyers, sellers, and admins.
     + Handle real-time updates for bids, notifications, and auction statuses.
     + Ensure responsiveness and accessibility across devices.
2. **Business Logic Layer**
   * **Purpose:** Manage core functionalities and workflows of the auction system.
   * **Components:**
     + **Auction Engine:** Manages auction creation, bidding processes, and timer functionalities.
     + **User Management Service:** Handles registration, login, roles, and profile management.
     + **Payment Module:** Manages payment methods, including COD, escrow, and online payments.
     + **Notification Service:** Sends real-time alerts (e.g., outbids, auction end) via email, SMS, and push notifications.
   * **Responsibilities:**
     + Ensure real-time bid updates.
     + Validate bids, manage auction rules, and resolve conflicts.
     + Integrate fraud detection algorithms.
3. **Data Layer**
   * **Purpose:** Manage storage and retrieval of system data.
   * **Components:**
     + Relational Database (e.g., PostgreSQL): For structured data like user profiles, auction details, and transaction records.
     + NoSQL Database (e.g., MongoDB): For unstructured data like logs, notifications, and analytics.
     + Cache Layer (e.g., Redis): For high-speed access to frequently queried data, like active auctions and bids.
   * **Responsibilities:**
     + Maintain data consistency and integrity.
     + Provide APIs for seamless data retrieval and updates.
4. **Integration Layer**
   * **Purpose:** Facilitate communication with third-party services.
   * **Components:**
     + **Payment Gateways:** Support secure transactions for COD and online payments.
     + **Logistics APIs:** Handle shipping, tracking, and COD verification.
     + **Analytics and Monitoring Tools:** Collect and analyze user activity and system performance.
   * **Responsibilities:**
     + Manage external dependencies like payment and delivery services.
     + Ensure API reliability and fallback mechanisms for third-party failures.
5. **Infrastructure Layer**
   * **Purpose:** Provide a robust environment for system deployment and scalability.
   * **Components:**
     + Cloud Hosting (e.g., AWS, Azure, GCP): For deployment, scalability, and disaster recovery.
     + Load Balancers: Distribute traffic across servers to avoid overload.
     + Monitoring Tools (e.g., Prometheus, Grafana): Track system performance and uptime.
   * **Responsibilities:**
     + Enable auto-scaling to handle peak traffic during high-profile auctions.
     + Provide disaster recovery and backups to ensure data safety.

#### **Interaction Between Components**

1. **User Interaction Flow:**
   * Users interact with the app via the Presentation Layer.
   * User requests (e.g., placing a bid, viewing auctions) are processed by the Business Logic Layer.
   * Data retrieval and updates occur in the Data Layer, ensuring minimal latency through caching.
   * Notifications are sent via the Notification Service for key events (e.g., auction updates, bid success).
2. **Auction Process Flow:**
   * Sellers create auctions via the UI, processed by the Auction Engine.
   * Buyers place bids in real-time, validated and updated in the system through the Business Logic Layer.
   * Auction results are finalized, and payment/fulfillment processes are initiated using Payment and Logistics APIs.
3. **COD Workflow:**
   * Buyers select COD at checkout, triggering the Logistics API for order confirmation.
   * Delivery updates are tracked, and cash is reconciled via logistics partners before releasing funds to the seller.

#### 5.3 **Scalability and Extensibility**

* The architecture supports future feature additions (e.g., new auction types, advanced analytics) via modular components.
* High scalability is achieved through containerized microservices (e.g., using Docker and Kubernetes).
* APIs ensure seamless integration with new third-party services as the system evolves.
* System Level ArchitectureThe system-level architecture for the Online Auction System is designed to provide a modular, maintainable, and scalable structure. It decomposes the system into subsystems and components, detailing their roles, relationships, and interfaces.

#### **System Decomposition into Elements**

1. **User Interface Layer**:
   * Provides the interface for users (bidders, auctioneers, and admins) to interact with the system.
   * Includes web and mobile applications, ensuring accessibility across devices.
2. **Application Logic Layer**:
   * Responsible for implementing the business logic and core functionalities of the system, such as bid validation, auction management, and notifications.
   * Contains modules like:
     + **User Management**: Handles user authentication and profile management.
     + **Auction Module**: Manages the creation, display, and updating of auctions.
     + **Bid Module**: Processes bids and maintains bid histories.
     + **Notification System**: Sends real-time notifications to users.
3. **Data Access Layer**:
   * Interacts with the database to store and retrieve data related to users, auctions, bids, and notifications.
4. **Database Layer**:
   * Stores all persistent data, including user profiles, auction details, bids, chat messages, and system logs.
5. **External Systems**:
   * Payment Gateway: Integrates with external services for secure payment processing (optional, if included in the app).
   * Notification Services: Interfaces with external APIs for sending emails or push notifications.

## Software Architecture

The software architecture for the Online Auction System is designed with a layered approach for simplicity and modularity.

#### **Layers of Architecture**

1. **User Interface Layer**
   * Manages interactions with the users (bidders, auctioneers, admins).
   * Example technologies: React.js, Angular, or mobile frameworks like Flutter.
2. **Middle Tier (Business Logic)**
   * Handles the main business logic.
   * Manages bid validation, auction rules, and real-time updates.
   * Technologies: Node.js, Java Spring Boot, or .NET Core.
3. **Data Access Layer**
   * Interfaces between the application and the database.
   * Implements ORM frameworks like Hibernate or Sequelize.
4. **Database Layer**
   * Stores all application data persistently.
   * Firebase

# Design Strategy

The **design strategy** for the online auction app is centered around scalability, modularity, and user-centric functionality. This section outlines the key abstractions, mechanisms, and decision-making processes that shape the system's high-level organization while addressing considerations such as extensibility, reuse, interface paradigms, data management, and concurrency.

#### Future System Extension or Enhancement

**Strategy:**

* Use a **modular architecture** where each component (e.g., bidding engine, payment gateway, notification system) operates independently.
* Adopt **microservices** for core functionalities to enable easy addition or replacement of features.
* Ensure **API-driven development**, allowing integration with new third-party services without disrupting the system.

**Reasoning:**

* User needs and market trends evolve, necessitating a flexible system that supports new auction types (e.g., reverse auctions) or payment methods (e.g., crypto currencies).
* The micro services approach ensures isolated updates and prevents downtime.

**Trade-offs:**

* Increased complexity in managing inter-service communication and dependencies.
* Potential latency in distributed systems compared to monolithic architectures.

#### System Reuse

**Strategy:**

* Design reusable components for common functionalities like user authentication, payment processing, and notifications.
* Employ shared libraries and frameworks for tasks such as logging, error handling, and security.

**Reasoning:**

* Reuse reduces development effort for future projects or additional platforms (e.g., mobile and web).
* Standardized components improve system consistency and reduce maintenance overhead.

**Trade-offs:**

* Additional effort is required upfront to ensure components are generic and well-documented.
* Customization for specific use cases may require extending the reusable components.

#### User Interface Paradigms

**Strategy:**

* Use a **responsive design** paradigm to ensure a seamless experience across devices.
* Implement **role-based interfaces** tailored to buyers, sellers, and administrators.
* Focus on **real-time interactivity** with Web Socket or Server-Sent Events (SSE) for updates during auctions.

**Reasoning:**

* The app serves diverse user roles and device types, making adaptable UIs essential.
* Real-time feedback during bidding increases user engagement and ensures competitive fairness.

**Trade-offs:**

* Real-time UI updates require robust backend systems and increased network demands.
* Implementing role-specific interfaces can increase design complexity.

#### Data Management

**Strategy:**

* Use a combination of **relational databases** (e.g., Postgre SQL) for structured data (e.g., user profiles, auction details) and **No SQL databases** (e.g., Mongo DB) for unstructured data (e.g., logs, analytics).
* Employ **caching mechanisms** (e.g., Redis) for frequently accessed data like active auctions and bid histories.
* Design the system to support **data partitioning and replication** for high availability.

**Reasoning:**

* A hybrid database approach optimizes for performance and scalability.
* Caching reduces latency and ensures real-time data availability during critical auction moments.

**Trade-offs:**

* Managing hybrid databases introduces additional complexity.
* Consistency issues may arise between cached data and the primary database, requiring robust synchronization mechanisms.

#### Concurrency and Synchronization

**Strategy:**

* Use **optimistic locking** or **versioning** for handling concurrent updates in critical operations like placing bids.
* Employ **event-driven architecture** with message queues (e.g., Rabbit MQ, Kafka) to manage high-concurrency scenarios.
* Design backend services to be **stateless**, enabling horizontal scaling for increased concurrent user loads.

**Reasoning:**

* Auctions involve real-time interactions, making concurrency management crucial to avoid race conditions or data corruption.
* Event-driven systems decouple components, improving scalability and reliability under heavy loads.

**Trade-offs:**

* Increased complexity in managing distributed transactions.
* Real-time bid synchronization requires significant infrastructure investment.

# Detailed System Design

## Design Class Diagram

## Database Design

**7.2.1 Data 1: User**

| **Name** | User |
| --- | --- |
| **Alias** | Bidder, Auctioneer |
| **Content Description** | ID + Username + Password + Email + Role + Created\_At |
|  |  |

**Table: User**

| **Column Name** | **Description** |  | **Type** | **Length** | **Nullable** | **Default Value** | **Key Type** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Unique identifier for user |  | Integer | - | No | Auto-increment | PK |
| Username | Name of the user |  | String | 100 | No | NULL |  |
| Password | Encrypted password |  | String | 255 | No | NULL |  |
| Email | User’s email address |  | String | 150 | No | NULL |  |
| Role | Role of the user (bidder/seller) |  | String | 50 | No | 'Bidder' |  |
| Created\_At | Account creation timestamp |  | DateTime | - | No | CURRENT\_TIMESTAMP |  |

**7.2.2Data 2: Admin**

| **Name** | **Admin** |
| --- | --- |
| **Alias** | System Manager |
| **Where-used/how-used** | Used for managing users, auctions, and system disputes. |
| **Content Description** | ID + Username + Password + Email + Created\_At |

**Table: Admin**

| **Column Name** | **Description** | |  | **Type** |  | **Length** | **Nullable** | **Default Value** | **Key Type** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Unique identifier for admin |  | | Integer |  | - | No | Auto-increment | PK |
| Username | Name of the admin |  | | String |  | 100 | No | NULL |  |
| Password | Encrypted password |  | | String |  | 255 | No | NULL |  |
| Email | Admin’s email address |  | | String |  | 150 | No | NULL |  |
| Created\_At | Account creation timestamp |  | | DateTime |  | - | No | CURRENT\_TIMESTAMP |  |

**7.2.3 Data 3: Auction**

| **Name** | **Auction** |
| --- | --- |
| **Alias** | Item, Listing |
| **Where-used/how-used** | Used for creating auctions, placing bids, and notifications. |
| **Content Description** | ID + Title + Description + Start\_Time + End\_Time + Starting\_Bid + Current\_High\_Bid + User\_ID + Created\_At |

**Table: Auction**

| **Column Name** | **Description** | **Type** | **Length** | **Nullable** | **Default Value** |  | | **Key Type** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Unique identifier for auction | Integer | - | No | Auto-increment |  | | PK |
| Title | Title of the auction item | String | 200 | No | NULL |  | |  |
| Description | Description of the auction item | Text | - | Yes | NULL |  | |  |
| Start\_Time | Auction start time | DateTime | - | No | NULL |  | |  |
| End\_Time | Auction end time | DateTime | - | No | NULL |  | |  |
| Starting\_Bid | Minimum starting bid | Float | - | No | NULL | |  |  |
| Current\_High\_Bid | Current highest bid | Float | - | Yes | NULL | |  |  |
| User\_ID | ID of user creating the auction | Integer | - | No | NULL | |  | FK (User.ID) |
| Created\_At | Auction creation timestamp | DateTime | - | No | CURRENT\_TIMESTAMP | |  |  |

**7.2.4Data 4: Bid**

| **Name** | **Bid** |
| --- | --- |
| **Alias** | Offer |
| **Where-used/how-used** | Used for placing and tracking bids in auctions. |
| **Content Description** | ID + Amount + Timestamp + User\_ID + Auction\_ID |
|  |  |
|  |  |

**Table: Bid**

| **Column Name** | **Description** | **Type** | **Length** | **Nullable** | **Default Value** |  | **Key Type** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Unique identifier for the bid | Integer | - | No | Auto-increment |  | PK |
| Amount | Bid amount placed by the user | Float | - | No | NULL |  |  |
| Timestamp | Time the bid was placed | DateTime | - | No | CURRENT\_TIMESTAMP |  |  |
| User\_ID | ID of user placing the bid | Integer | - | No | NULL |  | FK (User.ID) |
| Auction\_ID | ID of auction the bid belongs to | Integer | - | No | NULL |  | FK (Auction.ID) |

**Data 5: Notification**

| **Name** | **Notification** |
| --- | --- |
| **Alias** | Alert |
| **Where-used/how-used** | Used to notify users about auction and bid updates. |
| **Content Description** | ID + Message + User\_ID + Timestamp |

**Table: Notification**

| **Column Name** | **Description** | **Type** | **Length** | **Nullable** | **Default Value** | **Key Type** |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Unique identifier for notification | Integer | - | No | Auto-increment | PK |
| Message | Notification content | Text | - | No | NULL |  |
| User\_ID | ID of user receiving the notification | Integer | - | No | NULL | FK (User.ID) |
| Timestamp | Time the notification was sent | DateTime | - | No | CURRENT\_TIMESTAMP |  |

## ER DIAGRAM

### 

**7.2.1.2 ER Data Model**

### **1. User Table**

| **Attribute Name** | **Data Type** | **Constraints** |
| --- | --- | --- |
| ID | Integer | Primary Key, Not null, Unique |
| Username | String | Not null |
| Password | String | Not null |
| Email | String | Not null, Unique |

### **2. Admin Table**

| **Attribute Name** | **Data Type** | **Constraints** |
| --- | --- | --- |
| ID | Integer | Primary Key, Not null, Unique |

### **Auction Table**

| **Attribute Name** | **Data Type** | **Constraints** |
| --- | --- | --- |
| ID | Integer | Primary Key, Not null, Unique |
| Title | String | Not null |
| Description | String | Optional |
| Start\_Time | DateTime | Not null |
| End\_Time | DateTime | Not null |

### **Bid Table**

| **Attribute Name** | **Data Type** | **Constraints** |
| --- | --- | --- |
| ID | Integer | Primary Key, Not null, Unique |
| Amount | Float | Not null |
| Timestamp | DateTime | Not null |
| User\_ID | Integer | Foreign Key references User(ID) |
| Auction\_ID | Integer | Foreign Key references Auction(ID) |

### **5 Relationships**

| **Relationship** | **Source Table** | **Source Attribute** | **Target Table** | **Target Attribute** | **Cardinality** |
| --- | --- | --- | --- | --- | --- |
| **User places Bid** | User | ID | Bid | User\_ID | 1 to Many |
| **Bid is placed on Auction** | Bid | Auction\_ID | Auction | ID | Many to 1 |
| **Admin creates Auction** | Admin | ID | Auction | Admin\_ID | 1 to Many |
| **Admin manages Auction** | Admin | ID | Auction | Admin\_ID | Many to Many\* |

**7.2.1.3 E/R Model Description**

1. **Entities:**
   * **User**: Represents the users participating in the auction system.
   * **Admin**: Represents the administrators managing the auctions.
   * **Auction**: Represents the auctioned items.
   * **Bid**: Represents the bids placed by users on auctions.
2. **Relationships:**
   * **User places Bid**: A user can place multiple bids, but each bid is associated with one user.
   * **Bid is placed on Auction**: A bid is linked to one auction, and an auction can have multiple bids.
   * **Admin creates Auction**: Admins are responsible for creating auctions.
   * **Admin manages Auction**: Admins also manage the auctions after they are created.
3. **Cardinalities:**
   * A **User** can place **0 or more Bids**, but a **Bid** is placed by **1 User**.
   * A **Bid** belongs to **1 Auction**, but an **Auction** can receive **0 or more Bids**.
   * An **Admin** can create or manage **0 or more Auctions**.

### 7.2.2 Data Dictionary

**7.2.2.1 Data 1: User**

**Attribute Name: User ID**

Description: Unique identifier for each user.

Data Type: Integer

Primary Key: Yes

Constraints: Not null, Unique

Attribute Name: User Name

Description: Name of the user.

Data Type: String

Constraints: Not null

Attribute Name: Email

Description: Email address of the user.

Data Type: String

Constraints: Not null, Unique

Attribute Name: Password

Description: Encrypted password for the user.

Data Type: String

Constraints: Not null

Attribute Name: Contact Number

Description: Phone number of the user.

Data Type: String

Constraints: Optional

**7.2.2.2 Data 2: Item**

**Attribute Name: Item ID**

Description: Unique identifier for each auction item.

Data Type: Integer

Primary Key: Yes

Constraints: Not null, Unique

Attribute Name: Item Name

Description: Name of the auction item.

Data Type: String

Constraints: Not null

Attribute Name: Item Description

Description: Detailed description of the auction item.

Data Type: Text

Constraints: Optional

Attribute Name: Starting Bid

Description: Minimum bid amount for the item.

Data Type: Decimal

Constraints: Not null

Attribute Name: Auction End Time

Description: End time of the auction for the item.

Data Type: Date Time

Constraints: Not null

**7.2.2.3 Data 3: Bid**

**Attribute Name: Bid ID**

Description: Unique identifier for each bid.

Data Type: Integer

Primary Key: Yes

Constraints: Not null, Unique

Attribute Name: Bid Amount

Description: Amount placed by the bidder.

Data Type: Decimal

Constraints: Not null

Attribute Name: Bid Time

Description: Time when the bid was placed.

Data Type: Date Time

Constraints: Not null

Attribute Name: User ID

Description: Identifier of the user who placed the bid.

Data Type: Integer

Constraints: Foreign Key (references User ID in User table)

Attribute Name: Item ID

Description: Identifier of the item being bid on.

Data Type: Integer

Constraints: Foreign Key (references Item ID in Item table)

**7.2.2.4 Data 4: Chat**

**Attribute Name: Chat ID**

Description: Unique identifier for each chat instance.

Data Type: Integer

Primary Key: Yes

Constraints: Not null, Unique

Attribute Name: Sender ID

Description: Identifier of the user sending the message.

Data Type: Integer

Constraints: Foreign Key (references User ID in User table)

Attribute Name: Receiver ID

Description: Identifier of the user receiving the message.

Data Type: Integer

Constraints: Foreign Key (references User ID in User table)

Attribute Name: Message Content

Description: Content of the chat message.

Data Type: Text

Constraints: Not null

Attribute Name: Timestamp

Description: Time when the message was sent.

Data Type: Date Time

Constraints: Not null

**7.2.2.5 Data 5: Notification**

**Attribute Name: Notification ID**

Description: Unique identifier for each notification.

Data Type: Integer

Primary Key: Yes

Constraints: Not null, Unique

Attribute Name: Notification Type

Description: Type of notification (e.g., Bid Update, Auction Status).

Data Type: String

Constraints: Not null

Attribute Name: User ID

Description: Identifier of the user receiving the notification.

Data Type: Integer

Constraints: Foreign Key (references User ID in User table)

Attribute Name: Message Content

Description: Details of the notification.

Data Type: Text

Constraints: Optional

Attribute Name: Timestamp

Description: Time when the notification was sent.

Data Type: Date Time

Constraints: Not null

## Application Design

### 7.4.1 Sequence Diagram

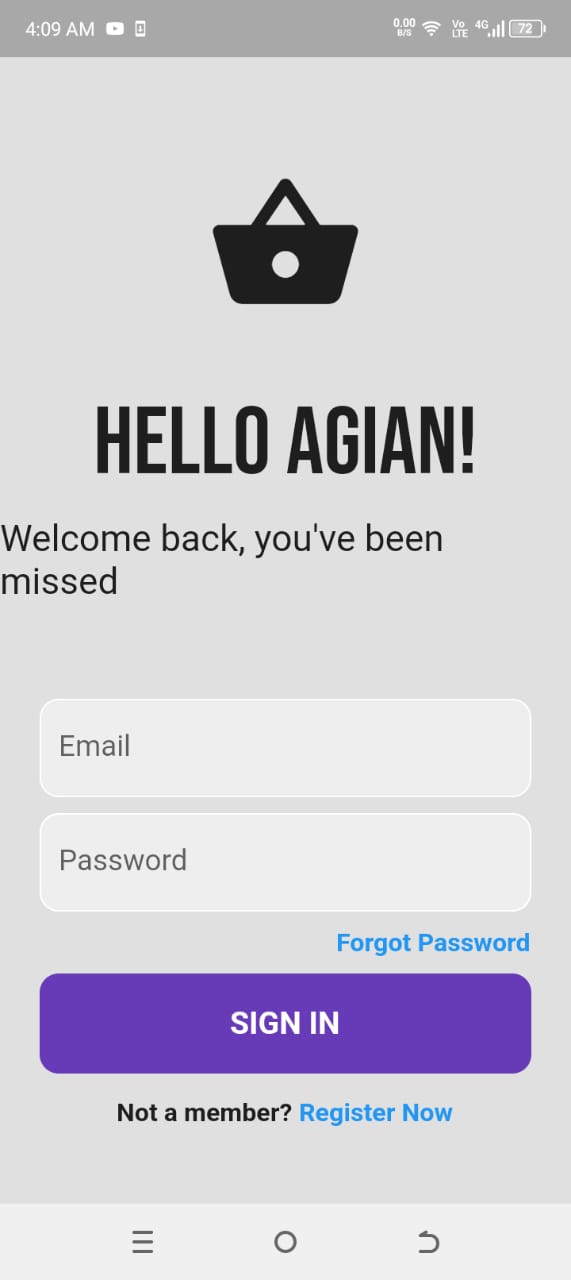
#### Sequence Diagram 1

### State Diagram

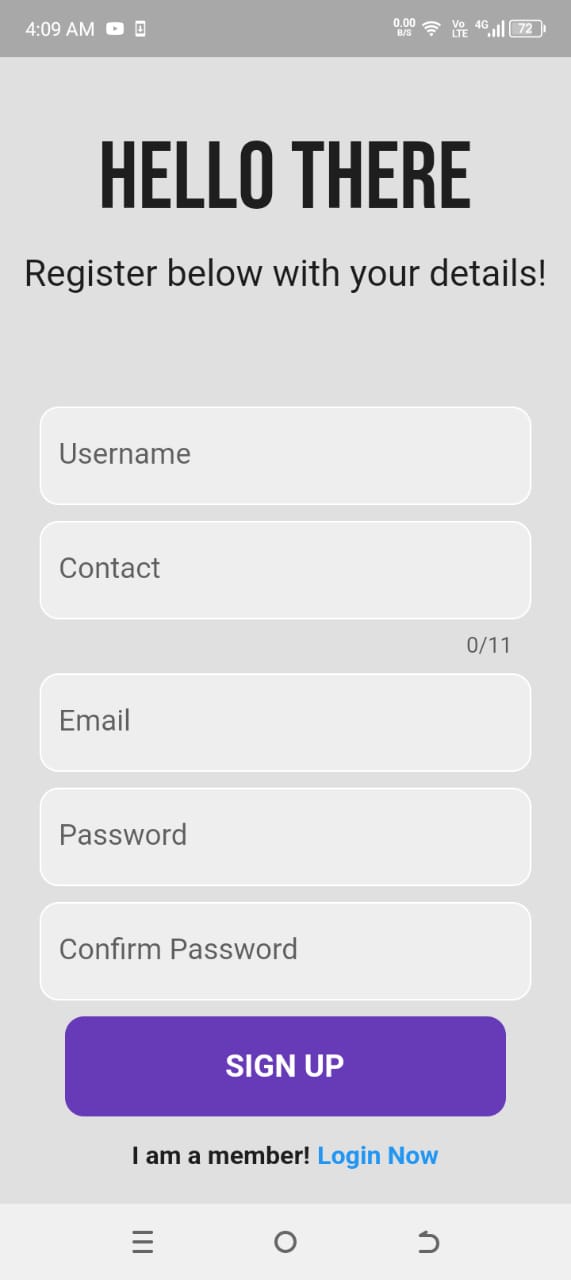
#### State Diagram 1

## GUI Design

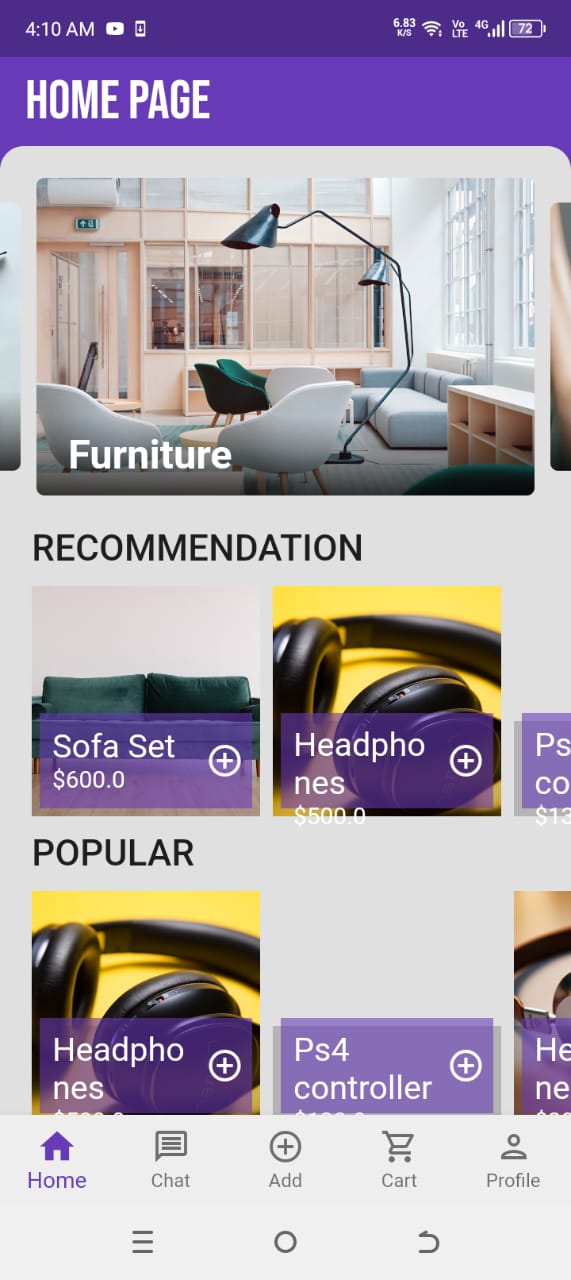
**7.5.1 Login Page - Mock Screen 1**



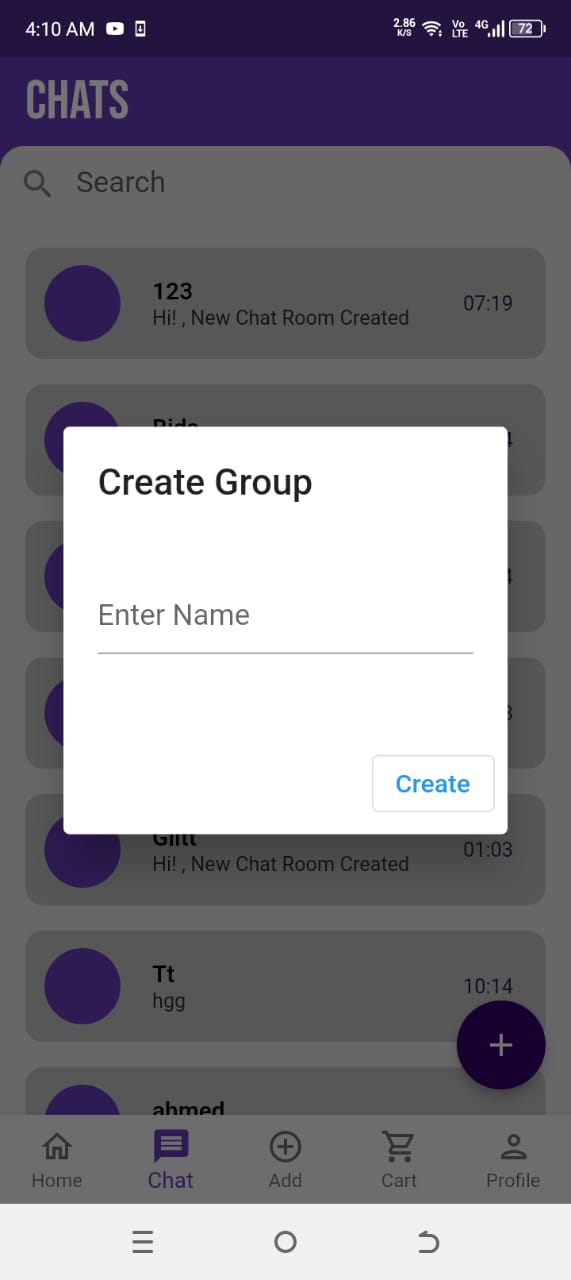
**7.5.2 Sign Up Page - Mock Screen 2**



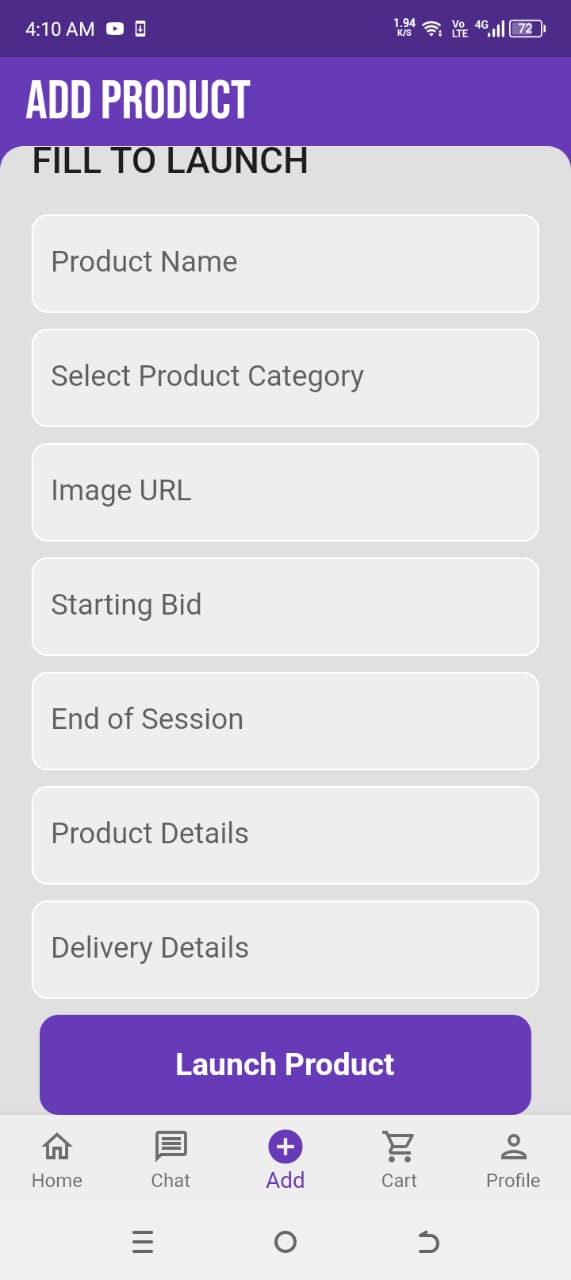
**7.5.3 HomePage - Mock Screen 3**



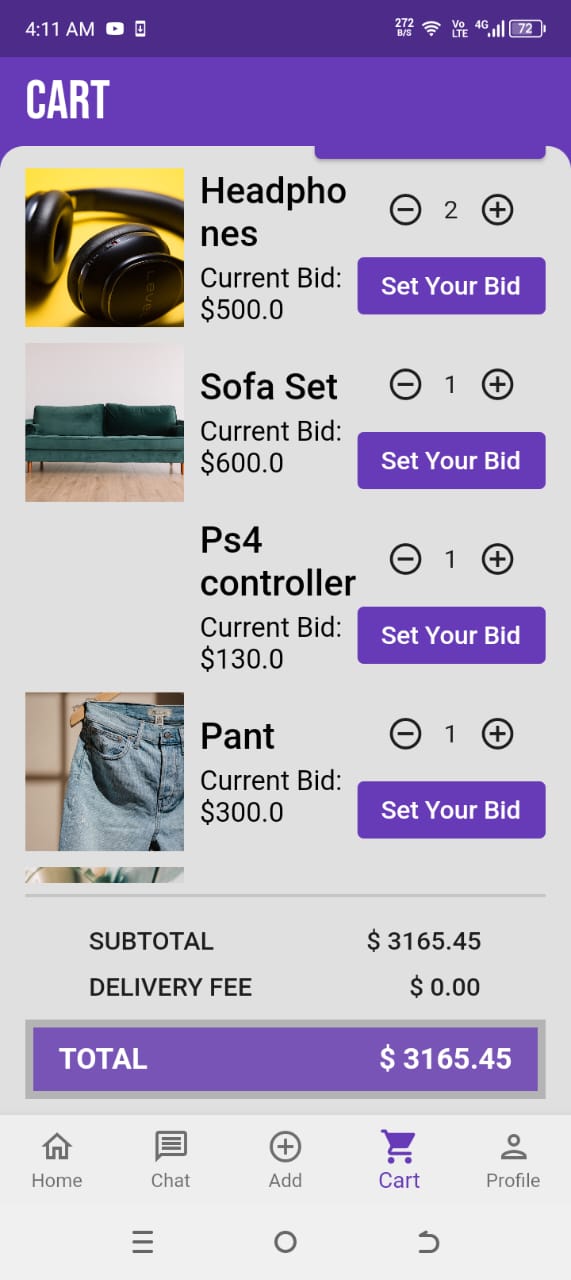
**7.5.4 Chat Room - Mock Screen 4**



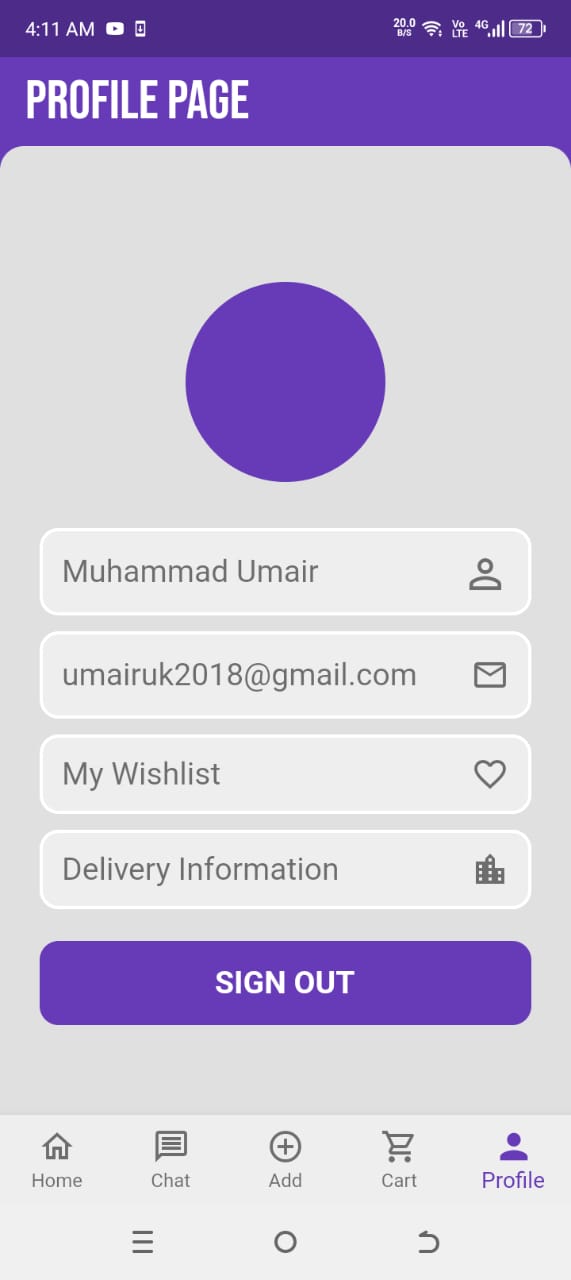
**7.5.5 Launch Product - Mock Screen 5**



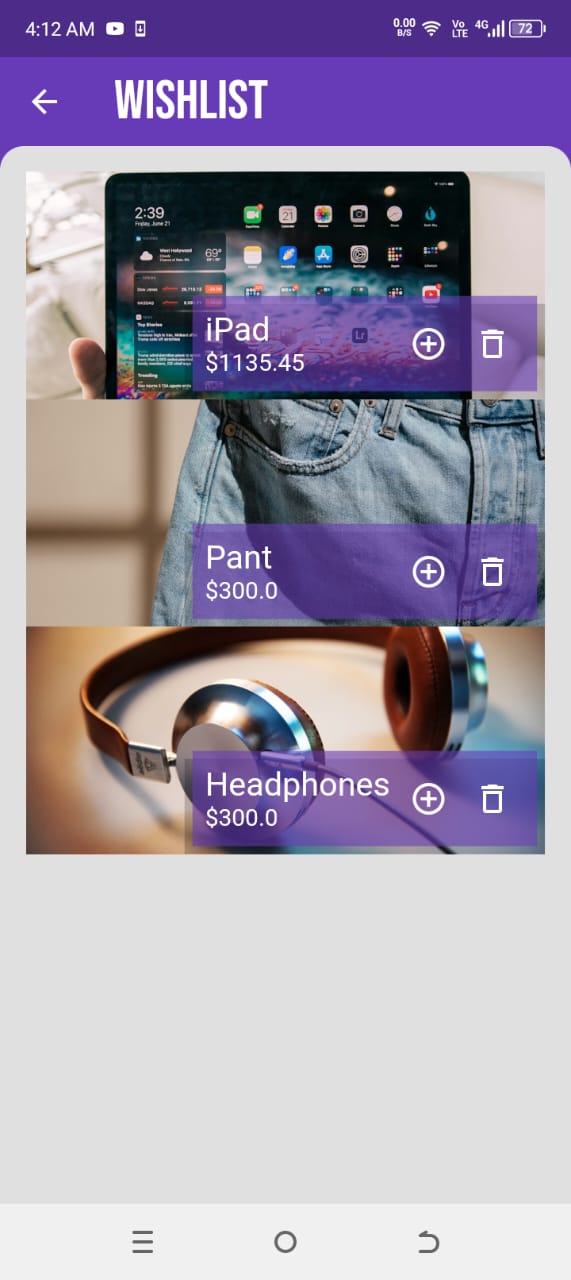
**7.5.6 Add To Cart - Mock Screen 6**



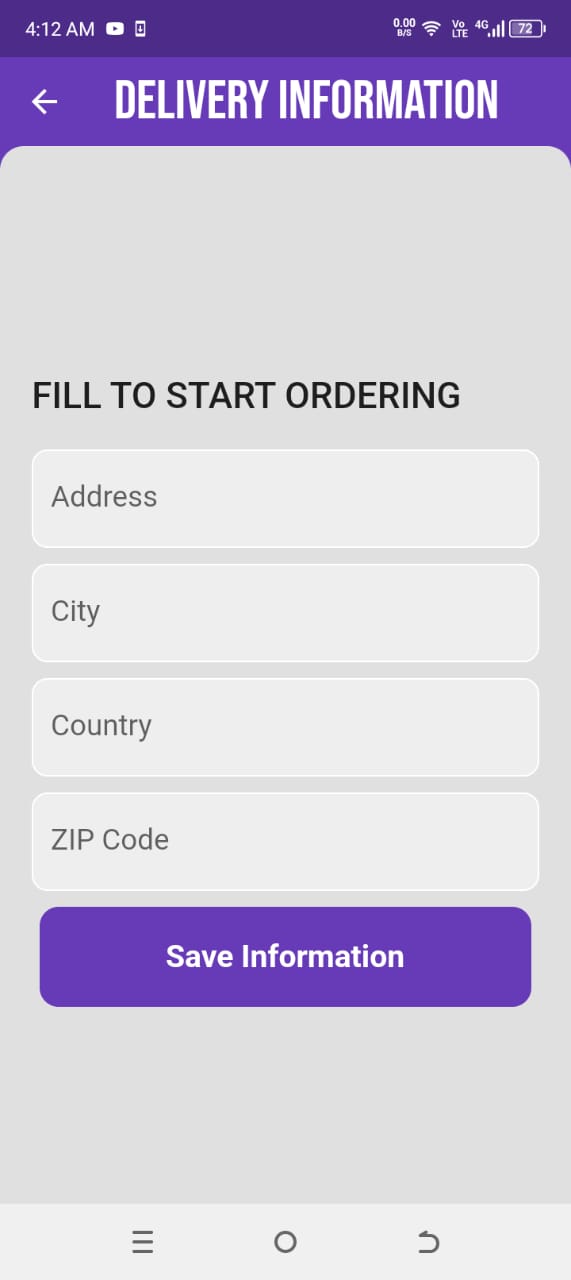
**7.5.7 Profile Page - Mock Screen 7**



**7.5.8 WhishList - Mock Screen 8**



**7.5.9 Delivery Information - Mock Screen 9**



# References

|  |  |
| --- | --- |
|  | D. Gupta, "How to Build an Online Auction Application? Benefits, Features, Costs", 2024. |
|  | G. Patil, "Online Auction App Development: A Complete Guide", India (Delhi), 2022. |
|  | J. Tiwary, "Auction App Development: Unlocking the Future of Online Bidding", India (Mumbai), 30 October, 2023. |
|  | M. Butt, "How to Create an Online Real-Time Bidding/Auction App", Mumbai (bendra), 2022. |
|  | Ajay, "How to Develop an Application for Online Auctions? Features, Advantages, and Costs", Chennai, 2024. |

### A4.OTHERTECHNICALDETAILDOCUMENTS

**Test Cases Document**

| **S. No** | **Description** | **Test Engineer** | **Start Date** | **End Date** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| 1 | Validate user login with correct and wrong info | Bismah | 02-Feb-2025 | 03-Feb-2025 |
| 2 | Check user registration and email verification | Umair | 04-Feb-2025 | 05-Feb-2025 |
| 3 | Test homepage tiles: Auction list, bids, chat etc. | Maham, Bismah | 06-Feb-2025 | 07-Feb-2025 |
| 4 | Post a product for auction | Bismah | 10-Feb-2025 | 11-Feb-2025 |
| 5 | Place a bid and test real-time update | Maham,Umair | 12-Feb-2025 | 13-Feb-2025 |
| 6 | Use chat room to message a seller | Bismah , Maham | 14-Feb-2025 | 15-Feb-2025 |
| 7 | Edit user profile and update password | Maham | 17-Feb-2025 | 18-Feb-2025 |
| 8 | Fill delivery info screen and submit | Umair | 19-Feb-2025 | 20-Feb-2025 |
| 9 | Add/remove item from wishlist | Bismah | 21-Feb-2025 | 22-Feb-2025 |
| 10 | View auction report with bidding history | Maham | 10-May-2025 | 10-May-2025 |
| 11 | View user activity report | Umair,Maham | 11-May-2025 | 11-May-2025 |
| 12 | Check auction summary report | Maham | 12-May-2025 | 12-May-2025 |
| 13 | View report of completed payments | Maham | 13-May-2025 | 13-May-2025 |
|  |  |  |  |  |

## **Module Name: Login Screen**

**Date**: 02-Feb-2025  
**Test Case ID**: TC-LS-001  
**Test Engineer**: Bismah

| **S. No** | **Steps** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 1 | Enter correct email and password | [Bismahimran31@gmail.com](mailto:Bismahimran31@gmail.com)  1234 | Login successful, goes to homepage | As expected | Pass |
| 2 | Enter correct email but wrong password | Bismahimran31@gmail.com / 0000 | Show “Invalid login” error | As expected | Pass |
| 3 | Leave password field empty | [bismah@gmail.com](mailto:bismah@gmail.com) / - | Show “Password is required” | As expected | Pass |
| 4 | Click on “Forgot Password” |  | Redirect to reset screen | As expected | Pass |

**Module Name: Registration Screen**

**Date**: 04-Feb-2025  
**Test Case ID**: TC-RS-001  
**Test Engineer**: Umair

| **S. No** | **Steps** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 1 | Fill name, email, password | Maham / [maham@gmail.com](mailto:maham@gmail.com) / xyz123 | Registration successful | As expected | Pass |
| 2 | Submit with empty fields | - | Show “All fields required” | As expected | Pass |
| 3 | Enter invalid email format | abc@com | Show “Enter valid email” | As expected | Pass |

## **Module Name: Bidding Screen**

**Date**: 6-Feb-2025  
**Test Case ID**: TC-BS-001  
**Test Engineer**: Maham, Bismah

| **S. No** | **Steps** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 1 | View product auction | - | Show item details with bid option | As expected | Pass |
| 2 | Place higher bid | 1500 | Bid accepted, status updated | As expected | Pass |
| 3 | Place lower bid | 1200 | Show “Bid too low” message | As expected | Pass |

### **Module Name: Product Posting Screen**

**Date**: 10-Feb-2025  
**Test Case ID**: TC-PRD-004  
**Test Engineer**: Maham Mirza

| **S. No** | **Steps** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 1 | Open “Post Product” screen | - | Form with fields is displayed | As expected | Pass |
| 2 | Enter item title and description | "Branded Shoes", "Limited Ed" | Title and description accepted | As expected | Pass |
| 3 | Upload image | image.jpg | Image uploaded and previewed | As expected | Pass |
| 4 | Submit without starting bid | - | Show error: “Starting bid required” | As expected | Pass |

### **Module Name: Place Bid Screen**

**Date**: 12-Feb-2025  
**Test Case ID**: TC-BID-005  
**Test Engineer**: Maham Mirza

| **S. No** | **Steps** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 1 | Select product | Product ID | Product detail page is shown | As expected | Pass |
| 2 | Place higher bid | 1500 | Bid placed successfully | As expected | Pass |
| 3 | Place lower bid | 900 | Error: Bid must be higher than current | As expected | Pass |

## **Module Name: Chat Room**

**Date**: 14-Feb-2025  
**Test Case ID**: TC-CR-001  
**Test Engineer**: Bismah, Maham

| **S. No** | **Steps** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 1 | Open chat screen | - | Chat loads with messages | As expected | Pass |
| 2 | Send message to seller | “Is it available?” | Message delivered to seller | As expected | Pass |
| 3 | Receive reply | - | New reply message shown in chat | As expected | Pass |

### Module Name: User profile and Update pass

**Date**: 17-Feb-2025  
**Test Case ID**: TC-UP-001  
**Test Engineer**: Maham Mirza

| **S. No** | **Steps** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 1 | Open Profile screen | - | Display user details (name, email etc.) | As expected | Pass |
| 2 | Tap on Edit Profile | - | Editable fields appear | As expected | Pass |
| 3 | Update name and phone number | Name: "Bismah", Phone: "0300..." | Changes saved and updated on profile | As expected | Pass |
| 4 | Leave required field empty | Name: "" | Show error: “Name is required” | As expected | Pass |
| 5 | Change password | Old: 1234, New: xyz@2025 | Password updated with confirmation | As expected | Pass |

### **Module Name: Delivery Info Screen**

**Date**: 19-Feb-2025  
**Test Case ID**: TC-DEL-008  
**Test Engineer**: Umair

| **S. No** | **Steps** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 1 | Fill delivery form | Name, Address, Phone | Details accepted and saved | As expected | Pass |
| 2 | Submit with empty fields | - | Show error: "All fields required" | As expected | Pass |
| 3 | View confirmation message | - | "Delivery info submitted" shown | As expected | Pass |
|  |  |  |  |  |  |

## **Module Name: Wishlist Screen**

**Date**: 21-Feb-2025  
**Test Case ID**: TC-WS-001  
**Test Engineer**: Bismah

| **S. No** | **Steps** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 1 | Add item to wishlist | Tap on "♡" icon | Item added to wishlist | As expected | Pass |
| 2 | Remove item from wishlist | Tap on "♡" again | Item removed from wishlist | As expected | Pass |

## **Module Name: Bidding History**

**Date**: 10-May-2025  
**Test Case ID**: TC-RPT-001  
**Test Engineer**: Maham

| **S. No** | **Steps** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
| 1 | Open Bidding History Report | - | Shows list of bids placed | As expected | Pass |
| 2 | Click on specific auction | - | Shows all bids with user names | As expected | Pass |
|  |  |  |  |  |  |

### **Module Name: User Activity**

**Date**: 11-May-2025  
**Test Case ID**: TC-RPT-011  
**Test Engineer**: Umair,Maham

| **S. No** | **Steps** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 1 | Open User Activity Report | - | Shows user login, posting, and bidding history | As expected | Pass |
| 2 | Select user | User ID | Displays that user's full activity timeline | As expected | Pass |
|  |  |  |  |  |  |

### **Module Name: Auction Summary**

**Date**: 12-May-2025  
**Test Case ID**: TC-RPT-012  
**Test Engineer**: Maham

| **S. No** | | **Steps** |  | **Input Data** | | | **Expected Result** | | **Actual Result** | | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | | Open Auction Summary |  | - | | | List of all auctions with summary shown | | As expected | | Pass |
| 2 | | Click specific auction |  | Auction ID | | | Shows full details of selected auction | | As expected | | Pass |
| 3 | View confirmation message | |  | | - |  | | "Delivery info submitted" shown | | As expected | Pass |

### **Module Name: Payment Status**

**Date**: 13-May-2025  
**Test Case ID**: TC-RPT-013  
**Test Engineer**: Maham

| **S. No** | **Steps** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 1 | Open Payment Status Report | - | Displays list of completed payments | As expected | Pass |
| 2 | View payment for one auction | Auction ID | Shows buyer name and payment status | As expected | Pass |

**UI/UXDetailDocument**

## 1. Introduction

This document explains how the design of the Online Auction App makes it **easy and enjoyable** to use. UI (User Interface) is about **what users see**, like buttons and screens, while UX (User Experience) is about **how users feel** when using the app — whether it’s smooth, fast, and frustration-free.

Our goal is to build an app that is **simple, clear**, and **works well on any mobile phone,** so that users can buy, sell, and bid on items without confusion.

## 2. Design Principles

We followed these **easy-to-understand rules** to make sure the app is friendly and looks good:

### Simplicity

* The app shows **only what is necessary**, so users don’t feel overwhelmed.
* Clean layout with readable fonts and good spacing.

### Consistency

* Same **color theme**, **button styles**, and **icons** throughout the app.

### Mobile First

* The design is made **especially for phones**, but it also works on tablets.

### Feedback and Response

* When users take an action (like placing a bid), the app **instantly shows a result** (success message, or error).

### Accessibility

* Easy to use for **all types of users**, including those who may have difficulty seeing or reading.

## UI Components

| **Page** | **What It Does** |
| --- | --- |
| **Login Screen** | Lets users enter their email and password to sign in. |
| **Sign Up Screen** | New users can create an account. |
| **Homepage** | Shows ongoing auctions, popular items, and buttons for navigation. |
| **Auction Detail Page** | Shows item picture, description, current bid, and bid button. |
| **Post Product Screen** | Sellers enter item details and upload an image. |
| **Chat Room** | Allows buyers and sellers to message each other. |
| **Wishlist** | Users can save favorite items. |
| **Delivery Info Page** | Buyers enter address and phone number for delivery. |
| **Profile Page** | View and update user info like name and password. |
| **Notification Popups** | Show when bids are placed or auctions end. |

## User Flow

### ****Buyer’s Flow****

1. Open app → Login or Sign up
2. Browse products → Tap on an auction
3. Place a bid → Get updates
4. Win auction → Enter delivery info
5. Receive item

### ****Seller’s Flow****

1. Login → Tap “Post Product”
2. Add item details → Set auction time
3. Submit → View bids in real-time
4. Message buyer → Deliver item

### ****Admin’s Flow****

1. Login → Manage users and items
2. View reports → Block spam users
3. Monitor activity

## 5. Tools and Technologies Used

| **Tool** | **Purpose** |
| --- | --- |
| **Flutter** | A tool used to design beautiful mobile apps (works for Android & iOS). |
| **Firebase** | A Google tool that stores user data, sends notifications, and handles login securely. |
| **Dart** | The programming language used to write the app in Flutter. |
| **Figma/Canva (for Mockups)** | Used to create the screen designs before building the real app. |
| **GitHub** | Stores our code safely and allows teamwork. |

## Summary

The UI/UX of our Online Auction App is carefully designed to be:

* Easy for first-time users
* Fast and mobile-friendly
* Visually clean and consistent
* Helpful through instant responses and notifications

**CodingStandardsDocument**

## 1. Introduction

This document explains the **coding rules** and **good habits** our team followed while building the Online Auction App. Following these standards helps us keep the code:

* **Clean and organized**
* **Easy to understand**
* **Secure**
* **Easy to update later**

## 2. General Guidelines

These are some **basic rules** that apply to all parts of the app:

* **Use meaningful names**  
  Example: placeBid() is better than pb() because it tells you what it does.
* **Keep your code neat**  
  Indent properly, leave spaces where needed, and avoid writing everything in one line.
* **Write comments**  
  Add short notes in the code to explain tricky parts.
* **Avoid repeating code**  
  Use functions to avoid writing the same code again and again.
* **Use version control**  
  All changes were saved using GitHub so nothing is lost and we can go back to older versions if needed.

## 3. Frontend Coding Standards (Flutter with Dart)

### File and Folder Structure

* Each screen has its own folder (e.g., login/, auction/)
* All images stored in assets/images/

### Naming Rules

* Files: lowercase\_with\_underscores.dart (e.g., login\_screen.dart)
* Classes: PascalCase (e.g., LoginScreen)
* Variables/Functions: camelCase (e.g., userName, placeBid())

### Widget Structure

* Use **StatelessWidget** when UI doesn't change.
* Use **StatefulWidget** when UI updates (like when bidding).

### UI Testing

* Used flutter\_test to test if widgets appear correctly (e.g., checking if login button shows up).

## 4. Backend Coding Standards

The backend logic is handled using **Firebase** with **Dart services**.

### API Integration

* Used Firebase APIs for authentication, database (Firestore), and messaging.

### Code Organization

* auth\_service.dart for login/signup
* auction\_service.dart for managing auctions
* chat\_service.dart for handling messages

### Function Standards

* Small and focused: Each function does one thing.
* Clear error messages: Show helpful alerts if something goes wrong.

### Backend Testing

* Checked if data is correctly saved, like:
  + Can user sign in?
  + Is a bid added to the database?
  + Can admin remove a user?

## 5. Database Standards

### Collection Naming

* Collection names are plural and lowercase: users, auctions, bids

### Timestamps

* Always saved in standard format: yyyy-mm-dd hh:mm

## 6. Security Practices

We followed **important safety rules** to keep user data and transactions secure.

### Passwords & Login

* Passwords are never stored as plain text.
* Used **Firebase Authentication** for secure login.
* Email verification is required.

### Data Protection

* Data transferred between users and app is **encrypted.**
* Only logged-in users can access their data.

### Role-Based Access

* Users can only do what their role allows:
  + Buyers can't delete auctions.
  + Admins have full control.

### Common Risks Avoided

* Prevented **fake bids** using real-time validation.
* Blocked **invalid login attempts** with error alerts.
* Monitored **user activity** to detect spam or abuse.

## Summary

By following these coding standards, we:

* Keep the code **neat and easy to understand**
* Make the app **secure and stable**
* Allow future team members to **maintain or improve** the app easily

**Project Policy Document**

## 1. Introduction

This document explains the **rules**, **responsibilities**, and **working method** of our Online Auction App project. It helps all team members and supervisors understand how we work, what everyone’s role is, and how we keep the project on track.

Our goal is to **build a secure, easy-to-use app** where people can buy and sell items through bidding. To do that smoothly, we need good teamwork, communication, and clear policies.

## 2. Getting Started

This section helps understand how the project begins and how we organize our tasks.

### Team Members & Roles

| **Name** | **Role** |
| --- | --- |
| M. Umair | Team Leader (Manages tasks, reviews work) |
| Bismah Imran | Developer & Tester (Designs and tests screens) |
| Maham Mirza | Developer & Tester (Builds features, reports bugs) |
| Osama Ahmed Khan | Supervisor (Guides the team and approves work) |

### Timeline / Working Plan

| **Phase** | **Time Period** |
| --- | --- |
| Planning & Research | Noc 2024 |
| Designing Screens | Jan 2025 |
| Development (Coding) | Feb – Mar 2025 |
| Testing & Fixing Bugs | April 2025 |
| Final Submission | June 2025 |

### Tools We Use

| **Tool** | **Why We Use It** |
| --- | --- |
|  |  |
| Flutter | To build the app interface |
| Firebase | To save user data and manage login |
| GitHub | To store code and track changes |
| WhatsApp | For team communication |
| Google Docs | For writing and sharing reports |

## 3. Features of Our Working Style

### Team Rules

* Everyone must attend weekly meetings.
* Everyone should complete their assigned tasks on time.
* Help each other and respect everyone's ideas.

### Coding & Designing Rules

* Follow neat and clean coding.
* Design screens that are easy to use and look good.
* Test everything before submitting.

### Communication

* Use WhatsApp for quick updates.
* Share progress in group meetings.
* Inform the team in case of any delay or problem.

### Safety & Backup

* Save project files on GitHub to avoid losing work.
* Backup Firebase data weekly.

### Testing & Feedback

* Every feature must be tested before final use.
* Bugs are fixed during the testing phase.
* Users should get easy error messages if something goes wrong.

### Fair Use & Ethics

* No copying others’ code or using illegal software.
* Keep all user data private and protected.
* Only real items should be posted for auction — no scams.

## Summary

Our Project Policy makes sure that:

* Everyone in the team knows their role
* We work in an organized and respectful way
* The project is safe, well-tested, and delivered on time.

**User Manual Document**

## 1. Introduction

Welcome to the **Online Auction App** is a simple and fun way to **buy and sell items through bidding**. This app lets you find great deals, post your own items for sale, and take part in auctions in real time all from your mobile phone.

## Key Features

| **Feature** | **What It Does** |
| --- | --- |
|  |  |
| **Register/Login** | Create an account or log in securely. |
| **Browse Items** | See different items up for auction. |
| **Place Bids** | Take part in auctions and offer your price. |
| **Post Your Item** | Sell your item by setting a starting bid and auction time. |
| **Chat** | Talk to buyers or sellers directly inside the app. |
| **Notifications** | Get instant alerts for new bids or auction results. |
| **Wishlist** | Save items you like for later. |
| **Delivery Info** | Enter your address and phone number after winning. |
| **Profile Settings** | Change your personal info or password anytime. |

## Getting Started

### Step 1: Install the App

* Download from the **Google Play Store** (Android only).
* Open the app on your mobile phone.

### Step 2: Create an Account

* Tap on **Sign Up.**
* Fill in your name, email, password.
* You’ll get a **verification email** — open it and click the link.

### Step 3: Log In

* Use your email and password to **log in.**
* You’ll be taken to the **homepage** where you can start browsing.

### Step 4: Start Using the App

* **Buyers**: Tap on an item then Place bid then Get alerts.
* **Sellers**: Tap **Post Item**  then Add title, image, price then Start auction.

## Troubleshooting

| **Problem** | **What to Do** |
| --- | --- |
| Can’t log in | Check your email/password. Use **“Forgot Password”** if needed. |
| Didn’t get email verification | Check your **Spam or Junk** folder. Tap “Resend” in the app. |
| App is slow or stuck | Close and reopen the app. Make sure your internet is working. |
| Can't place a bid | Make sure your bid is **higher** than the current one. |
| Chat not working | Try refreshing the app or check your internet connection. |
| Delivery info not saving | Make sure all fields (name, phone, address) are filled correctly. |

## 5. Support

If you still need help, here’s how you can contact us:

* **Email Support:** [support@auctionapp.com](mailto:support@auctionapp.com)
* **In-App Help:** Go to Profile → Help Center
* **Live Chat:** Available 10:00 AM – 6:00 PM (Mon–Sat)
* **FAQs:** Check the section in the app

.

### A5.FLYER&POSTERDESIGN

**A6. COPY OF EVALUATION COMMENTS** **COPYOFEVALUATIONCOMMENTSBYSUPERVISOR FOR PROJECT – I MID SEMESTER EVALUATION**

APhotostat or scanned copyshould beplaced whensubmittingdocument toProject

Coordinator.(**Note**:Pleaseremovethislinewhenattachcopythatis required)

### COPYOFEVALUATIONCOMMENTSBYJURYFOR PROJECT – I END SEMESTER EVALUATION

APhotostatorscannedcopyshouldbeplacedwhensubmittingdocumentto Project Coordinator. (**Note**: Please remove this line when attach copy that is required)

### COPYOFEVALUATIONCOMMENTSBYSUPERVISOR FOR PROJECT – II MID SEMESTER EVALUATION

APhotostatorscannedcopyshouldbeplacedwhensubmittingdocumentto Project Coordinator. (**Note**: Please remove this line when attach copy that is required)

#### A7.MEETINGS’MINUTES&Sign-OffSheet

Original Documents should be placed when submitting document to Project Coordinator. Document should be signed bythe supervisor and all othermembers present in the meeting (wherever possible). (**Note**: Please remove this line when attach copy that is required) Weeklymeetings’minutesarerequired(heldwithSupervisorand/orwithclient).Important group discussions can also be included here.

### A8.DOCUMENTCHANGERECORD

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Change Details** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### A9.PROJECTPROGRESS

PhotostatofIncrementalversionsofRequirementSignoffsheetsubmittedtoProject Coordinator. (**Note**: Please remove this line when attach copy that is required)