IT2020 – Software Engineering
Year 2, Semester I, 2024
Group Assignment

<<Online Biding System >>

Group Details

Campus: Malabe

SE/OOP Group Number: SE/OOP/2024/S1/MLB/WE/G14

	Student Registration Number	Student Name
1	IT22190598	Gamalath K.H
2	IT22117250	SILVA A.S.N
3	IT22278708	Alwis L.W.R.T
4	IT22285010	Bamunuge M.M

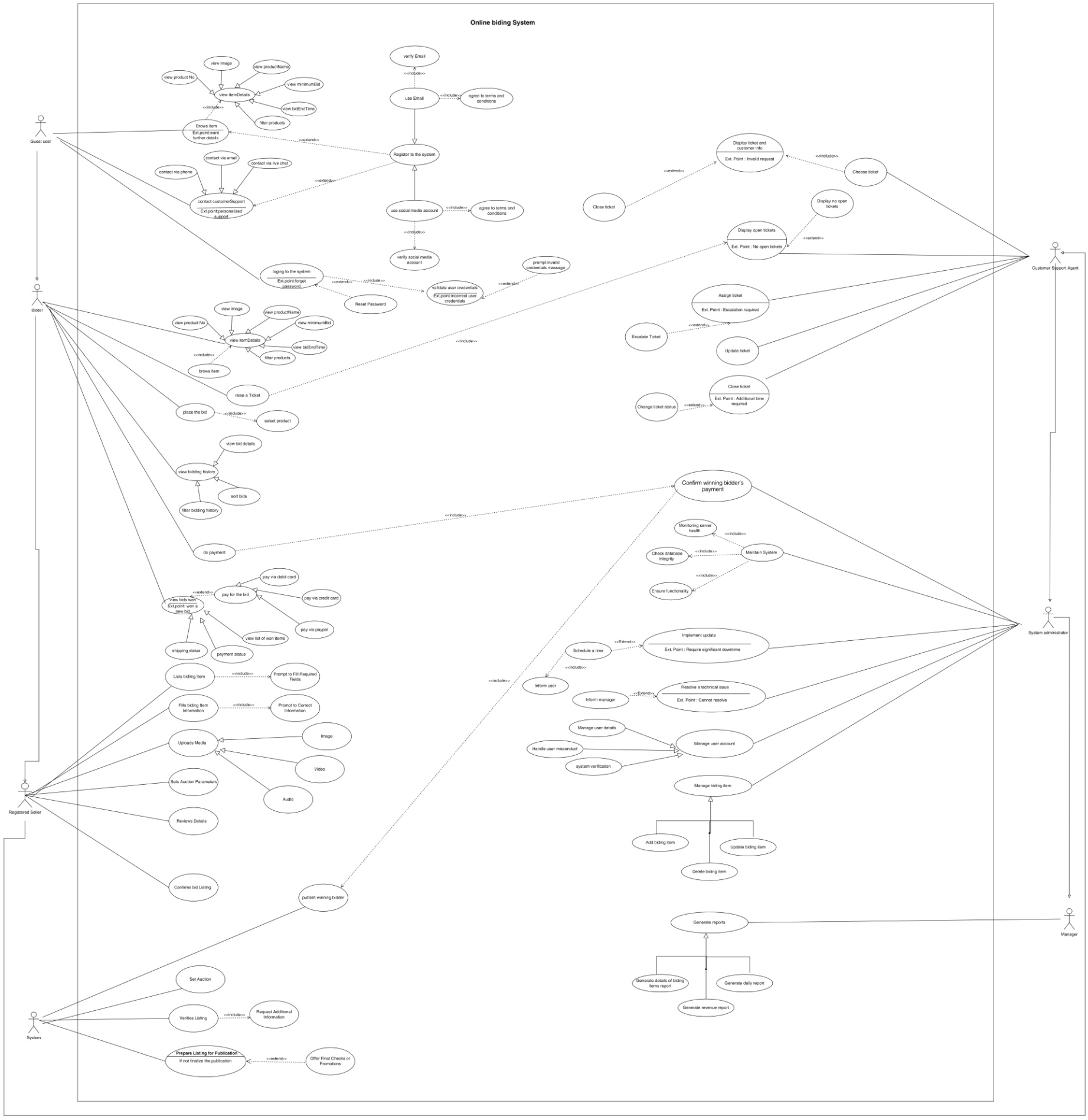
Title of the Project: Online Bidding System

Introduction

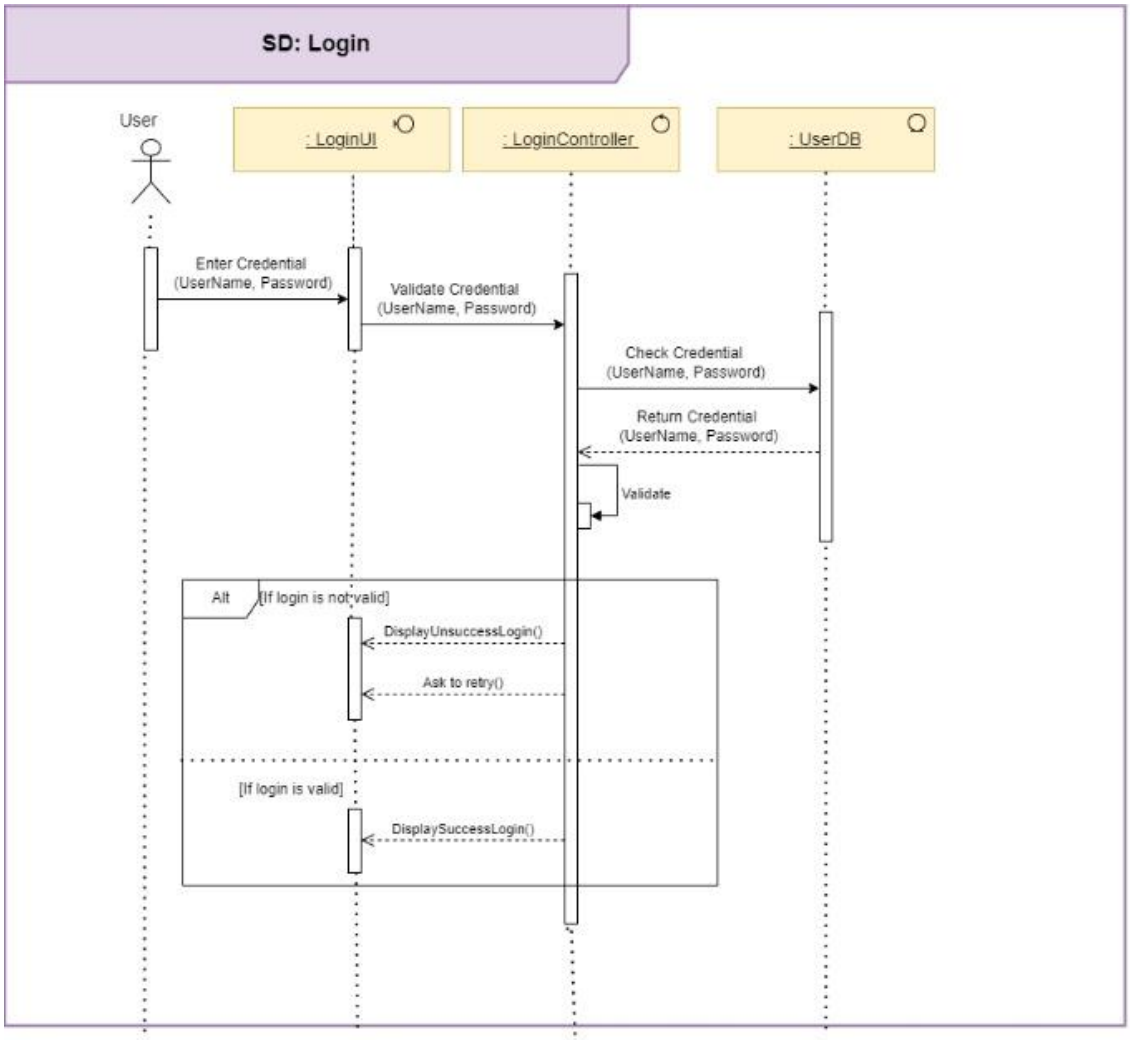
The Online Bidding System project is to provide a strong and intuitive online auction platform that will make it easier for people to purchase and sell a variety of goods through an online marketplace. Online bidding platforms have grown in popularity in the current digital era because they are easy to use, readily available, and can link buyers and sellers from different parts of the world.

The goal of this project is to provide a complete online bidding system that meets the needs of registered purchasers as well as guest users, ensuring a safe and easy bidding process. The Online Bidding System aims to completely transform the way online auctions are held by utilizing cutting-edge web technology and best practices in software development.

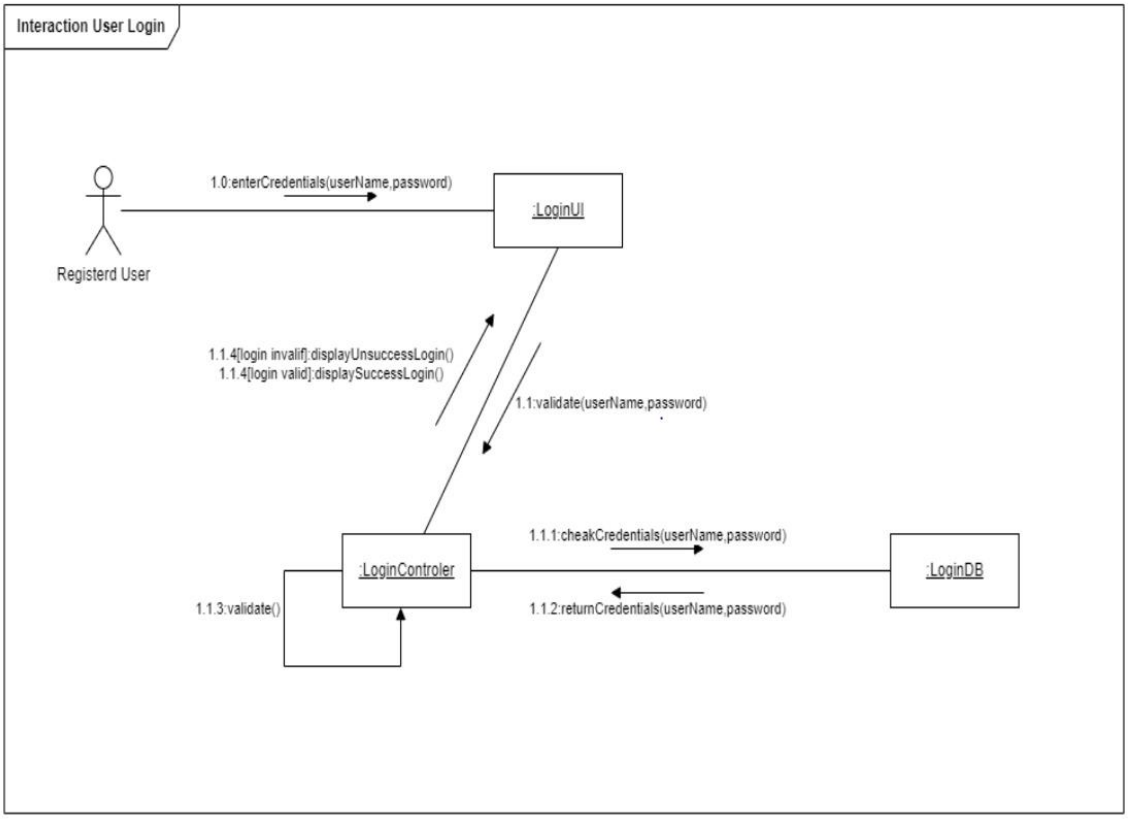
All parties involved in the Online Bidding System project, including individual sellers, companies, and auction fans, stand to gain from the project's successful completion. In addition to giving customers the chance to find unusual products and take part in thrilling bidding battles from the comfort of their homes, it will give sellers a platform to reach a larger audience and optimize the value of their items.


Online biding System
Guest user
view image
view productName
view product No
view minimumBid
view itemDetails
view bidEndTime
filter products
Brows item
Ext.point:want further details
contact via email
contact via live chat
contact via phone
contact customerSupport
Ext.point:personalized support
verify Email
<<include>>
use Email
<<include>>
agree to terms and conditions
Register to the system
<<extend>>
use social media account
<<include>>
agree to terms and conditions
<<include>>
verify social media account
loging to the system
Ext.point:forget password
<<extend>>
<<include>>
validate user credentials
Ext.point:incorrect user credentials
<<extend>>
prompt invalid credintials massage
Reset Password
Bidder
view image
view productName
view product No
view minimumBid
view itemDetails
view bidEndTime
filter products
<<include>>
brows item
raise a Ticket
place the bid
<<include>>
select product
view bid details
view bidding history
sort bids
filter bidding history
do payment
<<include>>
Confirm winning bidder's payment
pay via debit card
pay for the bid
pay via credit card
pay via paypal
<<extend>>
view bids won
Ext.point: won a new bid
view list of won items
shipping status
payment status
Lists bidding Item
<<include>>
Prompt to Fill Required Fields
Fills biding Item Information
<<include>>
Prompt to Correct Information
Uploads Media
Image
Video
Audio
Sets Auction Parameters
Reviews Details
Confirms bid Listing
publish winning bidder
Registered Seller
Set Auction
Verifies Listing
<<include>>
Request Additional Information
Prepare Listing for Publication
If not finalize the publication
<<extend>>
Offer Final Checks or Promotions
System
<<include>>
Close ticket
<<extend>>
Display ticket and customer info
Ext. Point : Invalid request
<<include>>
Choose ticket
Display no open tickets
Display open tickets
Ext. Point : No open tickets
<<extend>>
Customer Support Agent
Assign ticket
Ext. Point : Escalation required
<<extend>>
Escalate Ticket
Update ticket
Close ticket
Ext. Point : Additional time required
Change ticket status
<<extend>>
Monitoring server health
<<include>>
Check database integrity
<<include>>
Maintain System
<<include>>
Ensure functionality
System administrator
Implement update
Ext. Point : Require significant downtime
<<Extend>>
Schedule a time
<<include>>
Inform user
Inform manager
<<Extend>>
Resolve a technical issue
Ext. Point : Cannot resolve
Manage user details
Handle user misconduct
system verification
Manage user account
Manage biding item
Add biding item
Update biding item
Delete biding item
Generate reports
Generate details of biding items report
Generate daily report
Generate revenue report
Manager

Sequence diagram for Login (for all users)



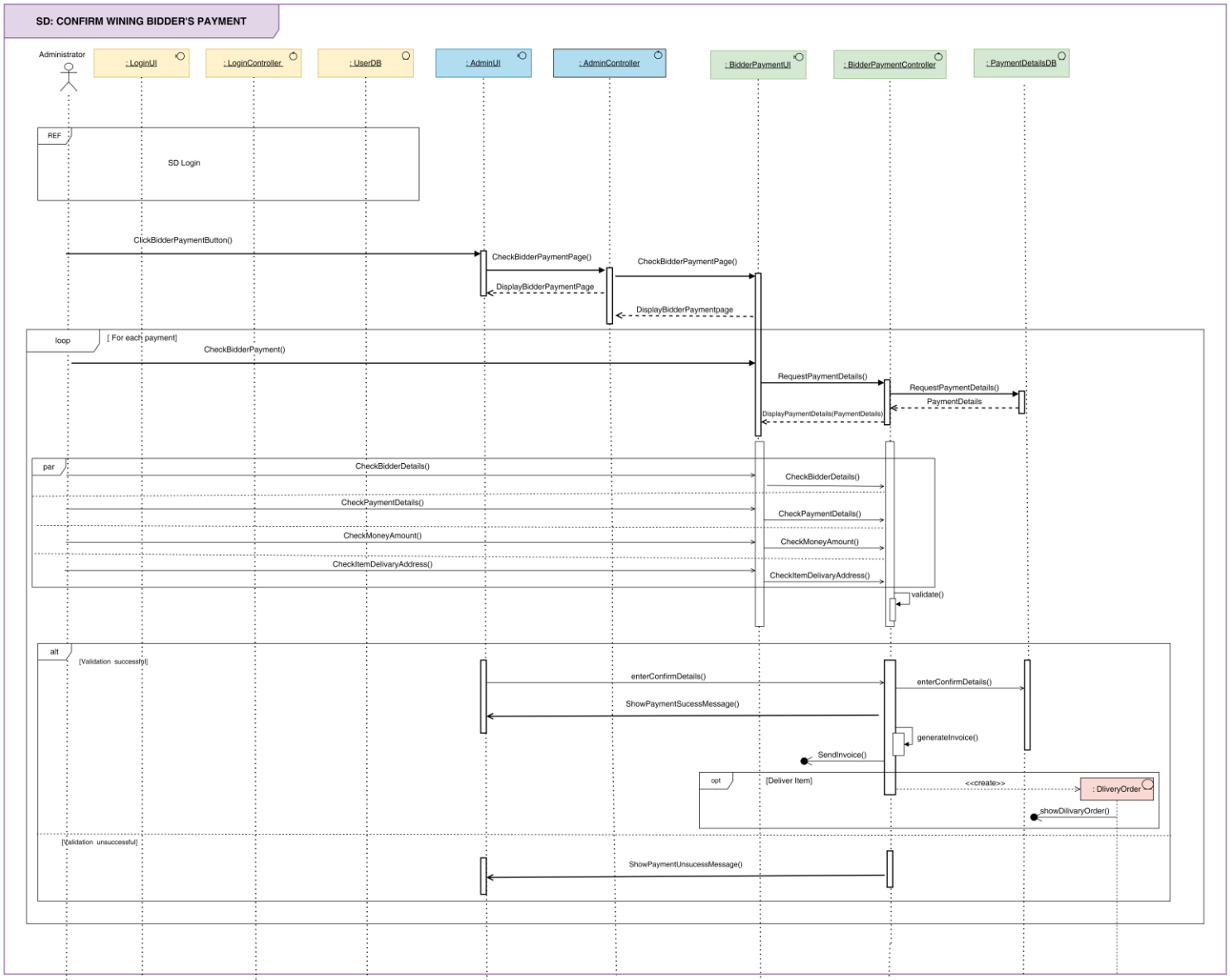
Communication diagram for login (For all users)

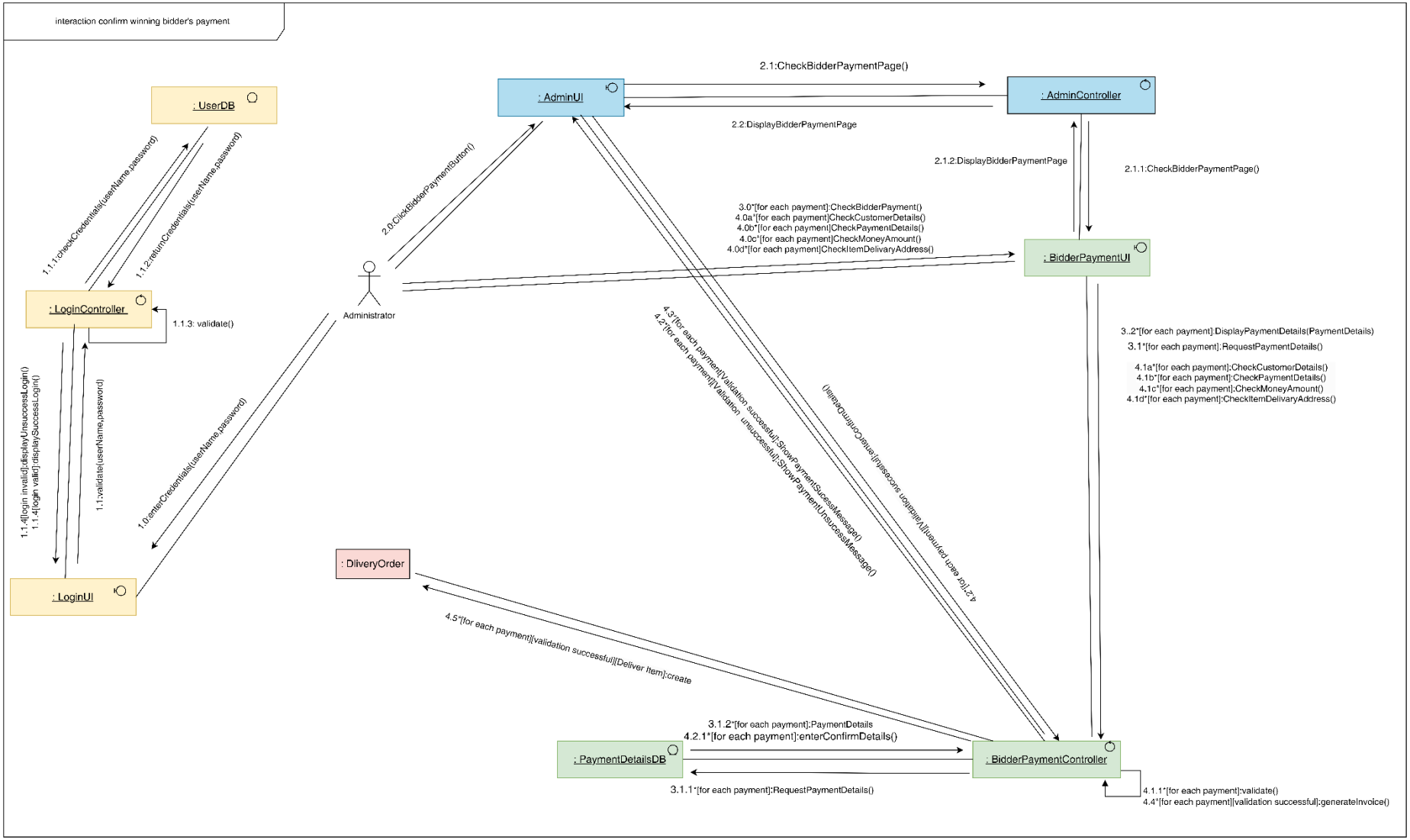


1. IT22190598 Gamalath K.H

Number	1	
Name	Confirm winning bidder's payment	
Summary	Administrator checks the winning bidder's Payment and validates the payment	
Priority	High	
Precondition	• Administrator account exists in the system. • Administrator has a valid username and password. • The auction for the item has concluded, with a winning bidder identified. • The winning bidder's payment information is submitted in the system.	
Postcondition	• Administrator successfully verifies and confirms the winning bidder's payment. • An invoice is generated for the winning bidder and added to the system. • A confirmation message is sent to the winning bidder, notifying them of successful payment processing and next steps (e.g., item delivery or collection). • Administrator this process until the last payment.	
Primary Actors	Administrator	
Triggers	Chosen the Bidder's payment details page	
Main Scenario	**Step**	**Action**
	1	Admin logs into the system with username and password.
	2	Admin navigates to the 'Bidder's Payment' section.
	3	Select a payment.
	4	Admin check the payment details (payment details, customer details, payment amount, Item delivery address)
	5	Validate the details.
	6	Add confirmed payment details to the database.
	7	System generates the invoice.
	8	System sends the invoice to the winning bidder.
Extension	1. a	Admin ID or password mismatch exits from the login page.
	5. a	If the bidder submits the invalid payment details, Send a message to the admin.
	5. b	If the bidder submits valid payment details, Send the message "Payment successful."

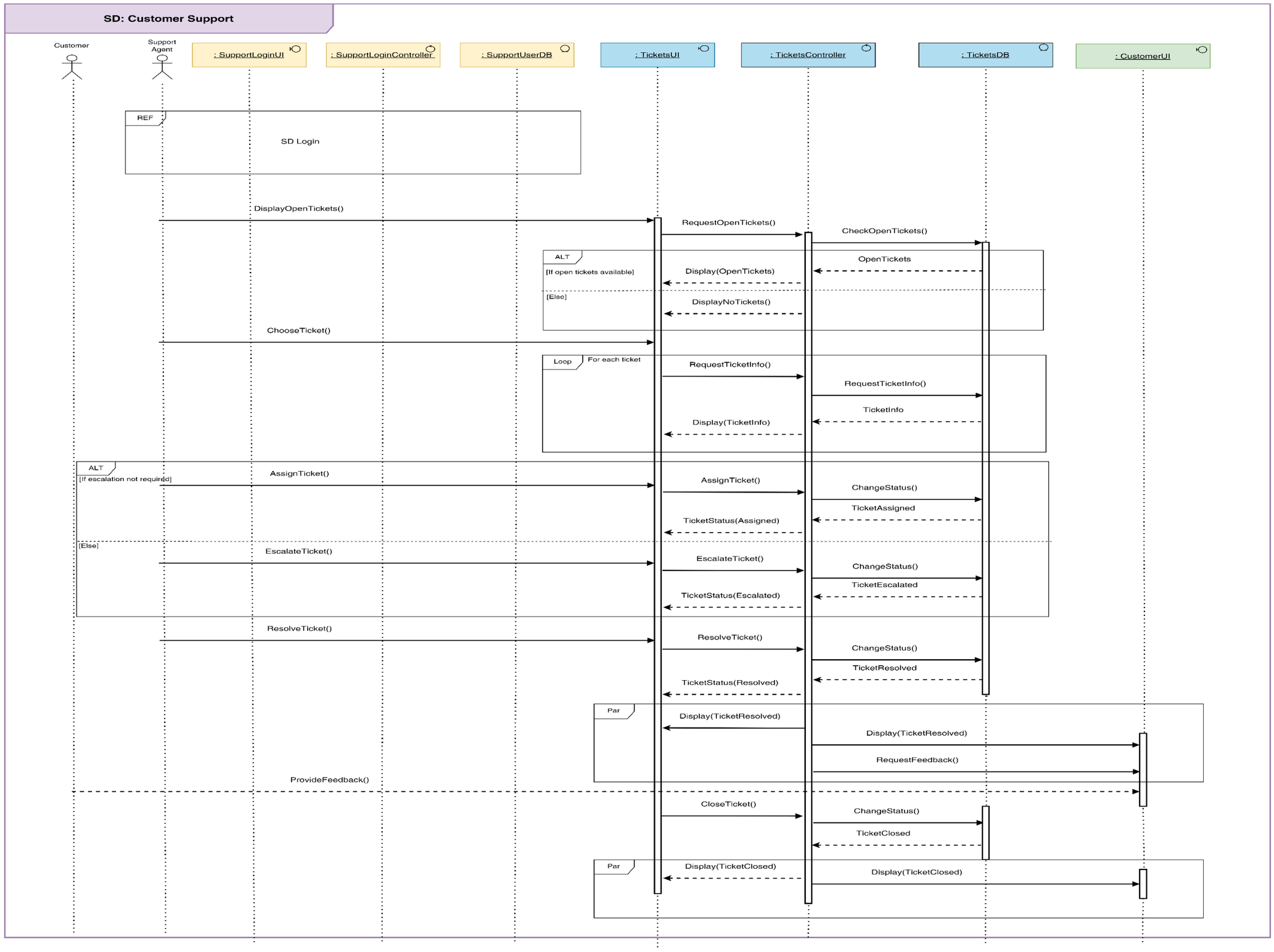
	8. a	If the winning bidder also wants to deliver the item make a delivery order

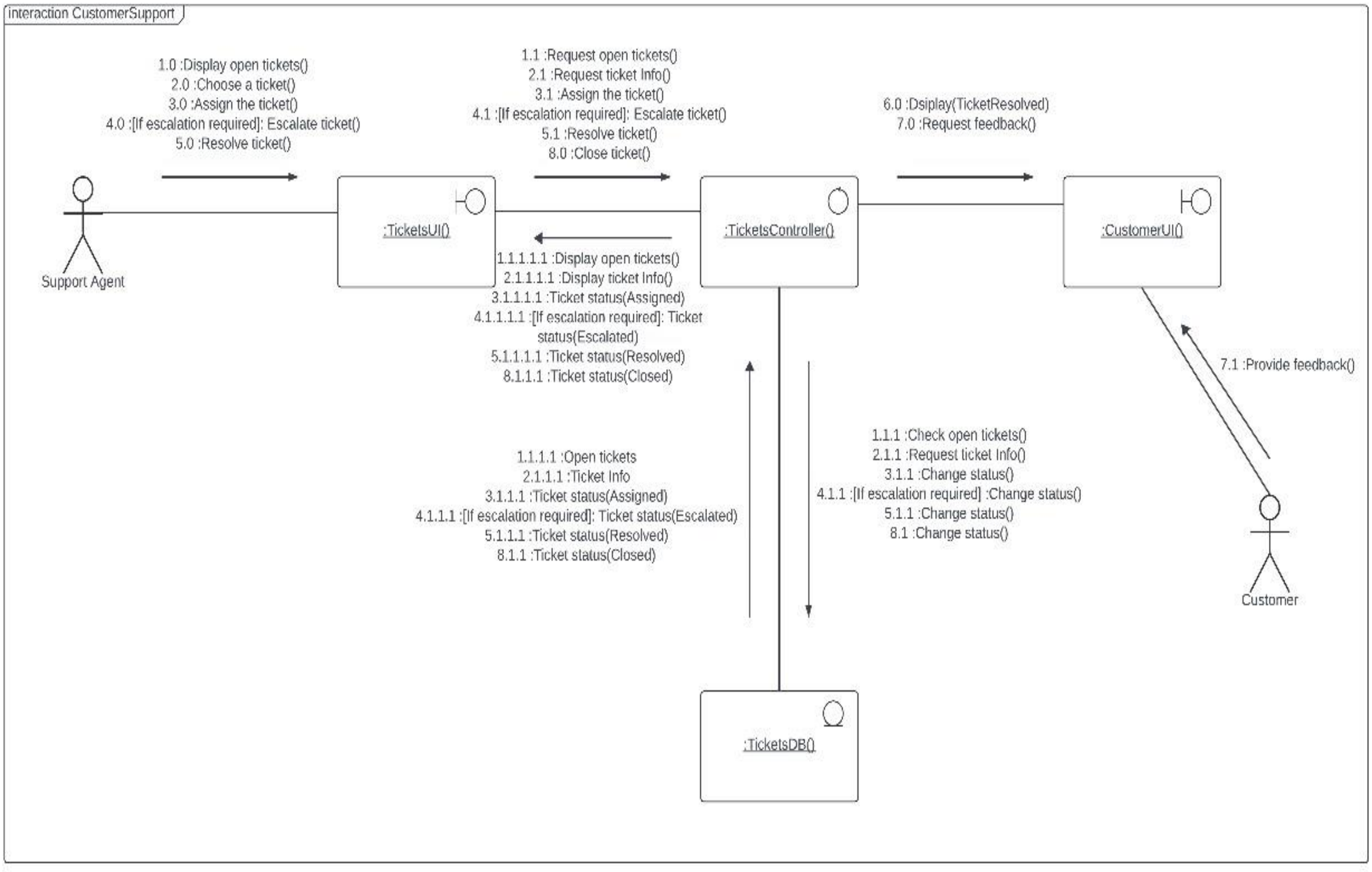

SD: CONFIRM WINING BIDDER'S PAYMENT
Administrator
: LoginUI
: LoginController
: UserDB
: AdminUI
: AdminController
: BidderPaymentUI
: BidderPaymentController
: PaymentDetailsDB
REF
SD Login
ClickBidderPaymentButton()
CheckBidderPaymentPage()
CheckBidderPaymentPage()
DisplayBidderPaymentPage
DisplayBidderPaymentpage
loop
[For each payment]
CheckBidderPayment()
RequestPaymentDetails()
RequestPaymentDetails()
PaymentDetails
DisplayPaymentDetails(PaymentDetails)
par
CheckBidderDetails()
CheckBidderDetails()
CheckPaymentDetails()
CheckPaymentDetails()
CheckMoneyAmount()
CheckMoneyAmount()
CheckItemDelivaryAddress()
CheckItemDelivaryAddress()
validate()
alt
[Validation successful]
enterConfirmDetails()
enterConfirmDetails()
ShowPaymentSucessMessage()
generateInvoice()
SendInvoice()
opt
[Deliver Item]
<<create>>
: DliveryOrder
showDilivaryOrder()
[Validation unsuccessful]
ShowPaymentUnsucessMessage()


interaction confirm winning bidder's payment
: UserDB
: AdminUI
: AdminController
2.1:CheckBidderPaymentPage()
2.2:DisplayBidderPaymentPage
2.1.2:DisplayBidderPaymentPage
2.1.1:CheckBidderPaymentPage()
1.1.1:checkCredentials(userName,password)
1.1.2:returnCredentials(userName,password)
2.0:ClickBidderPaymentButton()
3.0*[for each payment]:CheckBidderPayment()
4.0a*[for each payment]CheckCustomerDetails()
4.0b*[for each payment]CheckPaymentDetails()
4.0c*[for each payment]CheckMoneyAmount()
4.0d*[for each payment]CheckItemDelivaryAddress()
: BidderPaymentUI
: LoginController
1.1.3: validate()
Administrator
3..2*[for each payment]:DisplayPaymentDetails(PaymentDetails)
3.1*[for each payment]:RequestPaymentDetails()
4.1a*[for each payment]:CheckCustomerDetails()
4.1b*[for each payment]:CheckPaymentDetails()
4.1c*[for each payment]:CheckMoneyAmount()
4.1d*[for each payment]:CheckItemDelivaryAddress()
1.1.4[login invalid]:displayUnsuccessLogin()
1.1.4[login valid]:displaySuccessLogin()
1.1:validate(userName,password)
1.0:enterCredentials(userName,password)
4.3*[for each payment][Validation successful]:ShowPaymentSuccessMessage()
4.2*[for each payment][Validation unsuccessful]:ShowPaymentUnsuccessMessage()
4.2*[for each payment][validation successful]:enterConfirmDetails()
: DliveryOrder
: LoginUI
4.5*[for each payment][validation successful][Deliver Item]:create
3.1.2*[for each payment]:PaymentDetails
4.2.1*[for each payment]:enterConfirmDetails()
: PaymentDetailsDB
: BidderPaymentController
3.1.1*[for each payment]:RequestPaymentDetails()
4.1.1*[for each payment]:validate()
4.4*[for each payment][validation successful]:generateInvoice()

2. IT22117250 SILVA A.S.N

Number	2	
Name	Support Customer	
Summary	Customer support agent support the customer	
Priority	Medium	
Pre-conditions	Customer support agent has logged in to the system Customer has reached customer support	
Post-conditions	Customer has got a resolution The support case was escalated to the relevant party	
Primary Actors	Customer Support Agent	
Secondary Actors	Customer	
Trigger	Customer has chosen customer support	
Main Scenario	**Step**	**Action**
	1	System displays open tickets
	2	Customer support agent choose a ticket
	3	System displays ticket information
	4	System displays customer's details
	5	Customer support agent get the ticket assigned to him/herself
	6	Customer support agent resolves the issue
	7	Customer support agent updates the ticket
	8	Customer support agent closes the ticket
	9	System notifies the customer
	10	System requests customer's feedback
Extensions	**Step**	**Branching Action**
	1a	**No open tickets, system displays message**
	3a	**Invalid request, agent close ticket**
	5b	**Escalation required, agent escalates ticket**
	8a	**Additional time required, ticket status changed to WIP**
	10a	**Customer is not happy with the resolution, ticket reopens**

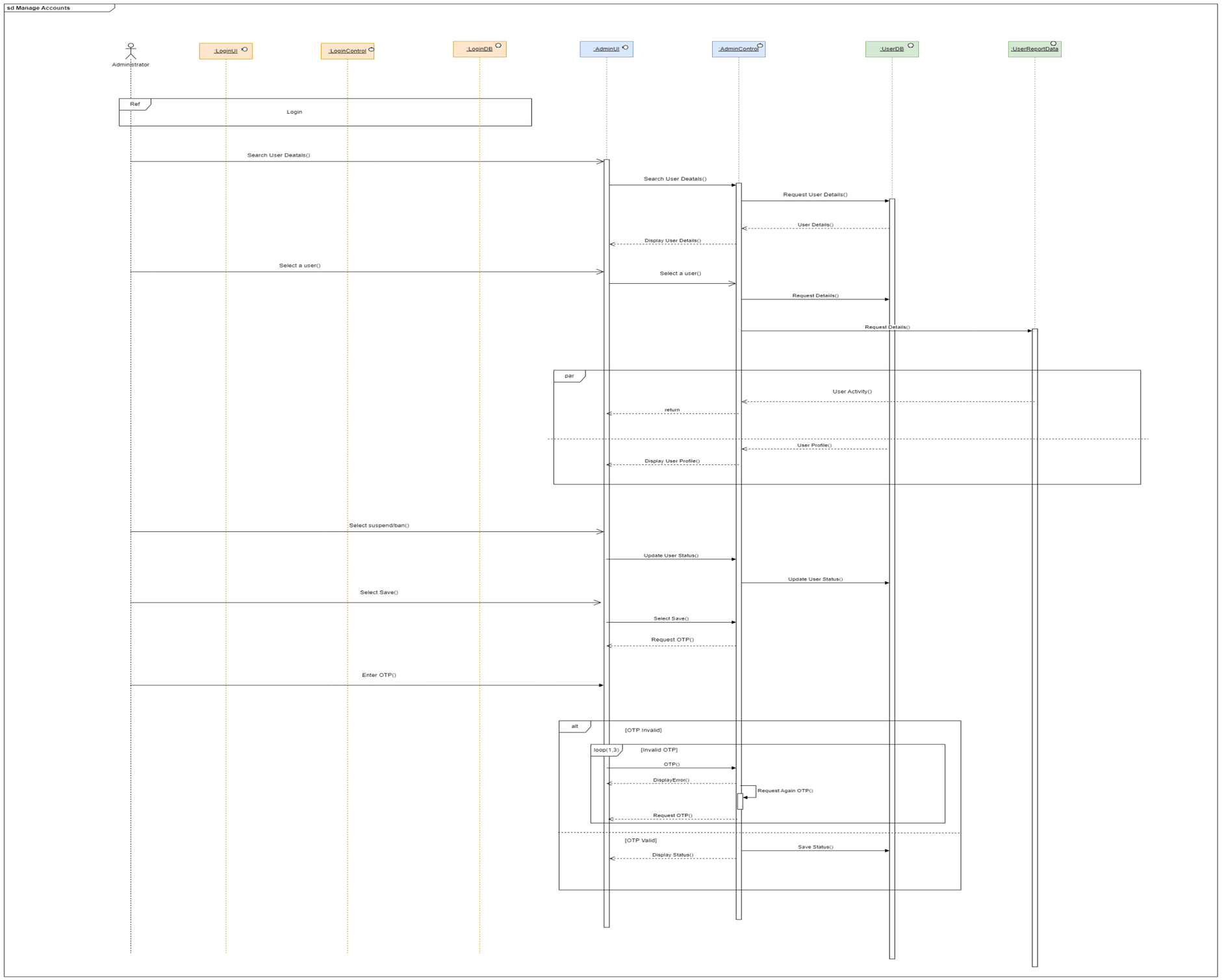

SD: Customer Support
Customer
Support Agent
: SupportLoginUI
: SupportLoginController
: SupportUserDB
: TicketsUI
: TicketsController
: TicketsDB
: CustomerUI
REF
SD Login
DisplayOpenTickets()
RequestOpenTickets()
CheckOpenTickets()
ALT
[If open tickets available]
OpenTickets
Display(OpenTickets)
[Else]
DisplayNoTickets()
ChooseTicket()
Loop
For each ticket
RequestTicketInfo()
RequestTicketInfo()
TicketInfo
Display(TicketInfo)
ALT
[If escalation not required]
AssignTicket()
AssignTicket()
ChangeStatus()
TicketAssigned
TicketStatus(Assigned)
[Else]
EscalateTicket()
EscalateTicket()
ChangeStatus()
TicketEscalated
TicketStatus(Escalated)
ResolveTicket()
ResolveTicket()
ChangeStatus()
TicketResolved
TicketStatus(Resolved)
Par
Display(TicketResolved)
Display(TicketResolved)
RequestFeedback()
ProvideFeedback()
CloseTicket()
ChangeStatus()
TicketClosed
Par
Display(TicketClosed)
Display(TicketClosed)

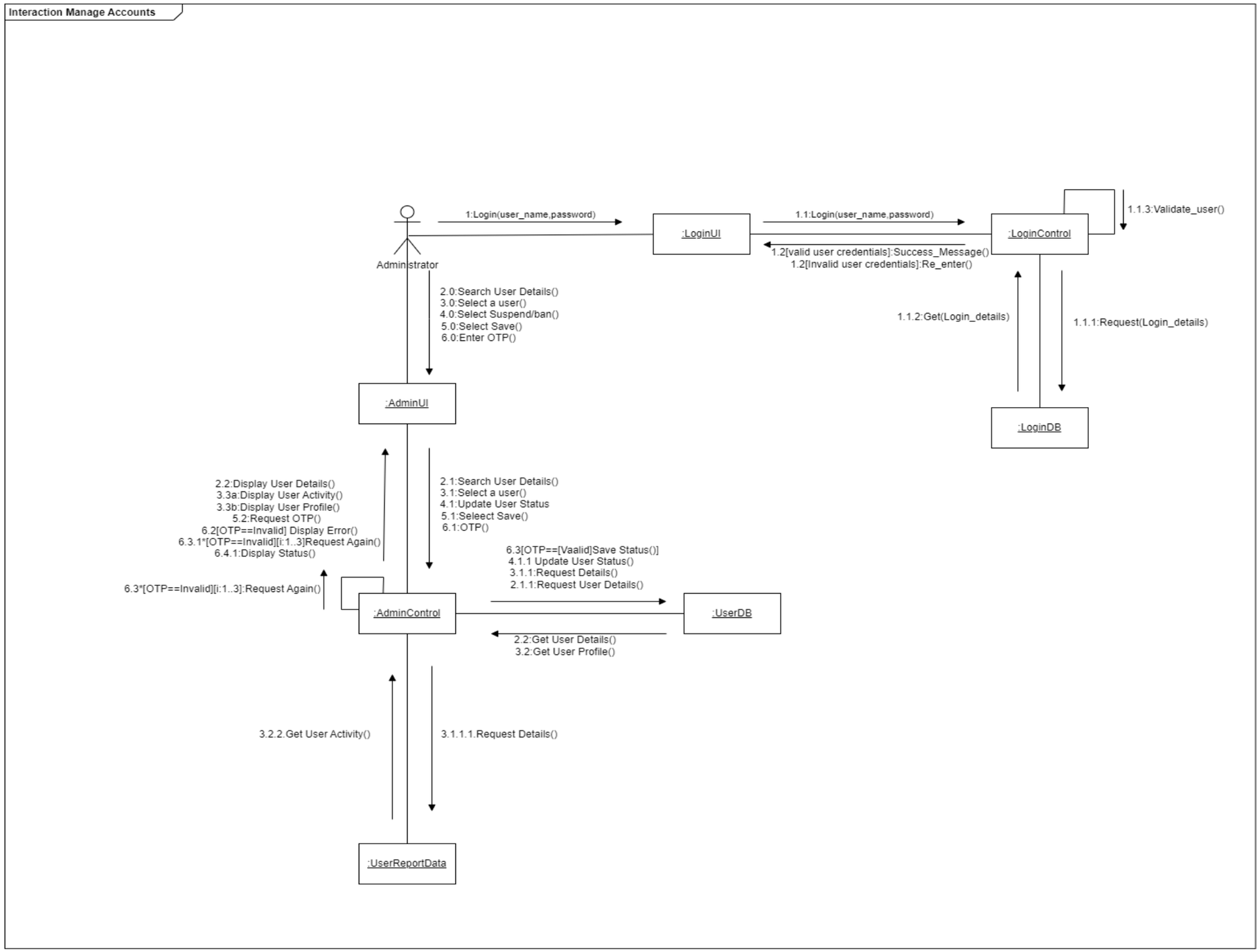

interaction CustomerSupport
1.0 :Display open tickets()
2.0 :Choose a ticket()
3.0 :Assign the ticket()
4.0 :[If escalation required]: Escalate ticket()
5.0 :Resolve ticket()
1.1 :Request open tickets()
2.1 :Request ticket Info()
3.1 :Assign the ticket()
4.1 :[If escalation required]: Escalate ticket()
5.1 :Resolve ticket()
8.0 :Close ticket()
6.0 :Dsiplay(TicketResolved)
7.0 :Request feedback()
Support Agent
:TicketsUI()
:TicketsController()
:CustomerUI()
1.1.1.1.1 :Display open tickets()
2.1.1.1.1 :Display ticket Info()
3.1.1.1.1 :Ticket status(Assigned)
4.1.1.1.1 :[If escalation required]: Ticket status(Escalated)
5.1.1.1.1 :Ticket status(Resolved)
8.1.1.1 :Ticket status(Closed)
7.1 :Provide feedback()
1.1.1 :Check open tickets()
2.1.1 :Request ticket Info()
3.1.1 :Change status()
4.1.1 :[If escalation required] :Change status()
5.1.1 :Change status()
8.1 :Change status()
1.1.1.1 :Open tickets
2.1.1.1 :Ticket Info
3.1.1.1 :Ticket status(Assigned)
4.1.1.1 :[If escalation required]: Ticket status(Escalated)
5.1.1.1 :Ticket status(Resolved)
8.1.1.1 :Ticket status(Closed)
Customer
:TicketsDB()

3. IT22278708 Alwis L.W.R.T

Number	3	
Name	Manage user accounts	
Summary	Administrator logs in to the system to manage and maintain user accounts	
Priority	Medium	
Pre-conditions	The administrator is logged into the system with appropriate privileges.	
Post-conditions	Providing a secure and inviting environment to all users.	
Primary Actors	Administrator	
Trigger	An administrator identifies a user account needing attention and takes action.	
Main Scenario	**Step**	**Action**
	1	The administrator logs in to the system.
	2	The administrator navigates to the administrator dashboard.
	3	**The administrator presents a list of user accounts.**
	4	The administrator opens a user account to see more details.
	5	The administrator identifies an account needing attention due to inappropriate behavior take actions.
	6	The system sends an email to the user regarding the action taken.
	7	The administrator saves all changes
	8	System sends OTP for verification before saving changes on system.
	9	The administrator enters the OTP.
	10	The administrator logs out of the system.
Extensions	**Step**	**Branching Action**
	1a	If the administrator enters an incorrect username or password, display an error message.
	1b	**The system asks for credentials again.**
	5a	The administrator sends a warning message to the user
	5b	The administrator suspended the account temporarily

	5c	The administrator permanently bans the user
	9a	The administrator enters an incorrect OTP
	9b	The system prompts for re-entry of the OTP

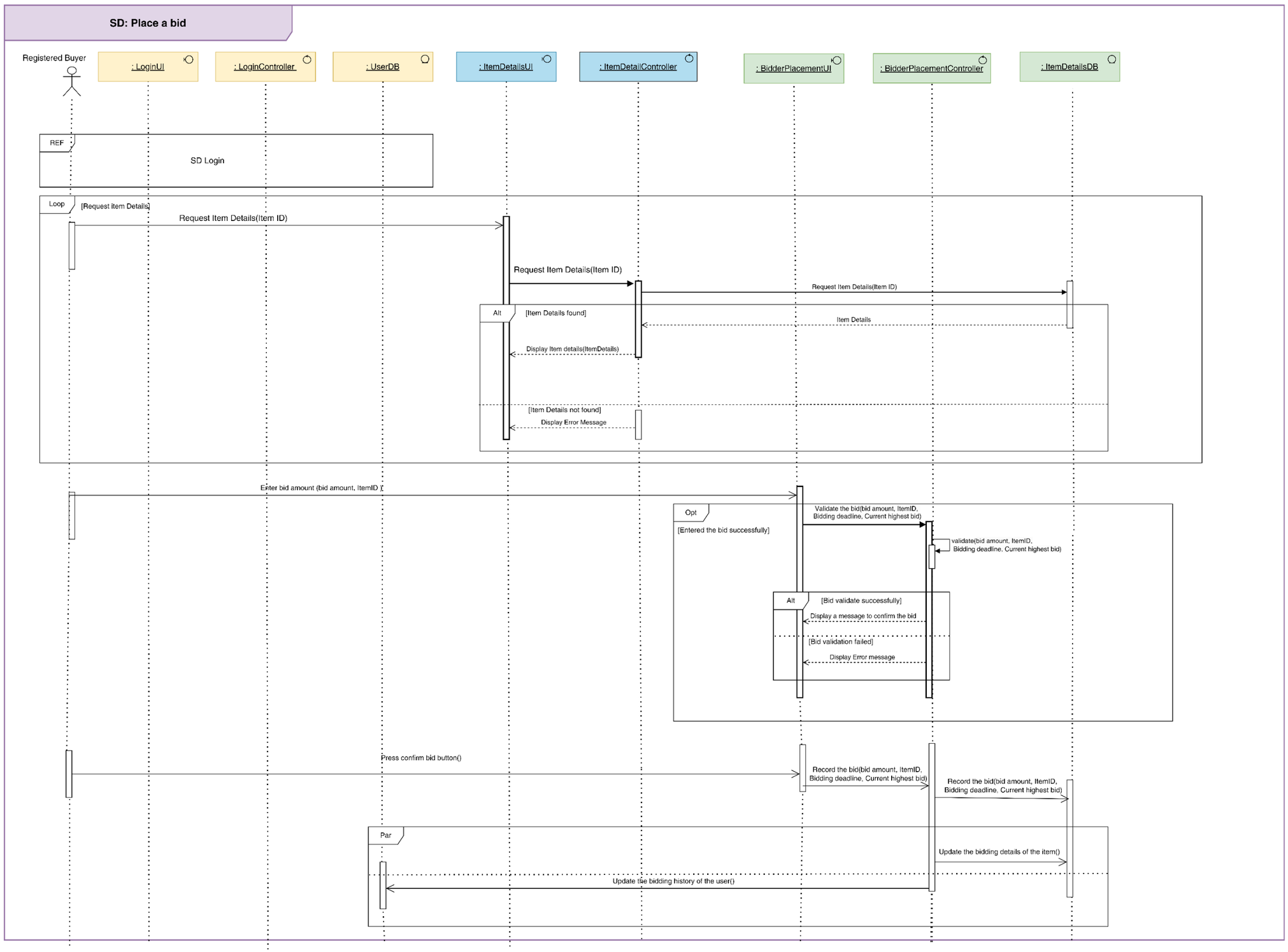

sd Manage Accounts
Administrator
:LoginUI
:LoginControl
:LoginDB
:AdminUI
:AdminControl
:UserDB
:UserReportData
Ref
Login
Search User Deatals()
Search User Deatals()
Request User Details()
User Details()
Display User Details()
Select a user()
Select a user()
Request Detailis()
Request Details()
par
User Activity()
return
User Profile()
Display User Profile()
Select suspend/ban()
Update User Status()
Update User Status()
Select Save()
Select Save()
Request OTP()
Enter OTP()
alt
[OTP invalid]
loop(1,3)
[Invalid OTP]
OTP()
DisplayError()
Request Again OTP()
Request OTP()
[OTP Valid]
Save Status()
Display Status()

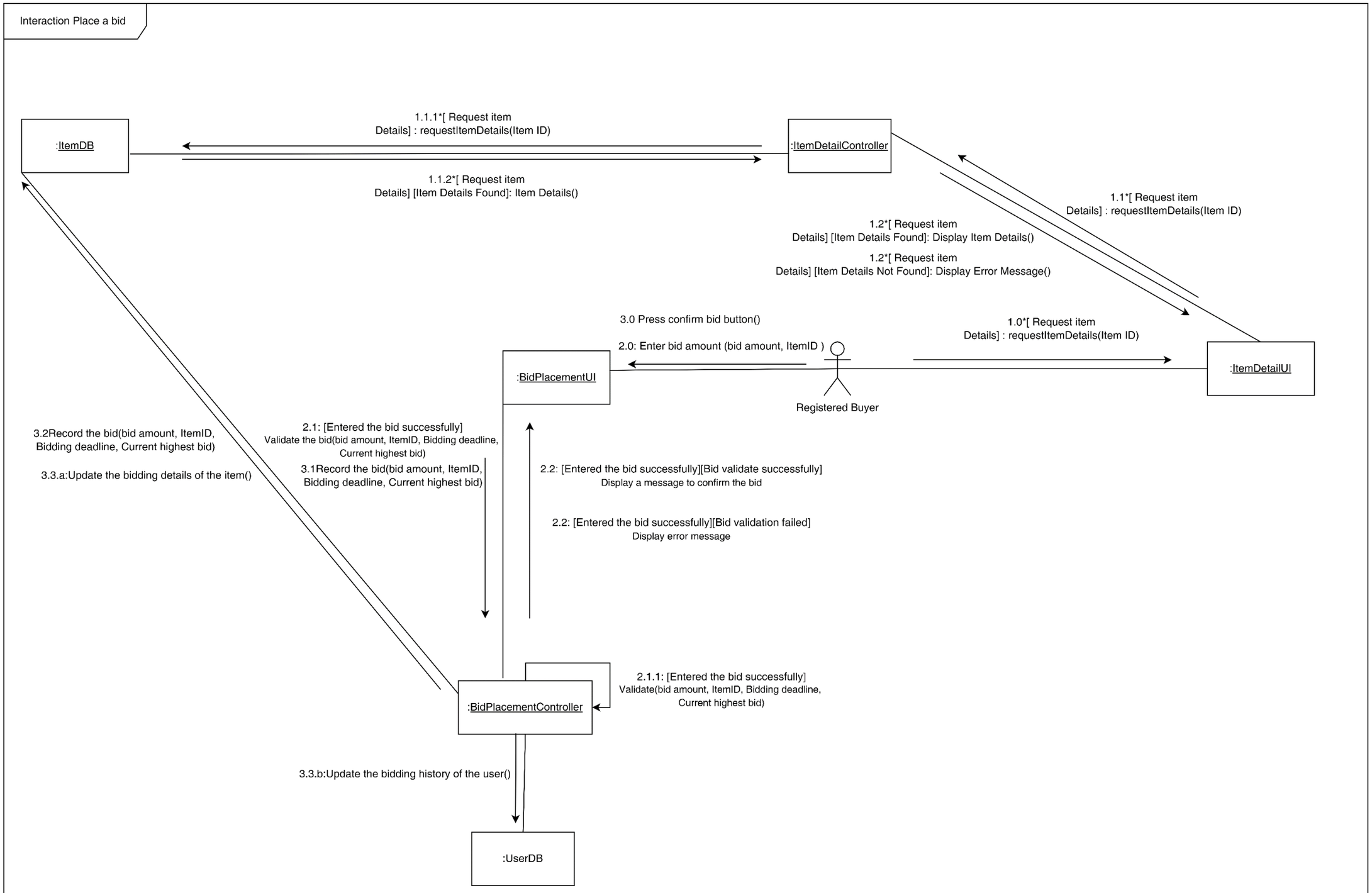

Interaction Manage Accounts
Administrator
1:Login(user_name,password)
:LoginUI
1.1:Login(user_name,password)
:LoginControl
1.1.3:Validate_user()
1.2[valid user credentials]:Success_Message()
1.2[Invalid user credentials]:Re_enter()
2.0:Search User Details()
3.0:Select a user()
4.0:Select Suspend/ban()
5.0:Select Save()
6.0:Enter OTP()
1.1.2:Get(Login_details)
1.1.1:Request(Login_details)
:LoginDB
:AdminUI
2.2:Display User Details()
3.3a:Display User Activity()
3.3b:Display User Profile()
5.2:Request OTP()
6.2[OTP==Invalid] Display Error()
6.3.1*[OTP==Invalid][i:1..3]Request Again()
6.4.1:Display Status()
2.1:Search User Details()
3.1:Select a user()
4.1:Update User Status
5.1:Seleect Save()
6.1:OTP()
6.3[OTP==[Vaalid]Save Status()]
4.1.1 Update User Status()
3.1.1:Request Details()
2.1.1:Request User Details()
6.3*[OTP==Invalid][i:1..3]:Request Again()
:AdminControl
:UserDB
2.2:Get User Details()
3.2:Get User Profile()
3.2.2.Get User Activity()
3.1.1.1.Request Details()
:UserReportData

4. IT22285010 Bamunuge M.M

Number	4	
Name	Place bid	
Summary	Registered buyer places a bid on an item in the online bidding system.	
Priority	High	
Pre-conditions	The registered buyer must be logged into their registered account. The registered buyer must have navigated to the page of the item they want to bid on.	
Post-conditions	The bid is recorded in the system's bid history for the item. The registered buyer receives confirmation of their bid placement. The current highest bid for the item is updated if the registered buyer's bid surpasses the previous highest bid.	
Primary Actors	Registered Buyer	
Trigger	The registered buyer navigates to the page of the item they wish to bid on and initiates the bid placement process.	
Main Scenario	**Step**	**Action**
	1	The registered buyer navigates to the item they are interested in bidding on.
	2	The system displays the details of the item including the current highest bid and bidding deadline.
	3	The registered buyer enters their bid amount into the bid field.
	4	The system validates the bid amount
	5	If the bid amount is valid, the registered buyer confirms their bid
	6	The system updates the bid history for the item with the registered buyer's bid.
	7	The system displays a confirmation message indicating that the bid has been successfully placed
Extensions	**Step**	**Branching Action**
	3a	If fails to place the bid, reload the page.

	4a	If the bid amount is invalid (lower than the current highest bid, not in the allowed bidding increments, or bidding deadline has passed), the system displays an error message and prompts the user to enter a valid bid amount.
	5a	If the registered buyer decides not to confirm the bid after entering the bid amount, they can choose to cancel the bid
	7a	If a system error occurs during the bid placement process (e.g., database connection failure, server timeout), the system displays an error message indicating the issue.


SD: Place a bid
Registered Buyer
: LoginUI
: LoginController
: UserDB
: ItemDetailsUI
: ItemDetailController
: BidderPlacementUI
: BidderPlacementController
: ItemDetailsDB
REF
SD Login
Loop
[Request Item Details]
Request Item Details(Item ID)
Request Item Details(Item ID)
Request Item Details(Item ID)
Alt
[Item Details found]
Item Details
Display Item details(ItemDetails)
[Item Details not found]
Display Error Message
Enter bid amount (bid amount, itemID)
Opt
[Entered the bid successfully]
Validate the bid(bid amount, ItemID, Bidding deadline, Current highest bid)
validate(bid amount, ItemID, Bidding deadline, Current highest bid)
Alt
[Bid validate successfully]
Display a message to confirm the bid
[Bid validation failed]
Display Error message
Press confirm bid button()
Record the bid(bid amount, ItemID, Bidding deadline, Current highest bid)
Record the bid(bid amount, ItemID, Bidding deadline, Current highest bid)
Par
Update the bidding details of the item()
Update the bidding history of the user()


Interaction Place a bid
:ItemDB
1.1.1*[Request item
Details] : requestItemDetails(Item ID)
1.1.2*[Request item
Details] [Item Details Found]: Item Details()
:ItemDetailController
1.1*[Request item
Details] : requestItemDetails(Item ID)
1.2*[Request item
Details] [Item Details Found]: Display Item Details()
1.2*[Request item
Details] [Item Details Not Found]: Display Error Message()
3.0 Press confirm bid button()
2.0: Enter bid amount (bid amount, ItemID)
1.0*[Request item
Details] : requestItemDetails(Item ID)
:BidPlacementUI
:ItemDetailUI
Registered Buyer
3.2Record the bid(bid amount, ItemID,
Bidding deadline, Current highest bid)
3.3.a:Update the bidding details of the item()
2.1: [Entered the bid successfully]
Validate the bid(bid amount, ItemID, Bidding deadline,
Current highest bid)
3.1Record the bid(bid amount, ItemID,
Bidding deadline, Current highest bid)
2.2: [Entered the bid successfully][Bid validate successfully]
Display a message to confirm the bid
2.2: [Entered the bid successfully][Bid validation failed]
Display error message
2.1.1: [Entered the bid successfully]
Validate(bid amount, ItemID, Bidding deadline,
Current highest bid)
:BidPlacementController
3.3.b:Update the bidding history of the user()
:UserDB

Reg. No	Name	Function Name
IT22190598	Gamalath K.H	Confirm the winning bidder's payment
IT22117250	SILVA A.S.N	Support Customer
IT22278708	Alwis L.W.R.T	Manage user accounts
IT22285010	Bamunuge M.M	Place bid