

Iteration 2

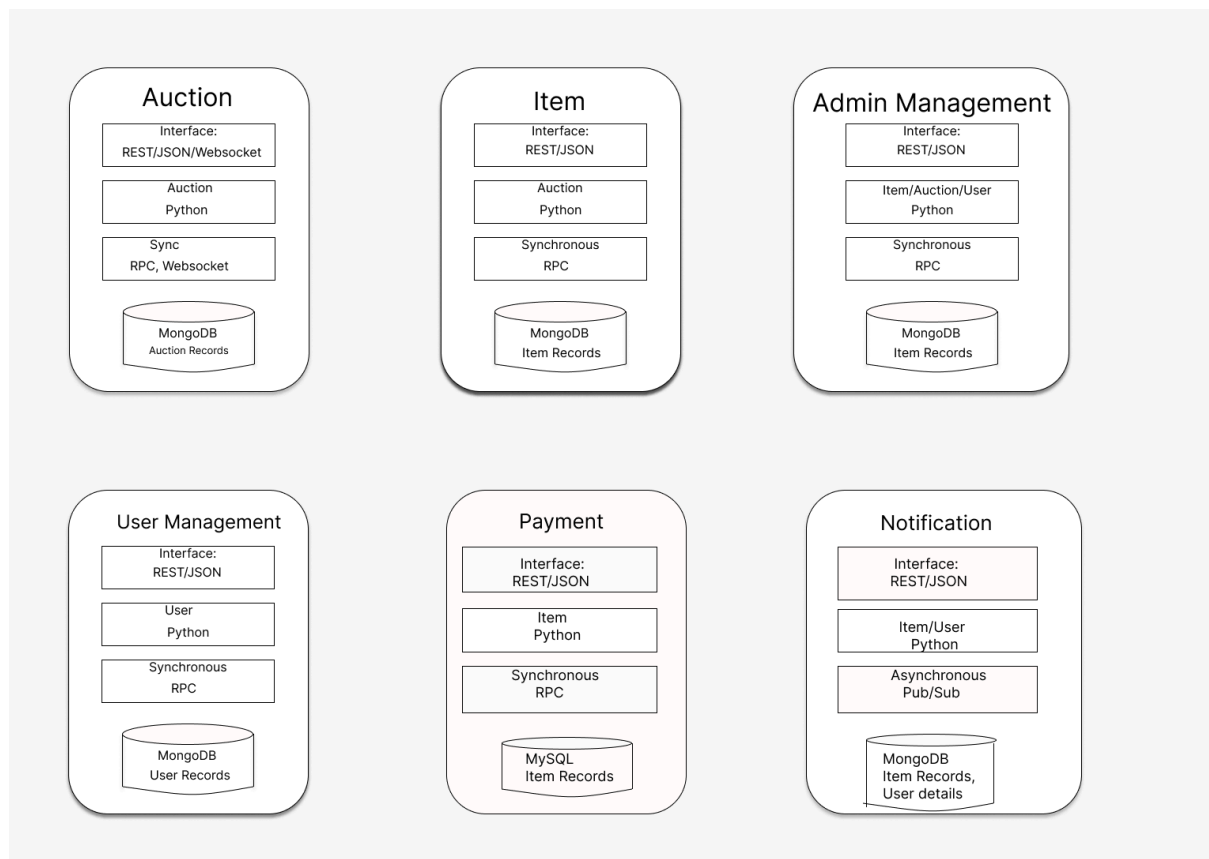
Team Name : HawkDev

Team Members -

- Yixin Liu
- Doondy Sai Vyshnavi Koyya
- Lihaowen Zhu
- Rui Qiu

Deliverable 1

<https://www.figma.com/design/RjHUdWdsH8vXLVAvSnZTdZ/MPCS51205-Iteration2-Deliverable1?node-id=0-1&t=BJSzEyOoOWuSJSsd-1>



- Auction
 - Interface: REST/JSON
 - Sync/Async, protocol: Synchronous, RPC, Websocket
 - Programming language: Python
 - Related contexts: Payment, Notification, Item
 - Tech stacks: MongoDB, Websocket
- User Management

- Interface: REST/JSON
 - Sync/Async, protocol: Synchronous, RPC
 - Programming language: Python
 - Related contexts: Notification, Admin
 - Tech stacks: MySQL
- Payment
 - Interface: REST/JSON
 - Sync/Async, protocol: Synchronous, RPC
 - Programming language: Python
 - Related contexts: Item
 - Tech stacks: MySQL
- Notification
 - Interface: REST/JSON
 - Sync/Async, protocol: Asynchronous, Pub/Sub
 - Programming language: Python
 - Related contexts: Item/User
 - Tech stacks: MongoDB
- Item
 - Interface: REST/JSON
 - Sync/Async, protocol: Synchronous, RPC
 - Programming language: Python
 - Related contexts: Auction
 - Tech stacks (Data store): MongoDB
- Admin Management
 - Interface: REST/JSON
 - Sync/Async, protocol: Synchronous, RPC; Asynchronous, Pub/Sub
 - Programming language: Python
 - Related contexts: Item/Auction/User
 - Tech stacks (Data store): MongoDB

Deliverable 2:

Microservice Architecture Catalog

Microservice	Service Operation
Auction	<ul style="list-style-type: none">• listItemsForAuction• placeBid• attachNotificationOnCondition• createAuction• updateAuction
User	<ul style="list-style-type: none">• buyItem• checkoutCart• addItemToWatchList• removeItemFromWatchList• createUser• updateUser• suspendUser
Payment	<ul style="list-style-type: none">• ProcessPayment• CompletePayment
Notification	<ul style="list-style-type: none">• SendWatchlistNotification• scheduleNotification

Item	<ul style="list-style-type: none"> • UpdateItemInfo • flagItem • deleteItem • createCategory
Admin	<ul style="list-style-type: none"> • RemoveUser • AddCategory • UpdateCategory • RemoveCategory • SearchFlaggedItem • SearchAuction • ExamineClosedAuction • RespondEmail

Service Operation Analysis

Service	Operation	Business Capability	Actor	UseCase or User Story
Auction	listItemsForAuction	Returns items available for auction that match the selected category or keyword filters.	buyers	Buyers check all items that are on auction
Auction	placeBid	Allows users to place bids on an active auction and increments the current bid if it's higher.	buyers	Place a bid for an auction
Auction	attachNotificationOnCondition	Send notification or alerts to buyer/ seller on certain condition	sellers/ buyers	Alert users when auction nearly ends

Auction	createAuction	Initialize a new auction for an item with specified details such as bid window and min/ max price	sellers	Create an auction for an item
Auction	updateAuction	Update info of an auction	sellers	Change minimum bid price
User	buyItem	Buy an item and place it into user's cart	buyers	Buy and item
User	checkoutCart	Checkout all items in cart	buyers	Checkout all items in cart
User	addItemToWatchList	Adds items to a user's watchlist with specified criteria, such as maximum bid price.	buyers	Add an item to watchlist since the buyer has a willing to buy
User	removeItemFromWatchList	Remove items to a user's watchlist	buyers	
User	createUser	Create a user	All users/ Admin	Create a user
User	updateUser	Update user info	All users/ Admin	Update user info
User	suspendUser	Suspend a user	All users/ Admin	Suspend a user when the user violates regulations
Payment	ProcessPayment()	General payment processing for auctions.		

Payment	CompletePayment()	Confirms that the payment has been successfully processed.		
Notification	SendWatchlistNotification()	Sends an email notification when an item on the user's watchlist matches their criteria.		
Notification	ScheduleNotification()	Schedules the notification to be sent to the user		
Item	UpdateItemInfo()	Update the information of an item.		
Admin Management	RemoveUser()	Remove a user account from the		
Admin Management	AddCategory()	Add an item category to the auction site.		
Admin Management	UpdateCategory()	Update the information of an item category.		
Admin Management	RemoveCategory()	Remove an item category from the auction site.		
Admin Management	SearchFlaggedItem()	Search and view all items flagged by users.		

Admin Management	SearchAuction()	Search and view all in-progress auctions sorted by the soonest ending time.		
Admin Management	ExamineClosedAuction()	Examine the metrics for closed auctions in a given timeframe.		
Admin Management	RespondEmail()	Examine customer emails and respond to them.		

Service Operation Decomposition:

Microservice: Auction	
Service Operation: listItemsForAuction	
Return Value: items	Communication Mechanism Request/ Reply
Responsibilities: Returns items available for auction that match the selected category or keyword filters.	Communication Model Synchronous
Parameters: categoryID, keywordFilters	Pre- and Post- conditions (including latency issues)

--	--

Microservice: Auction	
Service Operation: placeBid	
Return Value: Json of operation results	Communication Mechanism Request/ Reply
Responsibilities: Allows users to place bids on an active auction and increments the current bid if it's higher	Communication Model Synchronous
Parameters: auctionID, userID, bidAmount	Pre- and Post- conditions (including latency issues)

Microservice: Auction	
Service Operation: attachNotificationOnCondition	
Return Value: Json of operation results	Communication Mechanism Request/ Reply
Responsibilities: Send notification or alerts to buyer/ seller on certain condition	Communication Model Synchronous
Parameters: Message, condition	Pre- and Post- conditions (including latency issues) Notification microservice must be running to schedule, send notifications

Microservice: Auction	
Service Operation: createAuction	
Return Value:	Communication Mechanism

JSON of auction details	Request/ Reply
Responsibilities: Initialize a new auction for an item with specified details.	Communication Model Synchronous
Parameters: itemID, startTime, endTime, otherCriteriaForAuction	Pre- and Post- conditions (including latency issues)

Microservice: Auction	
Service Operation: updateAuction	
Return Value: JSON of auction details	Communication Mechanism Request/ Reply
Responsibilities: Update auction	Communication Model Synchronous
Parameters: auctionId, details	Pre- and Post- conditions (including latency issues)

Microservice: User	
Service Operation: buyItem	
Return Value: Json of item info	Communication Mechanism Request/ Reply
Responsibilities: Add to item user's cart after checking criteria of item and auction	Communication Model Synchronous
Parameters:	Pre- and Post- conditions (including latency issues)

ItemId, userId	
----------------	--

Microservice: User	
Service Operation: checkoutCart	
Return Value: Json of operation status	Communication Mechanism Request/ Reply
Responsibilities: Proceeds with the checkout for all items in the user's cart.	Communication Model Synchronous
Parameters: userId	Pre- and Post- conditions (including latency issues)

Microservice: User	
Service Operation: addItemToWatchList	
Return Value: Json of operation results	Communication Mechanism Request/ Reply
Responsibilities: Adds items to a user's watchlist with specified criteria, such as maximum bid price.	Communication Model Synchronous
Parameters: userId, itemID, criteria	Pre- and Post- conditions (including latency issues)

Microservice: User	
Service Operation: createUser/ updateUser/ SuspendUser	
Return Value: Json of operation results	Communication Mechanism Request/ Reply

Responsibilities: Manage user account status	Communication Model Synchronous
Parameters: userId, (userInfo)	Pre- and Post- conditions (including latency issues)

Microservice: Payment	
Service Operation: ProcessPayment();	
Return Value: This operation returns a success or failure with paymentID	Communication Mechanism
Responsibilities: This operation handles payment initiation, processes the transaction with payment gateways and updates payment status in database	Communication Model
Parameters: Payment_id Amount	Pre- and Post- conditions (including latency issues)

Payment_method Item_id user_id	
--------------------------------------	--

Microservice: Payment	
Service Operation: CompletePayment();	
Return Value: Returns payment confirmation ID and status code of success or failure	Communication Mechanism
Responsibilities: Confirms the user about the payment processed	Communication Model
Parameters: Payment_id Transaction_status user_id	Pre- and Post- conditions (including latency issues)

Microservice: Notification	
Service Operation: SendWatchlistNotification()	
Return Value: Returns a success or failure message with the message, notification id, item details and user_id	Communication Mechanism
Responsibilities: This operation checks if any items on the user's watchlist match the criteria they have set and triggers notifications accordingly.	Communication Model

Parameters: user_id watchlist_id Item_id Item_name Price Category User_email	Pre- and Post- conditions (including latency issues)
---	--

Microservice: Notification	
Service Operation: ScheduleNotification()	
Return Value: Returns a success or failure message with the message, notification id, timestamp	Communication Mechanism
Responsibilities: Schedule the notification to trigger at a specified time or upon a certain condition (e.g., when a specific item status changes).	Communication Model
Parameters: user_id watchlist_id Item_id Item_name Price Category User_email scheduled_time	Pre- and Post- conditions (including latency issues)

Microservice: Item	
Service Operation: UpdateItemInfo()	
Return Value:	Communication Mechanism

Returns a success or failure message with the updated item details	
<p>Responsibilities:</p> <p>This operation updates the information of an item, including quantity, description, shipping costs, and Buy Now feature.</p>	Communication Model
<p>Parameters:</p> <p>user_id item_id item_name update_field update_data</p>	Pre- and Post- conditions (including latency issues)

Microservice: Admin Management	
Service Operation: StopAuction()	
<p>Return Value:</p> <p>Returns a success or failure message about the operation of stopping an auction.</p>	Communication Mechanism
<p>Responsibilities:</p> <p>This operation stops an in-progress auction earlier than its end time. This operation can only be called by an admin.</p>	Communication Model
<p>Parameters:</p> <p>admin_id auction_id</p>	Pre- and Post- conditions (including latency issues)

Microservice: Admin Management	
Service Operation: RemoveUser()	

<p>Return Value:</p> <p>Returns a success or failure message with the details of the detailed user.</p>	Communication Mechanism
<p>Responsibilities:</p> <p>This operation removes or blocks a user chosen by an admin from the auction site.</p>	Communication Model
<p>Parameters:</p> <p>admin_id user_id</p>	Pre- and Post- conditions (including latency issues)

Microservice: Admin Management	
Service Operation: AddCategory()	
<p>Return Value:</p> <p>Returns a success or failure message with the details of the newly added category.</p>	Communication Mechanism
<p>Responsibilities:</p> <p>This operation adds a new category to the auction site if it doesn't exist.</p>	Communication Model
<p>Parameters:</p> <p>admin_id category_info</p>	Pre- and Post- conditions (including latency issues)

Microservice: Admin Management	
Service Operation: UpdateCategory()	
Return Value:	Communication Mechanism

Returns a success or failure message with the details of the updated category.	
<p>Responsibilities:</p> <p>This operation updates the details of a category if the category exists in the auction site.</p>	Communication Model
<p>Parameters:</p> <p>admin_id category_id category_info</p>	Pre- and Post- conditions (including latency issues)

Microservice: Admin Management	
Service Operation: RemoveCategory()	
<p>Return Value:</p> <p>Returns a success or failure message with the details of the removed category.</p>	Communication Mechanism
<p>Responsibilities:</p> <p>This operation removes a category from the auction site if the category exists.</p>	Communication Model
<p>Parameters:</p> <p>admin_id category_id</p>	Pre- and Post- conditions (including latency issues)

Microservice: Admin Management	
Service Operation: SearchFlaggedItem()	
<p>Return Value:</p> <p>Returns a list of items flagged by any users.</p>	Communication Mechanism

Responsibilities: This operation finds a list of items flagged by users for admins to analyze.	Communication Model
Parameters: admin_id	Pre- and Post- conditions (including latency issues)

Microservice: Admin Management	
Service Operation: SearchAuction()	
Return Value: Returns a list of auctions currently in progress.	Communication Mechanism
Responsibilities: This operation finds all auctions currently in progress, which are sorted by ending time ascendingly.	Communication Model
Parameters: admin_id	Pre- and Post- conditions (including latency issues)

Microservice: Admin Management	
Service Operation: ExaminClosedAuction()	
Return Value: Returns a list of closed auctions with their details.	Communication Mechanism
Responsibilities:	Communication Model

This operation updates the details of a category if the category exists in the auction site.	
Parameters: admin_id category_id category_info	Pre- and Post- conditions (including latency issues)

Microservice: Admin Management	
Service Operation: RespondEmail()	
Return Value: Returns a success or failure message with the details of the email and response.	Communication Mechanism
Responsibilities: This operation allows an admin to respond to a certain email from the customer support.	Communication Model
Parameters: admin_id email_id email_address response	Pre- and Post- conditions (including latency issues)

Technology Stack Decomposition:

Microservice: Auction	
Programming Languages: Python	Stack Description REST/JSON endpoints for creating, updating auctions and for bidding management. Websocket interface can be used for live-updating of auction information. Datastore is MongoDB.
Interface Description: External user accesses through HTTP, RPC. External system interfaces in REST.	
Business Logic Description: ActionId, ItemId	Data storage Description A key-value store can handle the transaction and auction data.

Microservice: User	
Programming Languages: Python	Stack Description REST/JSON endpoints for creating, updating user info and for shopping cart/ wishlist management. Datastore is MySQL.
Interface Description: External user accesses through HTTP, RPC. External system interfaces in REST.	
Business Logic Description: UserId	Data storage Description User has strong relations to other microservices, thus a relational database is required.

Microservice: Payment	
Programming Languages: Python	Stack Description
Interface Description: REST/JSON endpoints for initiating and confirming payments.	

Business Logic Description: ProcessPayment CompletePayment	Data storage Description
--	--------------------------

Microservice: Notification	
Programming Languages: Python	Stack Description
Interface Description: REST/JSON to accept notification requests. Pub/Sub model using RabbitMQ to send notifications when certain item match in watchlist event is triggered	
Business Logic Description: SendWatchlist()	Data storage Description

Microservice: Item	
Programming Languages: Python	Stack Description
Interface Description: REST/JSON endpoints to update details of items.	
Business Logic Description: UpdateItemInfo()	Data storage Description:

Microservice: Admin	
Programming Languages:	Stack Description

Python	
<p>Interface Description:</p> <p>REST/JSON endpoints to manage users, auctions and categories, and search items and auctions.</p> <p>Pub/Sub model using RabbitMQ to respond to emails in customer support.</p>	
<p>Business Logic Description:</p> <p>StopAuction() RemoveUser() AddCategory() UpdateCategory() RemoveCategory() SearchFlaggedItem() SearchAuction() ExamineClosedAuction() RespondEmail()</p>	Data storage Description

Event Model:

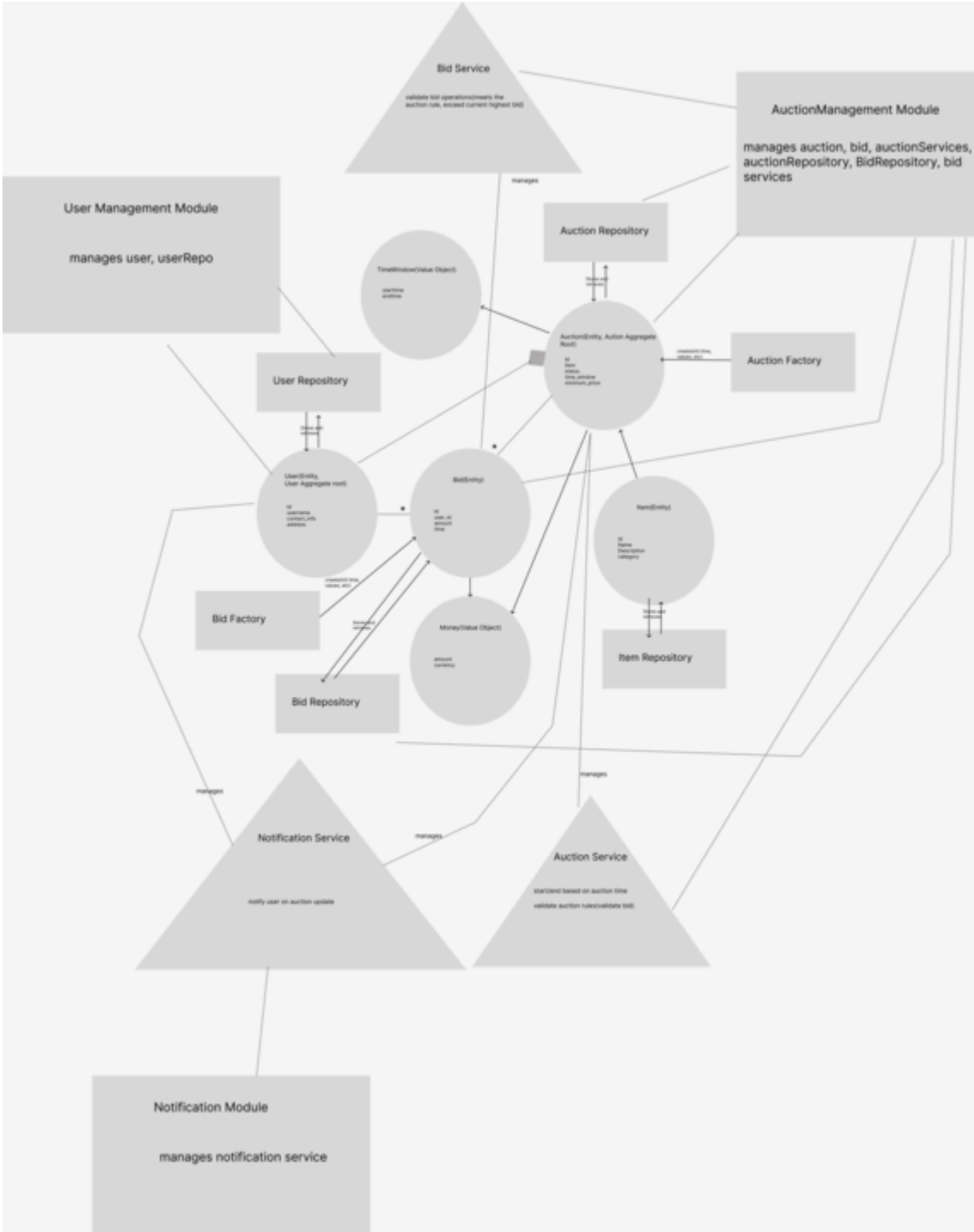
Asynchronous Event	Use Cases	Interested Parties	Associated Protocols
SendWatchlistNotification	Notify User	User, email	REST/JSON and AMQP
ScheduleNotification	Schedule notification for future events, such as item bid updates or watchlist alerts	User, email	REST/JSON and AMQP
Payment Confirmation	Notify user when a payment is successful	Email, User	AMQP
UpdateItemInfo	Item Detail Update	User, Item, Auction	REST/JSON
StopAuction	Suspend Auction	Auction,	REST/JSON

		User, Item, Admin	
RemoveUser	Remove/Block User	User, Admin	REST/JSON
AddCategory	Add New Category	Item, Admin	REST/JSON
UpdateCategory	Update Category Detail	Item, Admin	REST/JSON
RemoveCategory	Remove Category	Item, Admin	REST/JSON
SearchFlaggedItem	Search Flagged Item	Item, Admin	REST/JSON
SearchAuction	Search In-progress Auction	Auction, Admin	REST/JSON
ExamineClosedAuction	View Closed Auction	Auction, Admin	REST/JSON
RespondEmail	Respond Email	Admin, User	AMQP

Deliverable 3

Auction context Model

Figma: <https://www.figma.com/design/Gsk5cutvNshIEJV4GA631P/Untitled?m=dev&t=yr5eQ3QlPPGyCqKh-1>



Deliverable 4 ADR

See markdown files in the tarball