CardHub: Use Case Description

Levente Szabo - 618088

Use Case Number: 1

Name	**User Account Management**

Brief description	**This use case allows customers to register for an account and login to access the CardHub system**

Actors	**Customer**

Preconditions
For registration: None. For login: User must have a registered and verified account.

Flows of Events:

1. Basic Flows

1.1.0 Register Account

Step	User Actions	System Actions
1	The customer goes to CardHub website and clicks "Register" button	The system shows registration form with fields for email, password (enter twice), first name, last name, and phone number (optional)
2	The customer fills out the form, accepts Terms of Service, and clicks "Create Account"	The system checks the input: email format is correct and not already used, password is strong enough (8+ characters with uppercase, lowercase, and number), passwords match, required fields are filled. The system creates user account with role "Customer", securely stores the password, creates a verification code, and sends verification email. The system shows: "Account created! Please check your email to verify."
3	The customer checks email and clicks verification link	The system checks the code, marks email as verified, and shows: "Email verified! You can now log in."

Postconditions

Postconditions
Customer account is created and saved in database. Customer can log in.

Business Rules
Email addresses must be unique. Passwords must be 8+ characters with uppercase, lowercase, and number. Passwords are securely stored. Email verification required. Verification links expire after 24 hours.

1.2.0 Login

Step	User Actions	System Actions
1	The user goes to CardHub and clicks "Login"	The system shows login form with email, password, "Remember me" checkbox, and "Forgot password?" link
2	The user enters email and password and clicks "Login"	The system checks email format and fields are not empty. System looks up user by email and verifies password. System checks email is verified and account not disabled.
3		The system creates user session, records the login time, and redirects based on role: Customer → Customer dashboard, Staff → Staff dashboard, Administrator → Admin dashboard. System shows welcome message with user's name.

Postconditions
User is logged in with active session.

Business Rules
Maximum 5 failed login attempts per 15 minutes per internet address. Sessions expire after 24 hours of not using the system. Error messages don't reveal which accounts exist. All login attempts are recorded.

Use Case Number: 2

Name	**Card Inventory Management**

Brief description	**This use case allows store staff and administrators to add and update trading cards in the store's inventory**

Actors	**Store Staff, Store Administrator**

Preconditions
User must be logged in with Staff or Administrator role. Card reference database available.

Flows of Events:

1. Basic Flows

1.1.0 Add Card to Inventory

Step	User Actions	System Actions
1	The staff clicks "Add to Inventory" button	The system shows card entry form
2	The staff enters card details: game, card name (with search suggestions), set name, condition (Mint/Near-Mint/Played/Damaged), quantity, foil variant	The system searches reference database and shows suggestions with card images. System looks up selected card and gets current market price from Card Price Service.
3		The system shows card details: image, type, rarity, color, market price. System calculates buy/sell prices: Buy = Market × 60%, Sell = Market × 80%, adjusted for condition. System shows calculated prices.
4	The staff reviews prices (can change them), clicks "Add to Inventory"	The system checks all fields are filled and quantity is a positive number. System checks if this exact card (name, set, condition, foil) already exists. If exists: adds quantity to existing entry. If new: creates new entry. System saves to database and records action (staff member, time, details).
5		The system shows: "Added [quantity] × [card name] to inventory" and updates totals

Postconditions
Cards added to inventory. Inventory count updated.

Business Rules
All inventory changes are recorded with user and time for tracking. Buy/sell prices calculated from market price and store pricing rules. Condition affects price: Mint 100%, Near-Mint 90%, Played 70%, Damaged 40%. Inventory updates are safe from errors when multiple staff work at once.

1.2.0 Update Card Price

Step	User Actions	System Actions
1	The staff searches for and selects a card from inventory	The system shows the card details with current buy and sell prices
2	The staff clicks "Update Price"	The system shows price edit form with current prices filled in
3	The staff enters new buy price and/or sell price, clicks "Save"	The system checks prices are valid numbers. System updates the card prices in database and records the change (staff member, time, old prices, new prices).

Step	User Actions	System Actions
4		The system shows: "Price updated for [card name]"

Postconditions
Card price updated in inventory.

Business Rules
Only staff and administrators can update prices. Price changes are recorded. Manually changed prices stay until changed again.

Use Case Number: 3

Name	Customer Purchase
Brief description	This use case allows customers to browse cards, add to cart, and complete purchase for in-store pickup

Actors	Customer (Primary), Payment Gateway (Secondary), Email Service (Secondary)

Preconditions
For browsing: None. For checkout: Customer has items in cart, logged in or checking out as guest.

Flows of Events:

1. Basic Flows

1.1.0 Browse and Search Cards

Step	User Actions	System Actions
1	The customer goes to store section	The system shows card browsing page with search bar, filters, and featured cards
2	The customer enters search term (card name, set, keyword)	The system searches inventory database. System returns results showing card image, name, set, condition, price, quantity, sorted by how well they match.
3	The customer applies filters: game type, rarity, condition, price range	The system updates results based on filters right away
4	The customer clicks card to view details	The system shows detailed info: full image, description, all conditions/quantities, market price, store price

Postconditions
Customer can see available cards and details.

Business Rules

Business Rules
Search returns results in less than 1 second. Only cards with quantity greater than 0 are shown as available. Prices shown are current sell prices.

1.2.0 Add to Cart

Step	User Actions	System Actions
1	The customer views card detail, selects condition and quantity	The system checks requested quantity is available
2	The customer clicks "Add to Cart"	The system adds card with condition and quantity to cart. System shows: "Added to cart" and updates cart icon with item count. System updates cart total.

Postconditions
Card is in shopping cart.

Business Rules
Cart items saved per session (guests) or per account (logged-in users). Adding duplicate card increases quantity. Cart doesn't hold inventory until checkout starts.

1.3.0 Checkout and Pay

Step	User Actions	System Actions
1	The customer clicks "Checkout" from cart	The system checks all cart items still in stock and prices current. System shows checkout page: order summary (items, quantities, prices), subtotal, tax, total.
2	The customer enters contact info: email, phone, name	The system checks required fields are filled
3		The system shows store address and pickup hours. System shows estimated ready time (example: "Ready in 1 hour").
4	The customer clicks "Continue to Payment"	The system holds the cart items for 15 minutes so no one else can buy them. System shows payment options: Credit/Debit Card or PayPal.
5	The customer selects "Credit/Debit Card"	The system shows secure payment form from Stripe

Step	User Actions	System Actions
6	The customer enters card number, expiration, security code, billing zip code, clicks "Pay Now"	The system checks format and sends payment request to Payment Gateway with total. Payment Gateway checks card, processes payment, returns success.
7		The system creates order record: order number, customer info, items, payment details (last 4 digits only), status "Pending pickup", time. System removes items from inventory and releases holds. System sends confirmation email with order number, items, total, pickup instructions.
8		The system shows order confirmation: order number, success message, estimated pickup time. If logged in: adds order to history, clears cart.

Postconditions
Order completed, payment processed, items removed from inventory, confirmation sent. Ready for in-store pickup.

Business Rules
Never save full credit card numbers (payment security rules). Payment uses secure method through gateway. Secure connection required. Item holds expire after 15 minutes. Inventory updates are safe from errors. All transactions recorded. All orders are in-store pickup only.

Use Case Number: 4

Name	**Sell Card to Store**

Brief description	**This use case allows customers to get instant price quotes for their cards and sell them to the store for cash**

Actors	**Customer, Store Staff**

Preconditions
Customer logged in with account in good standing. For completing sale: Staff logged in.

Flows of Events:

1. Basic Flows

1.1.0 Get Price Quote and Create Sell Transaction

Step	User Actions	System Actions

Step	User Actions	System Actions
1	The customer goes to "Sell to Store" and clicks "Get Price Quote"	The system shows card search form
2	The customer searches for the card they want to sell	The system shows search suggestions with card images
3	The customer selects the card and enters condition and quantity	The system looks up market price from Card Price Service. System calculates store buy price (60% of market) adjusted for condition. System shows: "We will pay $XX cash for [quantity] × [card name] in [condition] condition"
4	The customer clicks "Accept Offer"	The system creates sell transaction record: transaction number, customer, card details, price offered, status "Pending In-Store Verification", time. System shows: "Bring your card to the store. Transaction number: [number]"
5		The system adds transaction to customer's "My Sell Transactions" and notifies staff

Postconditions
Sell transaction created and waiting for customer to bring card to store.

Business Rules
Buy prices = percentage of market (can be changed by admin). Condition affects price: Mint 100%, Near-Mint 90%, Played 70%, Damaged 40%. Price quote valid for 48 hours. Customer can only sell one card per transaction. Payment is cash only.

1.2.0 Complete Sell Transaction

Step	User Actions	System Actions
1	The customer brings card to store with transaction number. Staff looks up transaction	The system shows transaction details: customer name, card, condition stated, price offered
2	The staff examines the physical card	
3	If card condition matches what customer stated, staff clicks "Complete Transaction"	The system creates payment record and marks "Pay Customer $XX Cash". System updates transaction status to "Completed" and records completion time and staff member.

Step	User Actions	System Actions
4		The system shows: "Transaction completed. Pay customer $XX cash."
5	The staff pays the customer cash and adds the card to inventory	The system follows Add Card to Inventory use case (UC 2.1.0)

Postconditions
Sell transaction completed, customer paid cash, card added to inventory.

Business Rules
Staff must physically check card condition matches what customer stated. If condition is worse than stated, staff can refuse the transaction. Refused transactions are marked "Rejected" with reason. Payment is cash only.

Use Case Number: 5

Name	Peer-to-Peer Trading Marketplace

Brief description	This use case allows customers to post and browse trade offers for cards they want to trade with other customers

Actors	Customer

Preconditions
Customer logged in with account in good standing.

Flows of Events:

1. Basic Flows

1.1.0 Post Trade Offer

Step	User Actions	System Actions
1	The customer goes to "Community Trading" and clicks "Post Trade Offer"	The system shows trade offer form
2	The customer enters trade title (example: "Looking for Charizard cards")	
3	The customer enters description of what they have to trade and what they want	

Step	User Actions	System Actions
4	The customer optionally adds contact preferences (example: "Meet at store", "Can meet anywhere in city")	
5	The customer clicks "Post Offer"	The system checks title and description are not empty. System creates trade offer record: offer number, creator, title, description, contact preferences, status "Active", time posted. System saves to database.
6		The system shows: "Trade offer posted!" System shows the offer details with edit and delete options. System adds offer to public trade board.

Postconditions
Trade offer posted and visible on community trading board.

Business Rules
Maximum 5 active offers per user. Offers automatically expire after 60 days unless renewed. Users can edit or delete their own offers anytime.

1.2.0 Browse and Contact for Trade

Step	User Actions	System Actions
1	The customer goes to "Community Trading"	The system shows all active trade offers sorted by most recent
2	The customer can filter by keyword search	The system shows matching offers
3	The customer clicks on an offer to view details	The system shows full offer: title, description, what they have, what they want, contact preferences, poster's username, time posted
4	The customer clicks "Contact Trader"	The system shows poster's contact method (email or in-app message based on their preferences)
5	The customer sends message to discuss trade	The system delivers message to poster

Postconditions
Customer contacted poster to arrange trade.

Business Rules
All trades arranged outside the system. System is only a marketplace for posting offers. No verification or prices involved. Users meet and trade on their own terms.

Use Case Number: 6

Name	**Tournament Announcements**

Brief description	**This use case allows users to post tournament announcements and check store room availability**

Actors	**Customer**

Preconditions
User logged in. For posting tournament: Tournament room must be available for desired date/time.

Flows of Events:

1. Basic Flows

1.1.0 Post Tournament Announcement

Step	User Actions	System Actions
1	The user goes to "Tournaments" and clicks "Post Tournament"	The system shows tournament room calendar showing which dates/times are already reserved
2	The user enters tournament details: name, date, start time, end time, game, format, description/rules	
3	The user clicks "Check Room Availability"	The system checks if tournament room is available for the requested date and time. If available: shows "Room available". If not: shows "Room already reserved for this time" and highlights conflicting reservation.
4	If available, user clicks "Post Tournament"	The system checks all required fields filled and date/time in future. System creates tournament record: tournament number, organizer name, details, status "Active", time posted. System reserves the tournament room for that date/time. System saves to database.
5		The system shows: "Tournament posted! Number: [number]" System shows tournament detail page. System adds to public tournament calendar.

Postconditions
Tournament announced and tournament room reserved for that date/time.

Business Rules
Tournament room can only have one event at a time. Users can post tournaments if room is available. Entry fees and registration handled outside system by organizer. Maximum 2 active tournament posts per user.

1.2.0 View Tournament Announcements

Step	User Actions	System Actions
1	The user goes to "Tournaments"	The system shows all upcoming tournaments sorted by date
2	The user can filter by game type or date	The system shows matching tournaments
3	The user clicks on a tournament to view details	The system shows full details: name, date/time, game, format, description/rules, organizer name, how to contact organizer

Postconditions
User viewed tournament information.

Business Rules
Registration and fees handled by organizer outside system. System only shows tournament information and reserves the room. Past tournaments automatically archived after event date.