UML for Rental System Lendditt

**User Roles**

1. **General User (Renter/Owner)**
   * Browse available tools.
   * Filter tools based on categories, location, availability, price, and condition.
   * View tool details, ratings, reviews, and rental history.
   * View their own past and ongoing rentals.
   * Schedule and rent a tool.
   * Set rental duration (daily, weekly, monthly).
   * Communicate with tool owners via in-app messaging.
   * View and cancel upcoming rentals.
   * Manage payment information and make payments securely.
   * Rate and review rented tools after returning.
   * Receive notifications (rental confirmations, cancellations, reminders).
2. **Tool Owner**
   * List a tool for rent with detailed descriptions (condition, features, price).
   * Set availability calendar for their tool.
   * Approve or reject rental requests.
   * Receive notifications when a tool is rented or returned.
   * Set pricing for daily, weekly, or monthly rates.
   * Track tools rented out and manage payments.
   * Communicate with renters.
   * View reviews and ratings of their tools.
   * Remove or update tool listings.
3. **Administrator**
   * Monitor tool listings and user activity.
   * Handle disputes between renters and tool owners.
   * View analytics on app usage, tool rentals, etc.
   * Manage fraud detection and blacklist users if needed.
   * Resolve payment issues or disputes.
   * Enforce cancellation policies.

**User Requirements:**

**Tool Listing and Availability**

* Users can list tools with:
  + Tool name, images, description, category.
  + Tool condition (new, like new, used).
  + Rental price (per day, week, month).
  + Availability calendar (block certain dates).
  + Security deposit amount (optional).
  + Delivery/pickup method and fees (if applicable).
* **Edge Cases:**
  + Tool listed with inaccurate description (requires reporting/flagging system).
  + Tool already booked on selected dates (availability update on real-time).
  + Tool owner does not update the availability calendar (auto-prompt to owners).

**Rental Scheduling and Request**

* Users can:
  + Schedule tool rental for specific dates and times.
  + View conflicting bookings if a tool is unavailable.
  + Receive booking confirmation from the owner.
  + Cancel rental request before approval by owner.
* **Edge Cases:**
  + Renter cancels the request after approval (cancellation fee or deposit forfeiture?).
  + Tool owner delays approving a rental request (automatic expiration of request after X hours).
  + Double-booking due to delays in calendar sync.

**Payments and Security**

* Users pay for rentals through secure gateways (credit/debit cards, e-wallets).
* Option for tool owners to ask for a security deposit.
* Payment release to owner upon successful rental completion.
* Refund process for cancellations or damaged tools.
* **Edge Cases:**
  + Payment gateway failure (retry payment process or offer alternate method).
  + Dispute over tool damage (involvement of admin for mediation).
  + Renter refuses to return the tool (reporting system, penalties).
  + Tool is returned late (late fees).

**User Profiles and History**

* Each user has a profile displaying:
  + Personal info (name, contact details, photo).
  + Tool rental history (rented and listed tools).
  + Ratings and reviews from other users.
* **Edge Cases:**
  + User account gets hacked (secure login with 2FA, password recovery).
  + Misuse of profiles for fraudulent purposes (reporting system, admin intervention).

**Ratings and Reviews**

* After a rental, both parties can rate each other (star rating) and leave reviews.
* Reviews can be flagged if inappropriate.
* **Edge Cases:**
  + False reviews (allow dispute/flagging for review content).
  + No reviews from either party (automatic reminders).
  + Manipulation of ratings (monitor for fake accounts or collaborations).

**Communication and Notifications**

* In-app messaging for renters and owners to communicate.
* Notifications for:
  + Rental requests, confirmations, cancellations.
  + Rental reminders (pick-up/drop-off times).
  + New reviews or ratings.
* **Edge Cases:**
  + Spam messages (limits on message frequency, report system).
  + User ignores important notifications (force push notifications or app prompts).

**Tool Pickup/Delivery**

* Tools can be picked up by the renter or delivered by the owner (if available).
* Option to set delivery fees.
* **Edge Cases:**
  + Tool gets lost or damaged during delivery (insurance or deposit handling).
  + Renter doesn’t pick up the tool on time (penalties, auto-cancellation).
  + Tool owner refuses to deliver (penalties for non-compliance).

**Rental Disputes and Issues**

* Report system for disputes over tool condition, delays, or damages.
* Admins mediate and make decisions on disputes.
* Refund policy for unsatisfactory rentals.
* **Edge Cases:**
  + False claims of damage or delay (admins to review photo evidence and messages).
  + Lost or stolen tools (deposit forfeiture or user compensation).
  + Repeated disputes from the same user (blacklisting or temporary ban).

**Tool Return Process**

* Tool return confirmation system for both parties.
* Option for owners to inspect tools and confirm condition before final payment release.
* **Edge Cases:**
  + Renter damages tool (partial or full deposit is withheld).
  + Late return without prior communication (late fees or penalties).

**Search and Filter Tools**

* Users can filter tools by:
  + Location (proximity search).
  + Tool type and category.
  + Availability (date range).
  + Price range.
  + Rating.
* **Edge Cases:**
  + Poor search results due to incorrect listings (user reporting system).

**Admin Panel for Monitoring**

* Admin can:
  + Oversee user activity and review flagged content.
  + Set policies on cancellations, payments, and disputes.
  + Monitor rental activity and statistics.
  + Handle legal concerns (data privacy, fraud).
* **Edge Cases:**
  + Admin receives too many dispute requests (automated or delegated system).
  + Tool owner repeatedly lists damaged tools (tool quality checks, bans).

**Classes:**

1. **User**
   * Attributes:
     + - userID
     + - name
     + - email
     + - password
     + - role
     + - profilePhoto
     + - contactDetails
     + - rentalHistory
     + + ratings
     + + reviews
   * Methods:
     + + login()
     + + register()
     + + updateProfile()
     + + viewHistory()
     + + rateUser()
     + + sendMessage()
2. **Tool**
   * Attributes:
     + - toolID
     + + name
     + + description
     + + category
     + + condition
     + + pricePerDay
     + + pricePerWeek
     + + pricePerMonth
     + + availability
     + + location
     + - owner
     + + ratings
     + + reviews
     + + securityDeposit
   * Methods:
     + + createListing()
     + + updateListing()
     + + viewAvailability()
     + + bookTool()
     + + updateAvailability()
     + + rateTool()
3. **Rental**
   * Attributes:
     + - rentalID
     + - renter
     + - tool
     + + startDate
     + + endDate
     + + status
     + + paymentStatus
     + + securityDepositStatus
     + + rentalCost
     + + pickupMethod
     + + deliveryMethod
     + + lateFees
   * Methods:
     + + createRental()
     + + cancelRental()
     + + approveRental()
     + + rejectRental()
     + + returnTool()
     + + paySecurityDeposit()
4. **Payment**
   * Attributes:
     + - paymentID
     + + amount
     + + paymentDate
     + + paymentMethod
     + + paymentStatus
     + + securityDeposit
     + - transactionID
   * Methods:
     + + processPayment()
     + + refundPayment()
     + + applyLateFees()
5. **Review**
   * Attributes:
     + - reviewID
     + + rating
     + + comment
     + + reviewDate
     + - user
     + - tool
   * Methods:
     + + addReview()
     + + flagReview()
     + + viewReview()
6. **Admin**
   * Attributes:
     + - adminID
     + + name
     + + email
   * Methods:
     + + monitorActivity()
     + + resolveDispute()
     + + manageUser()
     + + removeListing()
7. **Message**
   * Attributes:
     + - messageID
     + - sender
     + - receiver
     + + content
     + + timestamp
   * Methods:
     + + sendMessage()
     + + receiveMessage()

**Relationships:**

* **User** has a one-to-many relationship with **Tool** (A user can own multiple tools).
* **User** can rent multiple **Tools** through **Rental** (many-to-many relationship, mediated by the Rental class).
* **User** can make multiple **Payments** for **Rentals**.
* **User** can leave **Reviews** for both **Tools** and **Users**.
* **Admin** monitors **Users**, **Tools**, and **Rentals**, and can resolve disputes.
* **Message** is associated with **User** (one-to-many: a user can send multiple messages).

**Visibility Guidelines:**

* Use **+ (Public)** for methods and attributes that need to be accessed by other classes or users (e.g., login(), name, sendMessage()).
* Use **- (Private)** for internal attributes or methods that shouldn't be directly accessed (e.g., userID, password, transactionID).