CS-319

Deliverable 1

Team 06

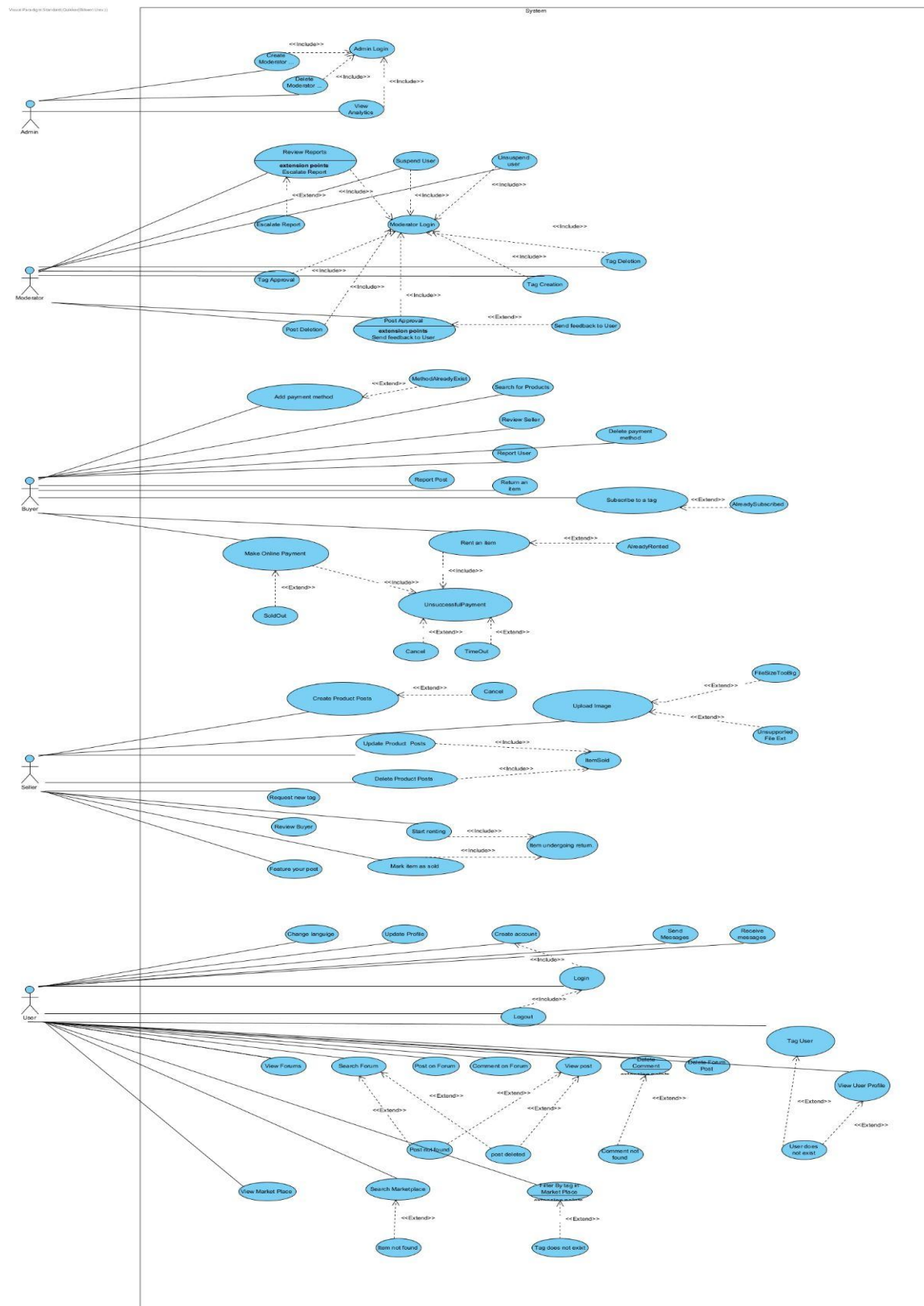## Group Members

Muhammad Rowaha

Ather Ilyas

Mehshid Atiq

Ghulam Ahmed

Ismail Özgenç

Date: 15/10/2023

1. Use Case Diagram

2. Use Case Details

2.1 - Admin

- Login
  - Name: Admin login
  - Participating actor: Admin
  - Entry Condition:
    - Admin has a valid username and password.
  - Exit condition:
    - Admin is logged in and has access to the admin dashboard.
  - Flow of events:
    - Admin enters username and password.
    - System verifies the credentials.
    - System grants access to the admin dashboard.
  - Special / quality requirements: Secure login mechanism to prevent unauthorized access. Two-factor authentication is recommended.
- Create moderator accounts
  - Name: Create Moderator Account
  - Participating actor: Admin
  - Entry Condition:
    - Admin is logged into the admin dashboard.
  - Exit condition:
    - New moderator account is created and active.
  - Flow of events:
    - Admin navigates to "Create Moderator" section.
    - Admin enters the required details for the new moderator.
    - System validates the input and creates a new moderator account.
    - System displays a success notification to the admin.
  - Special / quality requirements: Admin should be able to set unique permissions and access for each moderator.
- Delete moderator accounts
  - Name: Delete Moderator Account
  - Participating actor: Admin
  - Entry Condition:
    - Admin is logged into the admin dashboard.
    - There are active moderator accounts.
  - Exit condition:
    - Selected moderator account is deactivated and removed.
  - Flow of events:
    - Admin navigates to the list of active moderators.
    - Admin selects the moderator account to be deleted.
    - System prompts for confirmation.
    - Admin confirms the deletion.
    - System deactivates and removes the selected moderator account.
  - Special / quality requirements: Admins should receive a prompt to ensure the action is not accidental.

- View analytics
  - Name: View Analytics
  - Participating actor: Admin
  - Entry Condition:
    - Admin is logged into the admin dashboard.
  - Exit condition:
    - Admin views the desired analytics.
  - Flow of events:
    - Admin navigates to the "Analytics" section.
    - System displays various analytics such as user activity, sales trends, popular items, etc.
    - Admin can filter or select specific data points for detailed insights.
  - Special / quality requirements: The system should provide real-time updates and ensure data accuracy. The display should be user-friendly, allowing for graphical representations and easy understanding of trends.

2.2 - Moderator

- Moderator Login
  - Name: Moderator Login
  - Participating actor: Moderator
  - Entry Condition:
    - Moderator has a valid username and password.
  - Exit condition:
    - Moderator is logged in and has access to the moderator dashboard.
  - Flow of events:
    - Moderator enters username and password.
    - System verifies the credentials.
    - System grants access to the moderator dashboard.
  - Special / quality requirements: Secure login mechanism to prevent unauthorized access. Two-factor authentication is recommended.
- Review Reports
  - Name: Review Reports
  - Participating actor: Moderator
  - Entry Condition:
    - Moderator is logged into the website.
    - Reports are available to be reviewed. [optional]
  - Exit condition:
    - Moderator has reviewed and taken necessary action on the report.
  - Flow of events:
    - Moderator accesses the reports section.
    - The system displays a list of user reports.
    - Moderator selects a report to review.
    - The system displays report details.
    - Moderator takes an appropriate action (e.g., escalate, dismiss, or take corrective action).
- Escalate Report
  - Name: Escalate Report

- ○ Participating actor: Moderator
- ○ Entry Condition:
  - ■ Moderator is reviewing a severe report that needs higher attention.
- ○ Exit condition:
  - ■ Report has been escalated to higher authority or department.
- ○ Flow of events:
  - ■ While reviewing, Moderator identifies the report as severe.
  - ■ Moderator selects the option to escalate the report.
  - ■ System prompts for additional information or reasons for escalation.
  - ■ Moderator provides the required details.
  - ■ System marks the report as escalated and notifies the concerned authority or department.
- ○ Special / quality requirements: The system should log the escalation event for auditing purposes.
- ● Send Feedback to User
  - ○ Name: Send Feedback to User
  - ○ Participating actor: Moderator
  - ○ Entry Condition:
    - ■ A post is reviewed and not approved.
  - ○ Exit condition:
    - ■ Feedback is sent to the post creator about the reasons for non-approval.
  - ○ Flow of events:
    - ■ Moderator reviews a post and decides not to approve it.
    - ■ Moderator selects the option to send feedback to the user.
    - ■ System prompts for feedback details or reasons for non-approval.
    - ■ Moderator provides the feedback.
    - ■ System sends the feedback to the post creator.
  - ○ Special / quality requirements: The feedback should be clear, respectful, and constructive.
- ● Tag approval
  - ○ Name: Tag Approval
  - ○ Participating actor: Moderator
  - ○ Entry Condition:
    - ■ Moderator is logged into the website.
    - ■ There are tags awaiting approval.
  - ○ Exit condition:
    - ■ Moderator has approved or rejected the tag.
  - ○ Flow of events:
    - ■ Moderator accesses the tag approval section.
    - ■ The system displays a list of tags awaiting approval.
    - ■ Moderator selects a tag to review.
    - ■ The system displays tag details.
    - ■ Moderator decides to approve or reject the tag.
  - ○ Special / quality requirements: The system should provide a brief reason for the tag creation to aid the moderator in decision making.
- ● Tag creation
  - ○ Name: Tag Creation

- - Participating actor: Moderator
    - Entry Condition:
      - Moderator is logged into the website.
    - Exit condition:
      - Moderator has created a new tag.
    - Flow of events:
      - Moderator accesses the tag creation section.
      - The system prompts the moderator to enter tag details.
      - Moderator submits the new tag details.
      - The system saves and displays the new tag.
    - Special / quality requirements: The system should ensure that the tag does not already exist.
- Tag deletion
  - Name: Tag Deletion
  - Participating actor: Moderator
  - Entry Condition:
    - Moderator is logged into the website.
    - There are tags available to be deleted.
  - Exit condition:
    - Moderator has deleted the tag.
  - Flow of events:
    - Moderator accesses the tag section.
    - The system displays a list of tags.
    - Moderator selects a tag for deletion.
    - The system prompts the moderator to confirm deletion.
    - Moderator confirms deletion.
  - Special / quality requirements: The system should provide a warning about the potential impact of deleting the tag.
- Suspend User
  - Name: Suspend User
  - Participating actor: Moderator
  - Entry Condition:
    - Moderator is logged into the website.
    - There are users on the platform.
  - Exit condition:
    - Moderator has suspended a user.
  - Flow of events:
    - Moderator accesses the user section.
    - The system displays a list of users.
    - Moderator selects a user to suspend.
    - The system prompts the moderator to provide a reason for suspension.
    - Moderator submits the reason and confirms the suspension.
  - Special / quality requirements: The system should notify the suspended user about the suspension and the reason.
- Unsuspend User
  - Name: Unsuspend User
  - Participating actor: Moderator

- ○ Entry Condition:
  - ■ Moderator is logged into the website.
  - ■ There are users who are currently suspended.
- ○ Exit condition:
  - ■ Moderator has unsuspended a user.
- ○ Flow of events:
  - ■ Moderator accesses the suspended user section.
  - ■ The system displays a list of suspended users.
  - ■ Moderator selects a user to unsuspend.
  - ■ The system prompts the moderator to confirm the unsuspension.
  - ■ Moderator confirms.
- ○ Special / quality requirements: The system should notify the user about the change in their status.
- ● Post deletion
  - ○ Name: Post Deletion
  - ○ Participating actor: Moderator
  - ○ Entry Condition:
    - ■ Moderator is logged into the website.
    - ■ There are posts available on the platform.
  - ○ Exit condition:
    - ■ Moderator has deleted a post.
  - ○ Flow of events:
    - ■ Moderator accesses the posts section.
    - ■ The system displays a list of posts.
    - ■ Moderator selects a post for deletion.
    - ■ The system prompts the moderator to confirm deletion.
    - ■ Moderator confirms the deletion.
  - ○ Special / quality requirements: The system should archive the deleted post for a specific period for auditing purposes.
- ● Post Approval
  - ○ Name: Post Approval
  - ○ Participating actor: Moderator
  - ○ Entry Condition:
    - ■ Moderator is logged into the website.
    - ■ There are posts awaiting approval.
  - ○ Exit condition:
    - ■ Moderator has approved or rejected a post.
  - ○ Flow of events:
    - ■ Moderator accesses the post approval section.
    - ■ The system displays a list of posts awaiting approval.
    - ■ Moderator selects a post to review.
    - ■ The system displays post details.
    - ■ Moderator decides to approve or reject the post.
  - ○ Special / quality requirements: The system should notify the post creator about the approval status.

2.3 - Email Service

- Send Notification to User
  - Name: Send Notification to User
  - Participating actor: Email Service
  - Entry Condition:
    - A user action requires a notification (e.g., Post approval, Post deletion, User suspension, etc.).
    - Admin or moderator triggers the notification action from the system.
  - Exit condition:
    - Notification email has been sent to the user.
  - Flow of events:
    - A user action/event takes place that requires a notification.
    - The system sends a request to the Email Service with the necessary details (user's email, subject, message content).
    - Email Service processes the request.
    - Email Service sends an email notification to the specified user.
  - Special / quality requirements:
    - The email notification should be sent within 2 minutes of the request.
    - The content of the email should be clear, precise, and in line with the user action taken.
    - Email Service should provide a delivery status back to the application.
- Account Creation
  - Name: Account Activation
  - Participating actor: Email Service
  - Entry Condition:
    - Admin or user creates a new account.
    - The system needs to send an activation link or code to the email address.
  - Exit condition:
    - Activation email has been sent to the new user.
  - Flow of events:
    - Request for a new account is received.
    - The system generates an activation link or code for the new account.
    - The system sends a request to the Email Service with the new user's email details and the activation content.
    - Email Service processes the request.
    - Email Service sends the activation email to the new user.
  - Special / quality requirements:
    - The activation email should be sent immediately after account creation.
    - The activation link or code should be valid for 48 hours.
    - Email Service should provide a delivery status back to the application.
- Report Submission Confirmation
  - Name: Report Submission Confirmation
  - Participating actor: Email Service
  - Entry Condition:
    - A user submits a report about a post or another user.

- - - The system acknowledges the report and confirms its receipt to the user.
      - ○ Exit condition:
        - ■ Confirmation email has been sent to the reporting user.
      - ○ Flow of events:
        - ■ A user submits a report.
        - ■ The system sends a request to the Email Service with the reporting user's email details and a confirmation message.
        - ■ Email Service processes the request.
        - ■ Email Service sends the confirmation email to the reporting user.
      - ○ Special / quality requirements:
        - ■ The confirmation email should be sent within 5 minutes of the report submission.
        - ■ The content of the email should provide details about the report's receipt and expected follow-up time.
        - ■ Email Service should provide a delivery status back to the application.

2.4 - Payment Service

- ● Process Sale Transaction
  - ○ Name: Process Sale Transaction
  - ○ Participating actor: Payment Service
  - ○ Entry Condition:
    - ■ A buyer decides to purchase a second-hand item from a seller.
    - ■ The system initiates a payment transaction for the specified amount.
  - ○ Exit condition:
    - ■ Payment has been processed, and both the buyer and seller are notified.
  - ○ Flow of events:
    - ■ Buyer selects an item and chooses to purchase.
    - ■ The system calculates the total amount and initiates a payment request to the Payment Service.
    - ■ Payment Service processes the payment request and deducts the amount from the buyer's account.
    - ■ Upon successful transaction, Payment Service notifies the system.
    - ■ The system confirms the sale and notifies both the buyer and seller.
  - ○ Special / quality requirements:
    - ■ Transaction should be processed securely, with all data encrypted.
    - ■ In case of payment failure, the system should be notified immediately.
    - ■ The buyer should be redirected to a confirmation page post successful transaction.
- ● Rent an Item
  - ○ Name: Rent an Item
  - ○ Participating actor: Payment Service
  - ○ Entry Condition:
    - ■ A renter decides to rent an item listed on the platform.
    - ■ The item is available for the specified rental period.

- ○ Exit condition:
  - ■ Payment for the rental has been processed, and both the renter and the item's owner (lender) are notified.
- ○ Flow of events:
  - ■ Renter browses the platform and selects an item they wish to rent.
  - ■ Renter chooses the desired rental period and sees the total rental cost.
  - ■ Renter initiates the rental process, prompting the system to request payment via the Payment Service.
  - ■ Payment Service processes the payment, deducting the amount from the renter's account.
  - ■ Upon successful transaction, Payment Service notifies the system.
  - ■ The system confirms the rental and notifies both the renter and the lender.
- ○ Special / quality requirements:
  - ■ All transactions should be processed securely, with end-to-end encryption.
  - ■ The system should provide an option for renter's insurance or a security deposit, if required by the lender.
  - ■ Any terms and conditions related to the rental (e.g., return condition, late fees) should be clearly communicated to the renter before payment.

2.5 - USER

- ● Login
  - ○ Participating Actor: User (Buyer/Seller)
  - ○ Entry Condition:
    - ■ User has a registered account with a valid username and password.
  - ○ Exit Condition:
    - ■ User is successfully logged in and gains access to personalized features.
  - ○ Flow of Events:
    - ■ User launches the buying and selling app.
    - ■ Selects the "Login" option.
    - ■ User enters a valid username and password in the designated fields.
    - ■ The system validates the entered credentials against the stored user database.
    - ■ If the credentials are correct, the system grants access to the user account.
    - ■ User is directed to the personalized dashboard displaying relevant buying and selling information.
  - ○ Special/Quality Requirements:
    - ■ The login process employs robust encryption to safeguard user credentials during transmission and storage.
    - ■ Passwords are securely hashed and stored.
    - ■ Two-Factor Authentication (2FA):

- ■ The system provides clear and user-friendly error messages for incorrect login attempts.
- ■ After a certain number of failed attempts, the system may temporarily lock the account to prevent unauthorized access.
- ■ Automatic logout occurs after a period of inactivity.
- Logout
  - Participating Actor: User (Buyer/Seller)
  - Entry Condition:
    - ■ User is currently logged into the buying and selling app.
  - Exit Condition:
    - ■ User is successfully logged out and no longer has access to the personalized features.
  - Flow of Events:
    - ■ User selects the "Logout" option within the app.
    - ■ The system may present a confirmation prompt to ensure the user intends to log out.
    - ■ The system terminates the user's active session.
    - ■ The user is redirected to the app's login screen.
    - ■ A confirmation message is displayed indicating a successful logout.
  - Special/Quality Requirements:
    - ■ Ensure that all active sessions and associated session data are properly terminated upon logout.
    - ■ Any session-related cookies or tokens are invalidated.
    - ■ The logout process should not be susceptible to session fixation or other security vulnerabilities.
    - ■ If the user is inactive for a specified period, implement an automatic logout mechanism to enhance security.
    - ■ If the user has any unsaved changes, inform them before proceeding with the logout.
- Create Account
  - Name: User Account Creation
  - Participating Actor: User (Buyer/Seller)
  - Entry Condition:
    - ■ User does not have an existing account in the buying and selling app.
  - Exit Condition:
    - ■ User successfully creates an account and gains access to personalized features.
  - Flow of Events:
    - ■ User selects the "Create Account" or "Sign Up" option within the app.
    - ■ User provides required information, including username, email, password, and any other necessary details.
    - ■ The system validates the entered information, ensuring it meets the required criteria (e.g., valid email format, strong password).
    - ■ If the information is valid, the system creates a new user account and associates it with the provided credentials.
    - ■ A confirmation message is displayed, informing the user that the account creation was successful.

- ■ Optionally, the system may automatically log in the user after account creation.
  - ○ Special/Quality Requirements:
    - ■ Implement secure password storage practices, such as hashing, to protect user passwords.
    - ■ Optionally, send a verification email to the user to confirm the legitimacy of their provided email address.
    - ■ Provide a secure method for users to recover their account in case they forget their credentials.
    - ■ Implement measures to prevent automated bots from creating fake accounts.
    - ■ Ensure users are informed about the app's privacy policy and terms of service during the account creation process.

- ● View User Profile
  - ○ Participating Actor: User (Buyer/Seller)
  - ○ Entry Condition:
    - ■ User is currently logged into the buying and selling app.
  - ○ Exit Condition:
    - ■ User successfully views the profile information of another user.
  - ○ Flow of Events:
    - ■ User navigates to a section within the app where profiles can be viewed, such as a user's profile list or a product listing with user details.
    - ■ User selects the profile of another user they want to view, typically by tapping on their username or profile picture.
    - ■ The system fetches and displays relevant profile information, such as username, contact details, bio, and any other public details
    - ■ If the user being viewed is a seller, the app displays their seller ratings, which may be shown as stars, numerical scores, or other visual indicators.
    - ■ The user has the option to read detailed feedback left by other buyers who have purchased items from the seller.
    - ■ If applicable, the user may have options to interact with the viewed profile, such as sending a message, following, or viewing listed items.
    - ■ User has the option to return to the previous screen or navigate back to the main interface of the app.
  - ○ Special/Quality Requirements:
    - ■ Ensure that only the information marked as public or shared by the user is displayed to others.
    - ■ Display the most up-to-date information about the user by fetching real-time data from the server.
    - ■ Respect and adhere to the privacy settings of each user, allowing them to control the visibility of their information.
    - ■ If there are interactive features, validate that the user viewing the profile has the necessary permissions to perform actions like sending messages or following.

- Optimize data retrieval to provide a smooth and fast profile viewing experience.

- Update Profile
  - Participating Actor: User (Buyer/Seller)
  - Entry Condition:
    - User is currently logged into the buying and selling app.
  - Exit Condition:
    - User's profile information is successfully updated.
  - Flow of Events:
    - User navigates to the "Profile" or "Account Settings" section within the app.
    - Within the profile settings, the user selects the "Edit Profile" or a similar option.
    - User updates the desired profile information, such as contact details, profile picture or any other relevant data.
    - The system validates the entered information, ensuring it meets any required criteria.
    - If the information is valid, the system updates the user's profile with the new data.
    - A confirmation message is displayed, informing the user that their profile has been successfully updated.
  - Special/Quality Requirements:
    - Ensure that any updated profile information is transmitted securely between the user's device and the server.
    - Implement validation checks to ensure the integrity and accuracy of the updated information.
    - Respect user privacy settings and permissions when updating certain sensitive information.
    - If applicable, include a mechanism for users to update their profile picture with appropriate file format and size restrictions.
    - Optionally, provide a notification to the user if any changes made to the profile require additional verification.
- Change Language
  - Participating Actor: User (Buyer/Seller)
  - Entry Condition:
    - User is currently using the buying and selling app and wants to change the language.
  - Exit Condition:
    - App interface language is successfully changed based on the user's preference.
  - Flow of Events:
    - User navigates to the "Settings" or "Preferences" section within the app.
    - Within the settings, the user looks for the "Language" or "Language Preference" option.
    - User selects "Turkish" or the equivalent language option from the available choices.

- The system may present a confirmation prompt to ensure the user wants to change the language.
- The system updates the app interface language to Turkish based on the user's selection.
- A confirmation message is displayed, informing the user that the language change was successful.
  - Special/Quality Requirements:
    - Ensure that the app supports the desired language, in this case, Turkish.
    - Implement a system that allows for dynamic language switching without requiring a restart of the app.
    - Save the user's language preference so that it persists across sessions.
    - Provide clear instructions or visual cues to help users locate and select their preferred language.
    - Ensure that the chosen language supports accessibility features, such as screen readers or other assistive technologies.
- Send Messages
  - Participating Actor: User (Buyer/Seller)
  - Entry Condition:
    - User is currently logged into the buying and selling app.
  - Exit Condition:
    - User successfully sends a message to another user.
  - Flow of Events:
    - User navigates to the messaging or chat section within the app.
    - User chooses the recipient from their contact list or recent conversations.
    - User starts a new message or selects an existing conversation.
    - User types the message they want to send to the selected recipient.
    - User presses the "Send" button to transmit the message.
  - Optional Attachments:
    - If applicable, the user may attach images, documents, or other files to the message.
    - The system confirms the successful delivery of the message, providing a visual cue or notification.
  - Special/Quality Requirements:
    - Implement a real-time messaging system to ensure quick and efficient message delivery.
    - If attachments are allowed, implement a secure mechanism for uploading, storing, and downloading files.
    - Provide notifications to users for new messages and ensure they are informed promptly.
    - Implement end-to-end encryption to secure message content during transmission.
    - Maintain a message history for users to review past conversations.
    - Implement a mechanism for users to block or report inappropriate messages.
- Receive Messages

- ○ Participating Actor: User (Buyer/Seller)
- ○ Entry Condition:
  - ■ User is logged into the buying and selling app.
- ○ Exit Condition:
  - ■ User successfully receives and views incoming messages.
- ○ Flow of Events:
  - ■ If the app supports push notifications, the user may receive a notification indicating a new message.
  - ■ User navigates to the messaging or chat section within the app.
  - ■ The app displays an indicator (e.g., a bold message, notification badge) to highlight the presence of unread messages.
  - ■ User selects the conversation containing the new message or taps on the notification to open the relevant chat.
  - ■ The system displays the received message, including text, images, or other attachments.
  - ■ The user may have options to reply to the message, or perform other relevant actions.
  - ■ If supported, the app provides real-time updates to reflect any additional messages received during the user's session.
  - ■ User has the option to return to the previous screen or navigate back to the main interface of the app.
- ○ Special/Quality Requirements:
  - ■ Implement a real-time messaging system to ensure quick and efficient delivery of messages.
  - ■ If attachments are allowed, implement a secure mechanism for uploading, storing, and downloading files.
  - ■ Provide notifications to users for new messages and ensure they are informed promptly.
  - ■ Implement end-to-end encryption to secure message content during transmission.
  - ■ Maintain a message history for users to review past conversations.
  - ■ Implement a mechanism for users to block or report inappropriate messages.

- ● View Post
  - ○ Activity: View Product Post from Market Place
  - ○ Participating Actor: User (Buyer)
  - ○ Entry Condition:
    - ■ User is logged into the buying and selling app.
    - ■ User makes a search for the product or scrolls through the available products
  - ○ Exit Condition:
    - ■ User successfully views the details of a product post on the marketplace.
  - ○ Flow of Events:
    - ■ User navigates to the "Marketplace" or "Browse" section within the app.

- ■ User scrolls through the available product listings or uses search and filtering options to find specific items.
- ■ User clicks or taps on a specific product post to view more details.
- ■ The system displays detailed information about the product, including:
  - ● Title: The name or title of the product.
  - ● Description: A comprehensive description providing additional information about the product's features, condition, and other relevant details.
  - ● Price: The selling price of the item.
  - ● Images: Clear and high-quality images showing the product from different angles.
- ■ User can view information about the seller, including their username, seller ratings, and feedback.
- ■ The user may have options to send a message to the seller for inquiries or negotiations.
- ■ The user may check the seller's ratings and feedback to assess the seller's reputation.
- ■ User has the option to return to the previous screen or navigate back to the main marketplace interface.
  - ○ Special/Quality Requirements:
    - ■ If the product is still available, implement real-time updates to reflect any changes in the product status or details.
    - ■ Ensure that the messaging system is secure and allows users to communicate safely about the product.
    - ■ Provide users with effective search and filtering options to help them find products that match their preferences.
- ● Post on forum
  - ○ Participating Actor: User
  - ○ Entry Condition:
    - ■ User is logged into the buying and selling app.
    - ■ User has an item they want to report as lost or found.
  - ○ Exit Condition:
    - ■ User successfully posts information about a lost or found item on the forum.
  - ○ Flow of Events:
    - ■ User navigates to the "Lost and Found" section within the app, which may be part of a community or forum feature.
    - ■ User chooses whether they are posting about a lost item, a found item, or seeking information about a lost item.
    - ■ User provides detailed information about the item, including:
      - ● Type of Item: Description of the lost or found item (e.g., keys, wallet, phone).
      - ● Date and Location: When and where the item was lost or found.
      - ● Unique Features: Any distinctive features that can help identify the item.
      - ● User may upload images of the lost or found item to visually aid the post.

- - - ● User can include any additional information that might be relevant, such as a contact number, reward details, or specific instructions.
      - ● If applicable, the user may tag the location where the item was lost or found, making it easier for others in the same area to see the post.
    - ■ The system provides a preview of the post for the user to review before finalizing.
    - ■ User confirms and posts the information about the lost or found item on the forum.
    - ■ If the app supports push notifications, the user may receive a notification confirming the successful posting of the lost and found information.

  - ○ Special/Quality Requirements:
    - ■ If possible, implement real-time updates to notify users about recently posted lost or found items.
    - ■ Implement moderation features to review and approve posts to prevent misuse or inappropriate content.
    - ■ Provide users with the ability to search and filter lost and found posts based on categories, locations, or keywords.
    - ■ Ensure that sensitive information, such as contact details, is handled with privacy in mind.
- ● Comment on Forum
  - ○ Name: Comment on Forum
  - ○ Participating Actor: User
  - ○ Entry Condition:
    - ■ User is logged into the buying and selling app.
    - ■ User is viewing a forum post.
  - ○ Exit Condition:
    - ■ User successfully adds a comment to a forum post.
  - ○ Flow of Events:
    - ■ User navigates to the forum section within the app and selects a post they want to comment on.
    - ■ User reads the content of the forum post and scrolls down to the comment section of the post and enters their comment in the provided text box.
    - ■ The system provides a preview of the comment for the user to review before finalizing.
    - ■ User confirms and posts the comment to the forum.
    - ■ If the app supports push notifications, the user may receive a notification confirming the successful posting of the comment.
    - ■ User has the option to respond to other comments on the forum post, creating a threaded discussion.
    - ■ The user may have the option to edit or delete their comment.
  - ○ Special/Quality Requirements:

- ■ Implement moderation features to ensure that comments adhere to community guidelines and prevent inappropriate content.
- ■ If possible, implement real-time updates to show new comments on the forum post without needing a manual refresh.
- ■ Encourage user engagement by allowing likes, upvotes, or other positive interactions with comments.
- ● Delete forum post
  - ○ Name: Delete Forum Post
  - ○ Participating Actor: User (Forum Post Author)
  - ○ Entry Condition:
    - ■ User is logged into the buying and selling app.
    - ■ User authored a forum post.
  - ○ Exit Condition:
    - ■ User successfully deletes their forum post.
  - ○ Flow of Events:
    - ■ User navigates to the forum section within the app and finds the post they want to delete.
    - ■ User selects the specific forum post, accesses the post options or settings,and locates the delete option within the post settings to initiate the delete action.
    - ■ The system presents a confirmation prompt to ensure the user intends to delete the post.
    - ■ User confirms the deletion, indicating their final decision to remove the forum post.
    - ■ If the app supports push notifications, the user may receive a notification confirming the successful deletion of the forum post.
    - ■ The forum post is immediately removed from public view.
    - ■ If applicable, the system may provide a brief window of time during which the user can undo the deletion in case of accidental actions.
  - ○ Special/Quality Requirements:
    - ■ Implement a clear and user-friendly confirmation mechanism to prevent accidental deletions.
    - ■ Ensure that the deletion process respects user privacy, especially in cases where the post may have attracted comments or interactions.
    - ■ Implement moderation controls to prevent misuse of the deletion feature and to review posts before they are permanently removed.
    - ■ Maintain an audit trail or log of deleted posts for moderation or accountability purposes.
- ● View Forum
  - ○ Participating Actor: User (Community Member)
  - ○ Entry Condition:
    - ■ User is logged into the buying and selling app.
  - ○ Exit Condition:
  - ○ User successfully views the Lost and Found forum to browse and interact with posts.
  - ○ Flow of Events:
    - ■ User navigates to the "Lost and Found" section within the app, often found in the community or forum features.

- There are categories within the Lost and Found forum (e.g., Lost Items, Found Items). The user can choose between lost and found forums.
- User can put date filters to search from postsaccording to their date of posting.
- User scrolls through the list of posts in the Lost and Found forum to discover items that may match their interests or to assist others.
- User clicks or taps on a specific post to view more details about a lost or found item.
- The system displays detailed information about the selected post, including:
    - Type of Item: Description of the lost or found item.
    - Date and Location: When and where the item was lost or found.
    - Unique Features: Any distinctive features mentioned by the user.
    - If the user included images in their Lost and Found post, these are displayed for visual reference.
- User has the option to engage with the post by leaving comments or providing information if they have relevant details about the item.
- User has the option to return to the main Lost and Found forum page to continue browsing other posts.
- ○ Special/Quality Requirements:
    - If possible, implement real-time updates to show new Lost and Found posts without requiring manual refresh.
    - Provide users with the ability to filter Lost and Found posts based on date.
    - Implement moderation features to ensure that posts adhere to community guidelines and prevent misuse.
    - Respect user privacy by ensuring that sensitive information in Lost and Found posts is handled appropriately.
- ● Tag user
    - ○ Participating Actor: User (Forum Post Author)
    - ○ Entry Condition:
        - User is logged into the buying and selling app.
        - User authored a forum post.
    - ○ Exit Condition:
        - User successfully tags another user in their forum post.
    - ○ Flow of Events:
        - User navigates to the forum section within the app and finds the post in which they want to tag another user.
        - User selects the specific forum post and accesses the post options or settings.
        - User locates the tag option within the post settings and initiates the tagging action.
        - User specifies the username or selects the user from a list to tag in the forum post.

- - - ■ If the app supports push notifications, the tagged user may receive a notification about being mentioned in the forum post.
      - ■ The forum post is updated to show the tagged user, creating a link to their profile or notifying them of the mention.
      - ■ If applicable, the system may provide a brief window of time during which the user can undo the tagging in case of accidental actions.
    - ○ Special/Quality Requirements:
      - ■ Implement a clear and user-friendly confirmation mechanism to prevent accidental taggings.
      - ■ Ensure that the tagging process respects user privacy, and users are notified about being tagged.
      - ■ Implement moderation controls to prevent misuse of the tagging feature and to review tags if necessary.

- ● View marketplace
  - ○ Participating Actor: User (Buyer/Seller)
  - ○ Entry Condition:
    - ■ User is logged into the buying and selling app.
    - ■ User wants to browse the most recent marketplace listings without using search or tag features.
  - ○ Exit Condition:
    - ■ User successfully views the most recent marketplace listings.
  - ○ Flow of Events:
    - ■ User goes to the marketplace or browse section within the app.
    - ■ The system displays a curated list of the most recently uploaded products or services.
    - ■ User scrolls through the displayed listings to explore the most recent additions to the marketplace.
    - ■ User clicks or taps on specific listings to view detailed information about the products or services.
    - ■ User has the option to explore further, click on individual listings, search for a specific product, filter products by tags or go back to the main marketplace interface.
  - ○ Special/Quality Requirements:
    - ■ Ensure that the list of most recent marketplace listings is updated in real-time to reflect the latest uploads.
    - ■ Optimize the system for efficient loading of recent listings, providing a seamless browsing experience.
    - ■ Design a clear and user-friendly interface for users to easily access and navigate through the most recent marketplace listings.
    - ■ Display upload dates or timestamps for each listing to provide users with information about when the products or services were added to the marketplace.
- ● Search Marketplace
  - ○ Activity: Search Marketplace with Tags
  - ○ Participating Actor: User (Buyer)
  - ○ Entry Condition:
    - ■ User is logged into the buying and selling app.

- ○ Exit Condition:
  - ■ User successfully searches and discovers relevant product listings on the marketplace using tags.
- ○ Flow of Events:
  - ■ User navigates to the "Search" section within the marketplace.
  - ■ User types the name of the product they want, this may include keywords that match the tags. Tags may include categories or specific attributes of the listed products in the app.
  - ■ User triggers the search function based on the selected tags.
  - ■ The app displays a list of products matching the selected tags. User can scroll through these results.
  - ■ User clicks or taps on a specific product from the search results to view more details.
- ○ Special/Quality Requirements:
  - ■ Implement real-time updates to reflect any changes in the product status or details.
  - ■ Optimize the tag-based search functionality for efficiency and accuracy.

- ● Filter Marketplace by tags
  - ○ Participating Actor: User (Buyer/Seller)
  - ○ Entry Condition:
    - ■ User is logged into the buying and selling app.
    - ■ User wants to filter marketplace listings based on specific tags.
  - ○ Exit Condition:
    - ■ User successfully filters and views marketplace listings according to the selected tags.
  - ○ Flow of Events:
    - ■ User goes to the marketplace or browse section within the app.
    - ■ User selects the filter option to refine marketplace listings based on tags.
    - ■ User specifies one or more tags relevant to their interests or search criteria.
    - ■ The system displays marketplace listings that match the specified tags.
    - ■ User explores the filtered results, viewing marketplace listings associated with the selected tags.
    - ■ User has the option to further refine the search or adjust the tag filtering as needed.
  - ○ Special/Quality Requirements:
    - ■ Ensure that the tag filtering mechanism is efficient and provides accurate results.
    - ■ Design a clear and intuitive user interface for selecting and adjusting tag filtering options.
    - ■ Implement real-time updates to reflect any changes in the marketplace listings matching the selected tags.
    - ■ Allow flexibility in selecting multiple tags to broaden or narrow the search based on user preferences.

2.6 - BUYER

- Search for Products
  - Entry Condition: User is logged into the app.
  - Exit Condition: User successfully finds and views product listings.
  - Flow of Events:
    - User navigates to the "Marketplace" section within the app.
    - User utilizes search and filtering options to find specific items.
    - The system displays relevant product listings based on the search criteria.
    - User clicks on a specific product to view more details.
    - User has the option to purchase the product, send a message to the seller.
  - Special/Quality Requirements:
    - Implement efficient search algorithms and indexing for quick and accurate results.
- Make Online Payment
  - Entry Condition: User has selected a product for purchase.
  - Exit Condition: Payment is successfully processed, and the purchase is confirmed.
  - Flow of Events:
    - User selects the "Buy" option for the chosen product.
    - The system displays the total amount.
    - User adds a new payment method to proceed.
    - The system securely processes the payment.
    - Upon successful payment, the system confirms the purchase and provides a receipt.
  - Special/Quality Requirements:
    - Implement secure payment gateways and encryption to protect financial information.
    - Provide receipts and transaction history for users to review their payments.
- Return an Item
  - Entry Condition: User has purchased an item.
  - Exit Condition: Return request is submitted, and the user receives confirmation.
  - Flow of Events:
    - User navigates to the "Order History" section within the app.
    - User selects the order containing the item to be returned.
    - User chooses the "Return Item" option and provides a reason.
    - The system reviews the return request and notifies the user of the approval or denial.
    - If approved, the user receives instructions for returning the item and any associated refund details.
  - Special/Quality Requirements:

- ■ Allow buyers to track the return status and receive updates on the return progress.
- ● Review Seller
  - ○ Entry Condition: User has completed a purchase.
  - ○ Exit Condition: User successfully submits a review for the seller.
  - ○ Flow of Events:
    - ■ User navigates to the "Order History" section within the app.
    - ■ User selects the completed order and chooses the "Review Seller" option.
    - ■ User provides a rating and a written review for the seller's performance.
    - ■ The system confirms the submission of the review, and it becomes visible to other users.
  - ○ Special/Quality Requirements:
    - ■ Implement a rating and review system that encourages genuine and constructive feedback.
- ● Subscribe to a Tag
  - ○ Entry Condition: User is logged into the app.
  - ○ Exit Condition: User is subscribed to a specific tag and receives relevant updates.
  - ○ Flow of Events:
    - ■ User navigates to the "Tags" section within the app.
    - ■ User selects a tag of interest.
    - ■ The system provides an option to subscribe to the selected tag.
    - ■ User confirms the subscription.
    - ■ The system sends e-mail notifications with the subscribed tag to user subscribed.
  - ○ Special/Quality Requirements:
    - ■ Allow users to manage their tag subscriptions and opt-out at any time.
    - ■ Provide relevant and timely updates based on the subscribed tags.
- ● Report User
  - ○ Entry Condition: User encounters inappropriate behavior from another user.
  - ○ Exit Condition: User successfully submits a report, and appropriate actions are taken.
  - ○ Flow of Events:
    - ■ User accesses the profile of the user to be reported.
    - ■ User selects the "Report User" option and provides details about the issue.
    - ■ The system reviews the report and takes necessary actions, such as warnings or account suspension.
    - ■ The reporting user may receive feedback on the resolution of the report.
- ● Report Post
  - ○ Entry Condition: User encounters an inappropriate or suspicious product listing.
  - ○ Exit Condition: User successfully submits a report, and appropriate actions are taken.

- Flow of Events:
    - User accesses the product post to be reported.
    - User selects the "Report Post" option and provides details about the issue.
    - The system reviews the report and takes necessary actions, such as removing the post or contacting the seller.
- Add Payment Method
    - Entry Condition: User is logged into the app.
    - Exit Condition: User successfully adds a new payment method.
    - Flow of Events:
        - User navigates to the "Payment Methods" section within the app.
        - User selects the option to add a new payment method.
        - User enters the required payment details, such as card information.
        - The system validates and securely stores the new payment method for future transactions.
- Delete Payment Method
    - Entry Condition: User is logged into the app.
    - Exit Condition: User successfully deletes an existing payment method.
    - Flow of Events:
        - User navigates to the "Payment Methods" section within the app.
        - User selects the payment method to be deleted.
        - User confirms the deletion, and the system removes the selected payment method from the user's account.

2.7 - SELLER

- Create Product Posts
    - Entry Condition: User is logged into the app.
    - Exit Condition: User successfully creates a new product post.
    - Flow of Events:
        - User navigates to the "Sell" section within the app.
        - User provides details about the product, including title, description, price, and images.
        - The system validates the entered information and prompts the user for any missing details.
        - User confirms the creation of the product post.
        - The system makes the post visible in the marketplace.
    - Special/Quality Requirements:
        - Verify the accuracy and completeness of product descriptions to prevent misleading or incomplete information..
- Feature Your Post
    - Entry Condition: User has created a product post.
    - Exit Condition: User successfully features the product post for increased visibility.
    - Flow of Events:
        - User navigates to the "Products" section within the app.
        - User selects the product post to be featured.
        - User chooses the "Feature" option.

- The system processes the feature request, and the post receives enhanced visibility.
- Mark Item as Sold
  - Entry Condition: User has successfully sold a product.
  - Exit Condition: User marks the item as sold, and the post is updated.
  - Flow of Events:
    - User navigates to the "Products" section within the app.
    - User selects the item sold.
    - User chooses the "Mark as Sold" option.
    - The system updates the product post to indicate that the item is no longer available.
- Request New Tag
  - Entry Condition: User identifies the need for a new tag/category.
  - Exit Condition: User successfully submits a request for a new tag.
  - Flow of Events:
    - User navigates to the "Tags" section within the app.
    - User selects an option to request a new tag.
    - User provides details about the proposed tag and its relevance.
    - The system reviews the request and notifies the moderator of the status.
  - Special/Quality Requirements:
    - Implement a clear and efficient process for tag requests and a review process for moderator approval.
- Upload Image
  - Entry Condition: User is creating or updating a product post.
  - Exit Condition: User successfully uploads images for the product post.
  - Flow of Events:
    - User selects the option to add images to the product post.
    - User chooses image files from their device's gallery or camera.
    - The system uploads and processes the images, displaying them in the product post.
    - User has the option to add captions or descriptions to the uploaded images, if desired.
- Delete Product Posts
  - Entry Condition: User has product posts that need to be removed.
  - Exit Condition: User successfully deletes one or more product posts.
  - Flow of Events:
    - User navigates to the "Products" section within the app.
    - User selects the product post(s) to be deleted.
    - User chooses the "Delete" option and confirms the action.
    - The system removes the selected product post(s) from the marketplace.
  - Special/Quality Requirements:
    - Ensure that seller reviews are constructive and adhere to community guidelines.
    - Notify the user if there are any ongoing transactions related to the product post.
- Review Buyer

- ○ Entry Condition: User has completed a sale with a buyer.
- ○ Exit Condition: User successfully submits a review for the buyer.
- ○ Flow of Events:
  - User navigates to the "Sales/ Order History" section within the app.
  - User selects the completed order and chooses the "Review Buyer" option.
  - User provides a rating and a written review for the buyer's behavior and transaction experience.
  - The system confirms the submission of the review, and it becomes visible to other users.
- ○ Special/Quality Requirements:
  - Ensure that seller reviews are constructive and adhere to community guidelines.
- Start Renting
  - ○ Entry Condition: User has a product suitable for rental.
  - ○ Exit Condition: User successfully lists the product for rent.
  - ○ Flow of Events:
    - User navigates to the "Sell" or "Create Listing" section within the app.
    - User provides details about the product, including title, description, rental terms, and images.
    - The system validates the entered information and prompts the user for any missing details.
    - User confirms the creation of the rental listing, and the system makes it visible in the marketplace under the "Rent" section.
  - ○ Special/Quality Requirements:
    - Implement clear and comprehensive rental terms, including duration, pricing, and conditions.

3. NFR

- Response Time:
  - ○ All actions related to user inputs should complete within 2 seconds.
- Data Integrity:
  - ○ In the event of a system crash during any upsert action, there should be no data loss.
- Timely Notification Delivery:

- ○ Notification to User: Must be sent within 2 minutes of the request.

  - ○ Account Activation: Must be sent immediately after account creation.

  - ○ Report Submission Confirmation: Must be sent within 3 minutes of report submission.

- ● Email Activation Link Validity:

  - ○ The activation link or code sent for account activation should remain valid for 48 hours.

- ● Payment Service:

  - ○ All payment transactions must be processed securely, with end-to-end encryption of data.

  - ○ In case of a payment failure, the user must be notified immediately.

  - ○ Upon successful transaction, the buyer must be redirected to a confirmation page.

  - ○ Any terms and conditions related to the rental (e.g., return condition, late fees) must be clearly communicated to the renter before payment is processed.

- ● Automatic Logout:

  - ○ Implement an automatic logout feature after 30 minutes of inactivity to enhance security.

- ● Cookie/Token Invalidation:

  - ○ Invalidate any session-related cookies or tokens upon logout.

- ● Unsaved Changes Prompt:

  - ○ Inform the user if there are any unsaved changes before proceeding with the logout.

- ● Email Verification:

  - ○ Send a verification email to the user to confirm the legitimacy of their provided email address.

- ● Profile Picture Requirements:

  - ○ If applicable, allow users to update their profile picture with a file format of JPEG or PNG, and a size restriction of maximum 5MB.

- ● Persistent Language Preference:

  - ○ Save the user's language preference so that it persists across sessions.

- Prompt Update:

  - Ensure that marking an item as sold promptly updates the product post to indicate its unavailability.

- Image Processing Speed:

  - Ensure that image uploads are processed within 5 seconds, providing a seamless experience for the seller.

- Transaction Notifications:

  - Notify the user if there are any ongoing renting/payment transactions related to the product post before deletion.

- Data Backup and Recovery:

  - Implement regular data backups (at least once a day) and a recovery mechanism in case of unexpected data loss or system failures.

4. Tech Stack

Frontend:

- Framework: Next.js

- UI Components: Material-UI (MUI) or Ant Design

Backend:

- Framework: Django (Python)

- Database Management System: PostgreSQL

System

Admin

Create Moderator

<<Include>>

Admin Login

Delete Moderator ..

<<Include>>

<<Include>>

View Analytics

Moderator

Review Reports
extension points
Escalate Report

Suspend User

Unsuspend user

<<Extend>>

<<Include>>

<<Include>>

<<Include>>

Escalate Report

Moderator Login

<<Include>>

<<Include>>

Tag Deletion

Tag Approval

Tag Creation

<<Include>>

<<Include>>

Post Deletion

Post Approval
extension points
Send feedback to User

<<Extend>>

Send feedback to User

Buyer

Add payment method

<<Extend>>

MethodAlreadyExist

Search for Products

Review Seller

Delete payment method

Report User

Report Post

Return an item

Subscribe to a tag

<<Extend>>

AlreadySubscribed

Make Online Payment

Rent an item

<<Extend>>

AlreadyRented

<<Include>>

<<Include>>

<<Extend>>

UnsuccessfulPayment

SoldOut

<<Extend>>

<<Extend>>

Cancel

TimeOut

Seller

Create Product Posts

<<Extend>>

Cancel

Upload Image

<<Extend>>

FileSizeTooBig

<<Extend>>

Unsupported File Ext

Update Product Posts

<<Include>>

ItemSold

Delete Product Posts

<<Include>>

Request new tag

Review Buyer

Start renting

<<Include>>

Item undergoing return

Feature your post

Mark item as sold

<<Include>>

User

Change language

Update Profile

Create account

Send Messages

Receive messages

<<Include>>

Login

Logout

<<Include>>

Tag User

View Forums

Search Forum

Post on Forum

Comment on Forum

View post

Delete Comment

Delete Forum Post

View User Profile

<<Extend>>

<<Extend>>

<<Extend>>

<<Extend>>

Post not found

post deleted

Comment not found

User does not exist

<<Extend>>

View Market Place

Search Marketplace

Filter By tag in Market Place

<<Extend>>

<<Extend>>

Item not found

Tag does not exist