

Team 07 – Members:

- Le Anh Duy – 23127011
- Tran Gia Huy – 23127199
- Nguyen Thanh Luan – 23127296
- Nguyen Thanh Tien – 23127539

Ho Chi Minh University of Science – Introduction to Software Engineering (23CLC03)
December 2025

Vision

The Manga Auto-Translation Platform is a web-based application that enables users to read manga in various languages by leveraging **Optical Character Recognition (OCR)** and **Machine Translation** technologies. Our vision is to break language barriers for manga fans worldwide by providing a seamless reading experience with real-time translated text displayed on manga pages. The platform will serve multiple types of users – casual readers, content uploaders, and administrators – in a unified system that balances user-generated content with quality control and community engagement.

This platform consists of two primary subsystems: a **Reader subsystem** for end-users who browse and read manga, and an **Admin/Uploader subsystem** for managing content and user roles. Readers can discover manga, view pages with automatic translations, track their reading progress, and interact through comments, ratings, and sharing. Uploaders (approved contributors) can upload manga scans which the system will automatically process (via OCR) and translate. Administrators oversee the platform by approving new content, managing user privileges, and ensuring the quality and compliance of uploaded chapters. By integrating third-party translation services and secure payment gateways, the system also supports premium content access and a reward mechanism for contributors. In summary, the vision is to create an inclusive manga reading community where content from any language can be enjoyed by everyone, with minimal friction and robust management support.

Use Cases

The following sections outline key use cases for the Manga Auto-Translation Platform, organized by actor. Each use case includes the primary actor, the actor's goal, a brief description, and a simplified main flow of events. The main actors in the system are:

- **Reader:** A regular end-user who reads manga on the platform (after registering an account). Readers can search for manga, read chapters (with translation options),

manage their favorites, comment on and rate content, share chapters, and purchase access to premium chapters.

- **Uploader:** A contributor who has been granted permission to upload manga content. Uploaders can submit new manga series or chapters to the platform. (A reader must request and be approved for this role before becoming an uploader.)
- **Admin:** An administrator with elevated privileges to manage the platform. Admins can approve or reject uploaded content, manage user accounts and roles (e.g., approve uploader requests, suspend users), and organize content (including featuring certain manga on the homepage).
- **Director:** A high-level stakeholder or super-admin who oversees platform performance and metrics. Directors can view aggregate statistics such as user traffic and revenue reports but typically do not manage day-to-day content or users directly.

Below are the important use cases grouped by these actors.

Reader Use Cases

UC1: User Registration

Actor: Reader (New User)

Goal: Create a new user account to access the platform's features.

Description: This use case allows a new visitor to register an account on the manga platform. By registering, the user becomes a **Reader** with a profile, which enables them to read and translate manga chapters, as well as use community features (favorites, comments, etc.). The registration process may support standard sign-up with email/password and third-party login options (e.g., Google or Facebook).

Main Flow:

1. The user navigates to the **Sign Up** page and chooses a registration method (direct form or third-party OAuth).
2. The user fills in required information (e.g., username, email, password) or authorizes via an external provider.
3. The system validates the input (ensuring the email is unique, password meets security criteria, etc.).
4. Upon successful validation, the system creates a new account for the user and securely stores the credentials (passwords are stored in encrypted form).

5. The user receives a confirmation (instant feedback or email verification if applicable) and can now log in as a registered Reader.

UC2: User Login

Actor: Reader (Registered User)

Goal: Access an existing account and enter the platform.

Description: This use case allows a previously registered user to log into the platform. By logging in, the user proves their identity and gains access to personalized features such as their reading history, favorite list, and any premium content they have purchased.

Main Flow:

1. The user navigates to the **Login** page (or selects a social login option).
2. The user provides their credentials (username/email and password) or uses an OAuth login button.
3. The system verifies the credentials against stored user data (for example, checking the hashed password for a match).
4. If the login information is correct, the system creates an authenticated session for the user.
5. The user is granted access to the platform as a Reader and is redirected to the homepage or their dashboard. (If credentials are invalid, an error is shown and the user may retry.)

UC3: Search for Manga

Actor: Reader

Goal: Find manga content by title, author, genre, or other criteria.

Description: The reader uses the search function to discover manga series or chapters of interest. This use case enables users to query the manga database using keywords or filters and receive relevant results quickly. Effective search helps readers navigate the large content library and find specific manga or explore new titles by genre or other metadata.

Main Flow:

1. The user enters a query or selects filters (e.g. title keywords, author name, genre, publication year) in the **Search** bar or advanced search form.
2. The system processes the query by looking up matching manga entries in the database (case-insensitive matching for text).

3. The system returns a list of results that match the query or filters, typically displaying the manga title, cover image, and brief info. Results are sorted by relevance or popularity.
4. The user views the search results and clicks on a desired manga title to view its details or start reading. (If no results are found, the system informs the user that no matching content exists.)

UC4: Read Manga Chapter (with Auto-Translation)

Actor: Reader

Goal: View a manga chapter's pages and read it in a preferred language using automatic translation.

Description: This core use case allows a reader to open a manga chapter and read through its pages. The system will display the original manga images and, upon request, overlay or show translated text for each page. The translation feature utilizes OCR to extract text from the images and then calls a machine translation service to get the text in the user's chosen language. The reader can toggle the translation on/off or select a language (if multiple target languages are supported). The system also tracks the reader's progress (which page they've reached, etc.) so that it can be saved for later sessions.

Main Flow:

1. The user selects a manga title and chooses a chapter to read, opening the **Reader** interface for that chapter.
2. The system loads the chapter's pages (images) and displays them to the user one by one (or in a continuous scroll, depending on the UI design).
3. The user enables the **Translate** option and selects a target language (e.g., English).
4. For each page the user views, the system uses OCR to detect Japanese/foreign text in the image and sends this text to an external translation API.
5. The translated text returned by the API is displayed on the page (for example, as an overlay on the image or in a sidebar), allowing the user to read the content in the chosen language.
6. As the user navigates through pages, the system **updates the reading progress** (e.g., marks the last page read, the current chapter, and the active translation language). This progress is saved to the user's profile periodically (e.g., every few seconds or when a page is changed) so that the user can resume later.

7. The user reads the manga with the assistance of the translated text. They can disable translation to see the original text at any time. The system ensures that each page's translation is fetched quickly (within a few seconds per page) for a smooth reading experience.

UC5: Manage Favorites List

Actor: Reader

Goal: Save or remove a manga in the user's personal favorites list for easy access.

Description: This use case allows a reader to maintain a list of favorite manga titles or chapters. By adding a manga to favorites, the user can quickly navigate to it later without searching. The system will store the favorites list in the user's profile, ensuring it persists across sessions. Users can also remove items from their favorites.

Main Flow:

1. While viewing a manga title or chapter, the user clicks the **"Add to Favorites"** button (e.g., a heart icon).
2. The system records this action by updating the user's profile to include the selected manga in their favorites list.
3. The user can navigate to their **Favorites** page at any time to see all saved manga. The newly added title now appears in that list, showing key info (cover, title, latest chapter, etc.).
4. (Variation – Remove favorite): On the Favorites page or manga page, if the user clicks **"Remove from Favorites"** for a particular title, the system will delete that entry from the user's favorites in the profile.
5. The favorites data is stored persistently (in the database), so even after logging out and back in, the user's favorites list remains intact.

UC6: Comment and Rate Content

Actor: Reader

Goal: Provide feedback on a manga chapter by commenting and/or rating it.

Description: This use case allows readers to interact with the community by commenting on a chapter and giving it a rating (e.g., 1–5 stars or a like/dislike). These interactions help drive user engagement and provide quality indicators for content. The system will display submitted comments to all readers and aggregate ratings to show an average score for each manga. Inappropriate content in comments should be detected or managed to maintain community standards.

Main Flow:

1. The user scrolls to the **comments section** for a chapter (usually below the manga pages or on a chapter discussion page).
2. The user enters a text comment into the comment box and optionally selects a rating (for example, 4 out of 5 stars) for the chapter.
3. The user submits their comment/review.
4. The system validates the input (e.g., checks that a comment is not empty and ensures it doesn't contain prohibited profanity or spam).
5. The system saves the comment and rating, associating it with the chapter and the user's profile.
6. The new comment appears immediately in the chapter's comment list, along with the user's name and timestamp. The chapter's overall rating is updated based on the user's input (e.g., recalculating the average rating).
7. Other readers can now see the comment and the updated rating. The system may flag or filter comments that violate content rules (either automatically or via admin review), ensuring community guidelines are upheld.

UC7: Share Manga to External Platforms

Actor: Reader

Goal: Share a manga chapter or title with others via social media or a direct link.

Description: This use case allows a reader to share content from the platform to external services, helping to attract new users and engage existing ones. The user can share a link to a manga (or a specific chapter/page) on social networks or messaging apps. The system facilitates this by providing shareable links or integrated share buttons for popular platforms.

Main Flow:

1. While reading a chapter or viewing a manga's detail page, the user clicks the **Share** button/icon.
2. The system presents sharing options (such as "Share on Facebook", "Share on Twitter", "Copy Link", etc.).
3. The user selects a platform or copy option. For example, if "Facebook" is chosen, the system might invoke a Facebook share dialog with a prepared link and preview of the manga. If "Copy Link" is chosen, the system places a direct URL to the manga/chapter page in the user's clipboard.

4. The user confirms the share action (if an external platform dialog appears) to post the content on their social feed or sends the link to others.
5. The system ensures the shared link directs recipients to the correct content on the manga platform (prompting them to register or log in if required to view the content). The share action is quick (should complete within a few seconds) to not disrupt the reading experience.

UC8: Purchase Premium Chapter Access

Actor: Reader

Goal: Pay for and access a manga chapter that requires purchase (premium content).

Description: In this use case, a reader wants to read a chapter that is marked as premium (requires payment or a subscription to access). The system uses a virtual currency (e.g., coins) or direct payment to allow the user to unlock the content. The user will go through a payment process, which could involve using a stored wallet balance or an external payment gateway (such as credit card or ZaloPay). Once payment is confirmed, the user gains immediate access to the chapter. This provides monetization for the platform and rewards content creators.

Main Flow:

1. The user attempts to open a **premium manga chapter** that is not yet unlocked on their account. The system displays the price (e.g., “5 coins” or a currency amount) and an option to purchase/unlock the chapter.
2. The user chooses to proceed with the purchase. If the platform uses an in-app coin wallet, the system will check the user’s coin balance:
 - If the user has sufficient coins, the system deducts the required number of coins for the chapter.
 - If the user does not have enough coins (or if using direct payment), the system redirects the user to a **Payment** page where they can add coins or pay using a selected method (credit card, ZaloPay, etc.). The user completes the payment transaction (which may involve entering payment details or confirming via an external payment service).
3. After a successful payment, the system marks the chapter as **unlocked** for the user.
4. The user is granted access to read the chapter immediately. The chapter’s pages load in the reader interface (as in UC4) and the user can now read them (with translation if needed) just like any free chapter.

5. The system records the transaction (for revenue tracking and for later reference in the user's purchase history). The entire purchase process is designed to be secure (using HTTPS and payment gateway security) and quick (typically completing within a few seconds).

Uploader Use Cases

(Uploaders are users who have been granted special privileges to contribute content. Before becoming an uploader, a user must go through an approval process by an admin.)

UC9: Request Uploader Role

Actor: Reader (Registered User aspiring to be Uploader)

Goal: Apply for uploader privileges in order to contribute manga content.

Description: In this use case, a regular reader who wishes to upload manga content submits a request to obtain the **Uploader** role. The platform requires such a request to be approved by an Admin to maintain content quality and security. The request typically includes some justification or agreement to terms by the user. Once the admin reviews and approves the request (see Admin use case for approval), the user is elevated to an Uploader and can then post content.

Main Flow:

1. The user (logged in as a Reader) navigates to their account settings or a dedicated **"Become an Uploader"** page.
2. The user fills out an **Uploader Request** form. This form might require certain information or confirmations, such as a statement of intent, understanding of content guidelines, or portfolio/experience (depending on platform policy).
3. The user submits the request.
4. The system records the uploader request and marks the user's status as "Pending Uploader Approval".
5. An Administrator is notified of the new request (or will see it in the admin dashboard). The user cannot upload content until approval.
6. The user is informed that their request has been sent and will be reviewed (typically within a specified timeframe, e.g., 48 hours). They must wait for the admin's decision. (The outcome of approval or rejection will be handled in the admin's use case. If approved, the user will be notified and gain Uploader capabilities.)

UC10: Upload Manga Content (New Series/Chapter)

Actor: Uploader

Goal: Submit new manga content (a series or a chapter) to the platform for readers to access.

Description: This use case allows an Uploader to add manga content to the platform. An uploader can create a new manga series (with its initial chapter) or add a new chapter to an existing series they have permission to contribute to. The process involves providing metadata about the manga/chapter and uploading image files for each page. After submission, the content does not become immediately public; it is sent for admin approval to ensure it meets guidelines. The system will handle OCR and prepare translations for the new content once approved. Uploads have size limits and must include required info so that the content is well-organized on the site.

Main Flow:

1. The uploader navigates to the **Upload** section of the site (this might be an uploader dashboard or a content management page available only to users with Uploader role).
2. The uploader chooses to create a **new series** or to add a **new chapter** to an existing series (the interface might offer both options).
 - If creating a new series: The uploader enters series information such as title, author, genre, year, and a summary description.
 - If adding a chapter to an existing series: The uploader selects the series from a list of their available titles and provides chapter-specific details (chapter number/title, brief summary).
3. The uploader uploads the manga pages for the chapter. For example, they select image files (scans) for each page, up to a defined limit (e.g., each chapter file set must be under 10MB or a certain number of pages for performance reasons).
4. The uploader submits the content for review by clicking **“Submit Chapter”** (or **“Submit Series”** if new).
5. The system performs preliminary validation: ensuring all required metadata is provided, images are in acceptable format, and the content size is within limits. If anything is missing or incorrect, the system prompts the uploader to fix it before resubmitting.
6. Upon successful submission, the new manga/chapter is saved in the system with a status of **“Pending Approval.”** The OCR and translation processing may be queued

but not published until approval. The system might automatically notify administrators of the pending content.

7. The uploader sees a confirmation that their chapter was uploaded successfully and is awaiting approval. They cannot edit the chapter while it is in review (or can edit if needed, depending on system design, but typically any changes would require re-approval).
8. (Post-condition: An admin will review the submission in a separate use case. If the admin approves, the system will publish the chapter live for readers, and possibly run the OCR/translation on the pages so that translations are ready. If rejected, the uploader might be notified of the rejection reason and possibly allowed to edit and resubmit.)

Note: Uploaders may also have capabilities to **manage their content** after posting. For example, they might be able to view a list of their uploaded chapters, see whether each is approved or still pending, and possibly withdraw or update a submission if it hasn't been approved yet. Those management actions are usually part of the uploader's dashboard but are not detailed here as separate use cases.

Admin Use Cases

UC11: Approve Uploaded Manga Content

Actor: Admin

Goal: Review new user-submitted manga content and decide whether to publish it on the platform.

Description: This use case is triggered when an uploader submits new content (a series or chapter) to the system (as described in UC10). The Admin plays the role of moderator and quality gatekeeper, checking that the content is appropriate, correctly categorized, and meets any community standards or legal requirements. The admin can either approve the content (making it live for all readers) or reject it (preventing it from being published, possibly with feedback to the uploader). This process ensures that inappropriate or low-quality content does not appear on the site without oversight.

Main Flow:

1. The admin logs into the **Admin Dashboard** of the platform.
2. The admin navigates to the **Pending Content** section, which lists all newly uploaded manga series/chapters awaiting approval.

3. The admin selects a particular submission to review. They can see the metadata (title, description, tags, etc.) and preview the pages (the uploaded images) and any extracted text via OCR.
4. The admin checks the content for compliance with guidelines (e.g., no prohibited imagery or text, correct genre classification, acceptable image quality).
5. **Approval:** If everything is in order, the admin clicks an **Approve/Publish** button. The system then changes the status of the content to **“Approved”** and makes it visible to all Readers on the platform. The manga chapter is now available in the appropriate catalog/category. (The system might also trigger background tasks to generate translated text for pages if not already done, so that readers can immediately use the auto-translate feature on this new content.)
6. **Rejection:** If the content is inappropriate or has issues, the admin can click **Reject**, optionally entering a reason or note (e.g., “violates content policy” or “pages incomplete/corrupted”). The system marks the submission as **“Rejected”** and notifies the uploader of the outcome (including the reason, if provided). Rejected content remains unavailable to readers. The uploader may be allowed to correct and re-submit the content depending on policy.
7. After either approving or rejecting, the admin continues with other pending submissions (if any). This use case may repeat for each item in the queue. The system keeps a record of the admin’s decision and the time it was made for auditing purposes.

UC12: Manage User Roles (Approve Uploader Requests)

Actor: Admin

Goal: Grant or modify user roles, particularly approving requests to become an Uploader.

Description: This use case covers the admin’s ability to manage user privileges on the platform. A common scenario is handling a reader’s request to gain uploader rights (as initiated in UC9). The admin reviews such requests and decides whether to approve them, thereby changing the user’s role. Additionally, this use case can encompass any role changes (e.g., promoting a user to moderator, or demoting an uploader back to regular reader, etc., if those roles exist in the system). It ensures that only trusted users gain content-uploading capabilities.

Main Flow:

1. The admin accesses the **User Management** section of the admin interface.

2. The system presents a list of user-related tasks, which includes pending **Uploader role requests** from users.
3. The admin selects a pending uploader request to view details. This may include the user's profile, their reason for requesting (if provided), and their activity history on the platform (to assess trustworthiness).
4. The admin makes a decision:
 - **Approve:** The admin grants the uploader role to the user. The system updates the user's account status to **Uploader** (adding the privileges to upload content). The user is notified (e.g., via notification or email) that they are now an approved uploader.
 - **Reject:** The admin denies the request. The system leaves the user as a regular Reader and may notify them that their request was not approved (optionally including a reason or a suggestion to try again later).
5. The system records the admin's action in an audit log (who approved which user, or who was rejected, and when).
6. (Additionally, the admin can use the user management interface to change roles of users directly as needed – for example, demoting an uploader if they abuse their role, or assigning other roles. These actions would follow a similar pattern: select user, choose new role, confirm change, and record it.)

UC13: Suspend or Reinstate User Account

Actor: Admin

Goal: Restrict a user's access by suspending their account, or reinstate a previously suspended account.

Description: This use case lets an admin enforce platform rules by locking or suspending user accounts that violate policies (e.g., posting inappropriate content, spamming, or other harmful behavior). When an account is suspended, the user cannot log in or use the platform. The admin can later reinstate the account if the suspension period ends or upon appeal. This is crucial for platform security and community health.

Main Flow:

1. The admin navigates to the **User Accounts** section of the admin dashboard (which lists all registered users and their statuses).
2. The admin searches for or selects the account of the user who needs to be managed.

3. To **suspend** an account: The admin clicks a **“Suspend User”** action for that account. The system asks for confirmation (and possibly a reason and duration for the suspension). The admin confirms the action.
4. The system updates the user’s status to **Suspended/Locked**, preventing that user from logging in or accessing any content. The suspension details (who performed it, when, reason) are recorded. Optionally, the user may receive a notification or email informing them that their account is suspended and why.
5. (Optional) To **reinstate** an account: The admin selects a suspended user and clicks **“Reinstate/Unlock”**. After confirmation, the system reactivates the user’s account, restoring their access. This is typically done after a suspension period lapses or if an appeal is granted.
6. The admin verifies that the user’s status is now updated accordingly. The system ensures that only admins can perform this action and that all such actions are logged for security auditing.

UC14: Manage Manga Database (Edit/Delete/Feature Content)

Actor: Admin

Goal: Modify or organize manga content on the platform, including updating details, removing content, or featuring content on the homepage.

Description: This use case gives the admin control over existing manga entries in the system. Admins can edit metadata (e.g., correct a title or genre), delete content that is found to be inappropriate or requested for removal, and mark certain manga or chapters as **“Highlighted”** or **“Featured”**. Featuring content means promoting it on the homepage or special sections (often used for popular or new releases) for a set duration. By managing the manga database, admins keep the content well-organized, up-to-date, and can curate what readers see prominently.

Main Flow:

1. The admin accesses the **Content Management** section of the admin panel, which lists all manga series and chapters in the database (possibly with search and filter options).
2. The admin selects a specific manga title or chapter to manage. They are presented with options such as **Edit**, **Delete**, or **Feature**.
3. **Edit Metadata:** The admin chooses to edit details of a manga/chapter (for example, to fix a typo in the title or update tags). The admin makes the necessary changes in

an edit form and saves. The system validates the input and updates the database record. Readers will now see the updated information on the site.

4. **Delete/Remove Content:** The admin opts to delete a manga series or a specific chapter (e.g., if it violates policies or by request of copyright holder). The system asks for confirmation because deletion is destructive. Upon confirming, the system removes or archives the content. The manga/chapter no longer appears for readers. (The system might remove all associated data like images and comments, or mark it as removed with a placeholder message for transparency.)
5. **Feature Content:** The admin marks a manga or a chapter as “Featured/Highlighted”. The system then flags this content to appear in a special **Featured section** on the homepage or at the top of search results, etc. While featuring, the admin can set a duration (for example, featured for 7 days by default). The system schedules an end-date for the feature. Featured content is typically chosen based on certain criteria (like high ratings or strategic promotion). Once featured, readers visiting the site will prominently see this content.
6. The system ensures only admins can perform these modifications. All changes (edits, deletions, featuring actions) are logged. Featured content automatically expires after the set period, after which the system removes its highlighted status unless the admin extends it.

Director Use Cases

UC15: View User Visit Statistics

Actor: Director

Goal: See analytics on user traffic and usage of the platform.

Description: This use case allows a Director or high-level admin to view aggregated statistics about the platform’s usage. The focus is on metrics like the number of users visiting over time, page views, popular titles, active readers, etc. These analytics help in assessing the platform’s growth and user engagement. The data should be presented in a clear format (dashboards or reports) and be up-to-date. Performance is important; even with large data, the reports should load quickly.

Main Flow:

1. The director accesses the **Analytics/Reports** section of the platform (likely a special dashboard available to directors or senior admins).
2. The director selects the **User Visits Statistics** report or dashboard. (They may choose a timeframe like daily, monthly, yearly, or real-time overview.)

3. The system gathers the relevant data from the database (and/or analytics tracking tools): e.g., number of unique visitors, total site visits, average session duration, geographic distribution of users, etc., depending on what analytics are tracked.
4. The system displays the statistics in a human-friendly format, which could include charts, graphs, and key numerical indicators. For example, a graph of daily active users over the past month, or a pie chart of reader vs uploader percentages.
5. The director reviews the data. The interface allows further filtering or drilling down if needed (e.g., view by specific date range or segment).
6. The system ensures the data shown is accurate and updated to the latest possible timeframe. The retrieval and rendering of the report happen within a few seconds for a smooth experience. The director can use this information to make strategic decisions about the platform.

UC16: View Revenue Reports

Actor: Director

Goal: Review the platform's revenue and financial statistics (e.g., income from premium chapters).

Description: In this use case, the Director views reports on revenue generated by the platform. This includes money earned from users purchasing premium chapters (or coin purchases) and possibly payouts or rewards given to uploaders if applicable. The report is usually organized by time period (monthly, yearly) to show trends and totals. It helps stakeholders understand the financial performance of the project.

Main Flow:

1. The director goes to the **Finance/Revenue** section of the reporting dashboard.
2. They select a **Revenue Report**, possibly choosing the period of interest (for instance, "Monthly Revenue - Last 12 months" or "Year-to-Date Revenue").
3. The system retrieves financial data, such as total earnings from user purchases, the number of transactions, average revenue per user, etc., from the payment records in the database. It may also compute uploader rewards paid out, to net the profit if needed.
4. The system displays the revenue information in a report format. This could include a table of months with corresponding revenue figures, and/or a line chart showing revenue growth over time. Key figures like total revenue for the selected period and comparisons to previous periods might be highlighted.

5. The director examines the revenue metrics. For example, they can see that “Revenue for November 2025 = 500 USD, up 10% from October” (or the equivalent in platform currency/coins). Annual revenue can also be viewed to gauge overall performance.
6. As with the stats dashboard, the system ensures the revenue report loads quickly (e.g., within 5 seconds) and accurately. All financial data is up-to-date up to the last completed transactions. These insights enable the director to make informed decisions about monetization strategies, marketing, and resource allocation for the platform.