Ho Chi Minh University of Science

Department of Information Technology

Software Requirements Specification

Course:	**Introduction to Software Engineering**
Instructor:	**Mr. Truong Phuoc Loc**
Class:	**23CLC03**
Group:	**07**
Students' name:	**Le Anh Duy – 23127011**
	Tran Gia Huy – 23127199
	Nguyen Thanh Luan – 23127296
	Nguyen Thanh Tien – 23127539

Ho Chi Minh City, November 2025

Table of Contents

1 Member Contribution Assessment

ID	Name	Contribution (%)	Signature
23127011	Le Anh Duy	100%	duy
23127199	Tran Gia Huy	100%	Huy
23127296	Nguyen Thanh Luan	100%	Luan
23127539	Nguyen Thanh Tien	100%	Tien

2 Problem Statement

2.1 Introduction

The proposed software system is a modern web-based application designed to provide users with an efficient, and digital manga reading experience. The system aims to address the need for a responsive and interactive platform that allows users to read manga from multiple countries and languages, while removing language barriers through integrated optical character recognition (OCR) and machine translation. The platform focuses on delivering real-time translation of manga pages, reliable content management for uploaders and administrators, and secure access control for all users.

This project integrates both frontend and backend technologies to deliver a full-stack solution suitable for deployment on cloud environments. The application is expected to support a diverse range of users (readers, uploaders, and admins), handle dynamic content such as manga chapters and comments, and ensure secure storage and retrieval of user data, translations, and payment information. A strong emphasis is placed on user experience, performance, scalability, and maintainability, so that the system can grow in terms of content volume, number of users, and supported languages.

2.2 Operating Environment

2.2.1. Client Side (Frontend) Environment

The system's user interface runs entirely in modern web browsers that support HTML5, CSS3, and ES6+ JavaScript. It is optimized for use on:
– Google Chrome
– Mozilla Firefox
– Microsoft Edge
– Safari (latest versions)

The Frontend layer is built using:
– **Next.js**: A React-based framework that provides routing, client-side rendering (CSR), and optional server-side rendering features. In this project, Next.js is primarily used in CSR mode to deliver a smooth reading experience, enabling fast navigation between pages and chapters without full page reloads.
– **Tailwind CSS**: A utility-first CSS framework that simplifies styling and enables high responsiveness across desktops, tablets, and mobile devices. This ensures that the manga reader, admin dashboard, and uploader interfaces are all mobile-friendly.

Together, these tools provide a responsive, mobile-friendly, and visually cohesive user interface for both the reader subsystem (manga browsing, reading, translation selection, comments, ratings) and the admin/uploader subsystem (content management, user management, statistics).

2.2.2. Server Side (Backend) Environment

The backend service is implemented using Node.js and the Express.js framework. It runs in a controlled server environment that encapsulates all dependencies and configuration using environment variables. This ensures consistent behavior across development, staging, and production deployments.

The backend environment includes
- Node.js (version 18.x or later) as the runtime environment for executing JavaScript/TypeScript code on the server.
- Express.js as the web application framework, handling routing, middleware, business logic, and RESTful APIs for both the reader and admin subsystems.
- A modular architecture that separates:
- Route definitions (API endpoints for readers, uploaders, and admins)
- Controllers (business logic such as chapter approval, translation requests, payment processing)
- Services (integration with OCR/translation APIs, email/OTP sending, payment gateways)
- Data access layer (queries to PostgreSQL using an ORM or query builder).

The backend is deployable on any Node.js-compatible hosting platform (e.g., Render, Railway, AWS, or similar cloud providers). In development, local machines or containerized environments (e.g., Docker) can be used to simulate the production setup. This configuration allows the backend to run independently of the local system's global Node.js installation and guarantees consistent execution across deployment targets.

2.2.3. Database Environment

To store and manage structured data, the system uses:
– **PostgreSQL**: A relational database management system (RDBMS) used to store users, roles, manga metadata, chapters, page URLs, translation records, reading progress, comments, ratings, and payment transactions. The schema is normalized to ensure data consistency and efficient querying.
– An optional **ORM or query builder** (Knex) may be used on the backend to define models, manage migrations, and simplify interactions with PostgreSQL, providing type safety and maintainable database access code.

The PostgreSQL database is hosted on a managed cloud service (Supabase,), offering:
– Automated backups and restore options
– High availability and fault tolerance
– Secure access control with user roles, SSL connections, and network-level restrictions

By leveraging PostgreSQL, the system benefits from strong transactional guarantees, powerful relational queries (e.g., joins for chapters and translations), and enforceable constraints (foreign keys, indexes) that maintain data integrity as the platform scales.

2.2.4. Cloud and Media Storage Environment

Manga page images and other media assets are uploaded and stored using a dedicated cloud storage service, such as:
– Cloudinary or a similar cloud-based media management service, which provides secure upload APIs for handling manga page images.
– Offers CDN delivery for fast global distribution of manga images
– Provides automatic image optimization, resizing, and transformation
– Ensures high-performance content delivery and caching
– Protects media uploads through API keys, secure tokens, and access control rules

After a file is uploaded, the storage service returns a secure URL; the system stores this URL inside PostgreSQL

This setup reduces server load, prevents storage bottlenecks, and guarantees efficient, scalable handling of high volumes of manga pages. The same URLs are also used when sending pages to the external OCR service, so the OCR engine can fetch the image directly from the cloud.

2.2.5. External OCR and Translation Service Environment

The system integrates with a third-party OCR and machine translation service, which runs as a separate external service and is accessed via secure HTTP APIs.

– The OCR service receives the manga page image (via URL) and returns the detected text and optional layout information (e.g., bounding boxes).
– The translation service receives the extracted source text and target language code, then returns the translated text.
– These services may be hosted by external providers (e.g., cloud AI APIs) or by a dedicated machine-learning microservice maintained by the project team.

All communication between the backend and the OCR/translation service must use HTTPS, API keys, and rate limiting to ensure security and stability.

2.3 Design and Implementation Constraints

2.3.1. Programming Languages

– Frontend: JavaScript/TypeScript (Next.js), HTML5, CSS3
– Backend: JavaScript/TypeScript (Node.js with Express.js)
– Database: SQL (PostgreSQL dialect) for schema definition and queries

2.3.2. Framework & Library Constraints

–The frontend must be developed using Next.js, leveraging its routing system, component-based architecture, and client-side rendering capabilities to ensure high performance and an optimal reading experience (e.g., preloading pages, smooth navigation, dynamic language switching).
– Tailwind CSS must be applied as the primary styling framework, providing utility-first classes for consistent, maintainable, and responsive UI design across reader, uploader, and admin interfaces.
– The backend must be implemented using Express.js exclusively as the core HTTP framework..
– RESTful design principles must be followed for all APIs, including clear resource-based endpoints, use of standard HTTP methods, and stateless request handling.

2.3.3. Database Constraints

–The database must be PostgreSQL, hosted on Supabase platform that allows collaboration among developers.
– Data must be stored in relational tables with clearly defined primary keys, foreign keys, and appropriate indexing to guarantee performance and data integrity.
– Migrations must be used to evolve the database schema in a controlled manner across development and production environments.

2.3.4. Security Constraints

– User authentication must be implemented securely using hashed passwords (e.g., bcrypt) and JWT (JSON Web Tokens) or secure session mechanisms. When JWTs are used, they should be stored in HttpOnly cookies, preventing client-side scripts from accessing authentication tokens and reducing exposure to potential attacks.
– After registration, an OTP (one-time password) must be generated and sent to the user's registered email address. OTPs must have a limited validity period and be stored securely (e.g., hashed or tokenized) in the database. Accounts that are not verified via OTP must remain in a "Pending Verification" state and must not have access to protected functionality.
– API keys and sensitive configuration values (PostgreSQL connection strings, media storage credentials, OCR/translation API keys, payment gateway secrets) must be stored exclusively in environment variable files (.env) and never hardcoded in the codebase or committed to version control.

– CORS (Cross-Origin Resource Sharing) policies must be properly configured to allow controlled communication between the frontend (Next.js) and backend (Express.js) applications. Only trusted domains are permitted, reducing the risk of unauthorized cross-origin requests. In development, only http://localhost:3000 and the local backend domain are allowed; in production, the rules must be updated to include the deployed frontend domain.
– All communication that involves user credentials, OTP verification, translation requests, and payment operations must occur over HTTPS to ensure encryption in transit and protect against man-in-the-middle attacks.

3 Requirements Overview

3.1 Stakeholders

STT	Stakeholder	Description
1	Director	- View website user visit statistics. - View monthly and yearly revenue reports.
2	Admin	- Add, update, and delete manga (comic management). - Assign roles to each member. - Approve new manga and review content before publishing. - Manage featured and suggested mangas displayed on the homepage. - Lock or suspend user accounts. - Approve user requests to become uploaders.
3	Uploader	- Upload mangas
4	User	- Register/login (Google, Facebook, etc.). - Ensure passwords are one-way encrypted. - Search for comics by author, genre, or publication year. - Apply to become an uploader. - Share comics on social media platforms.
5	Reader	- Use automatic comic translation. - Save comics to the favorites list. - Save reading progress (auto-update). - Comment and rate comic quality. - Make payments to read comics (multiple payment methods supported).
6	Purchase system	- Proceed transaction between reader and uploader

7	Translating system	- Provide translating version for mangas

3.2 Requirements

3.2.1. Functional Requirements Specification

Order	Functional requirement	Description
1	User visit statistics	Provide a dashboard for director to manage users activities and revenue report
	Revenue report	
2	Comic Management	Admin can review, delete, and update mangas
	Review content before publishing	Admin can review new managas and aprove it
5	Manage featured and suggested mangas	Admin can set featured or suggested stories to display on the home page
6	User authorization	Admin can assign roles to users
	User management	Response time less than or equal to 2 seconds; record operation history
8	Register uploader	User can register to become an uploader to have the right to post
9	Post mangas	Uploader can post stories.
10	Account Registration/Login	User can login/register an account (google, Facebook,...)
	Password security	The password to be encrypted and unreadable.
11	Search for comics	Users can find the author, genre, year of publication.
	Automatic comic translation	Readers have an automatic comic translation function.

	Manage your favorites list	Readers can save comics to their favorites list
	Save reading progress	Readers can save current progress of comics to read in the future.
12	Comments and reviews	Readers can comment and rate the quality.
13	Payment function	Readers need to pay for the story
14	Share content	Users can share mangas on social networks

3.2.2. Non-Functional Requirements Specification

Order	Non-functional requirement
1	Reports and statistics in director dashboard must be displayed in 5s
2	Admin operations must be applied in 2s
3	The system must ensure that only administrators can authorize.
4	All changes of manga must be updated immediately on the website in 2s
5	Sent to the administrator for approval less than a minute; response within 48 hours; only approved people have the right to post.
6.	Easy-to-use posting interface; upload stories less than 10MB/chapter; automatically send approval requests to administrators
7	Authentication must take ≤ 3 seconds, login data is secured with HTTPS
8	Passwords must be one-way encrypted (bcrypt or equivalent)
9	All results returned by the website must fast (response time less than 5s)
10	Favorite list data must be saved and not lost after logging out.
11	Reading progress should be updated automatically every 30 seconds or when changing

	pages (avoid spam)
12	Feedback must be displayed immediately after submission; the system must check offending content.
13	Transactions must be safe and secure; payment less than 5 seconds; support multiple methods (cards, Zalo-pay, etc.)
14	Get a shared link via platforms (FB, Zalo,...) in 2s

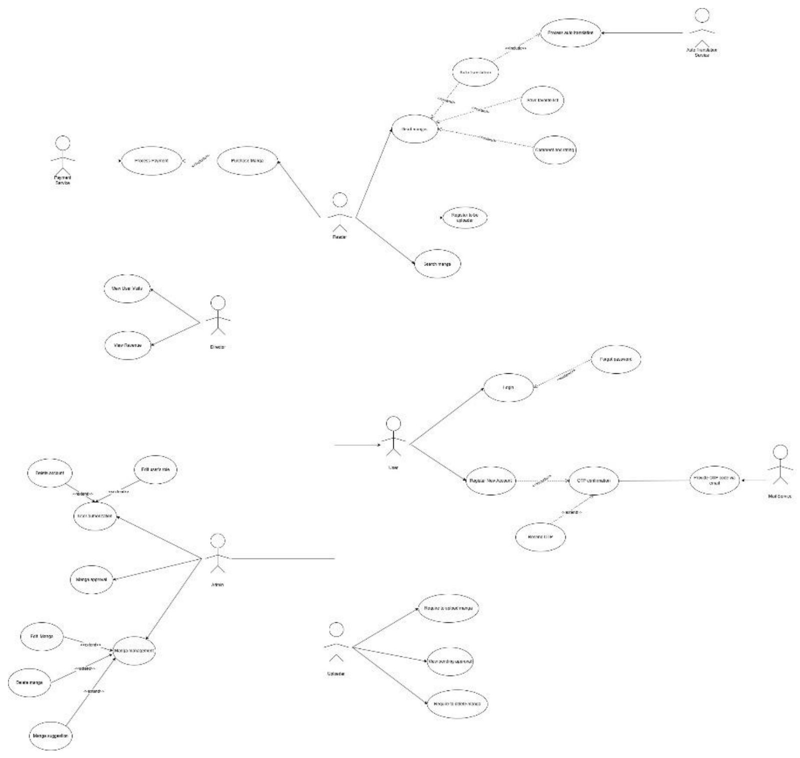
4 Requirements Analysis

4.1 *Use Case mode*

4.1.1. Use case link

Use Case diagram

4.1.2. Screenshots



4.2 Use Case Specification

4.2.1. Use Case 1

Use case name	Registration
Brief description	This case allows a new user (reader or potential uploader) to create an account on the platform. Registration ensures that the user's personal information is securely stored and encrypted, enabling access to features such as reading manga, tracking reading progress, commenting, rating, and later becoming an uploader.
Basic flow	1. User navigates to the website and selects "Register." 2. System displays the registration form. 3. User fills in required information. 4. System validates the input. 5. System encrypts and stores user information in the database. System confirms successful registration and may log the user in automatically.
Alternative flow	3.b Duplicate account: System notifies the user that the

	username/email is already in use and prompts to enter a different one. 4.b Invalid input: System prompts the user to correct invalid email or weak password.
Special requirements	● All sensitive data (passwords, email) must be encrypted. ● Registration should comply with data privacy and security standards. ● Input validation to prevent SQL injection or other attacks.
Preconditions	● User does not already have an account. ● User has access to the registration page.
Post conditions	● User accounts are created and stored securely. ● User may be logged in automatically (optional).
Extension points	● Login: After registration, user can proceed to login if not logged in automatically.

4.2.2. Use Case 2

Use case name	Login
Brief description	This use case allows a registered user to access their account on the platform. Logging in enables users to read manga, track reading progress, comment, rate chapters, share content, and access uploader features if permitted.
Basic flow	1. User navigates to the website and selects "Login." 2. System displays the login form. 3. User enters credentials: ● Username or email ● Password 4. System validates the credentials. 5. If credentials are correct, the system grants access and loads the user's personalized account features (reading history, translation settings, comments, ratings, etc.).
Alternative flow	4.b Invalid credentials: System displays an error message and prompts the user to retry. 2.b Forgot password: User requests a password reset; the system sends a verification link or code to reset the password. 4.c Account locked: If the account is locked or suspended, system notifies the user and prevents login.

Special requirements	● Login attempts should be protected against brute-force attacks. ● User passwords must be stored and verified securely (e.g., using encryption and hashing)..
Preconditions	● User has already registered an account. ● User is not currently logged in.
Post conditions	● User is logged in, and a session is active. ● Personalized features, including reading progress, translation options, and uploader permissions, are accessible.

4.2.3. Use Case 3

Use case name	**Search for Manga**
Brief description	This use case allows a user (reader) to search for manga chapters or titles using keywords, filters, or categories. The system returns relevant results, enabling the user to quickly find manga they want to read.
Basic flow	1. User navigates to the website and selects the search bar or search page. 2. User enters keywords (e.g., manga title, genre, author) and optionally selects filters (genre, language, release date). 3. System processes the input and searches the database for matching manga titles and chapters. 4. System displays a list of search results, showing basic information for each manga: title, genre, number of chapters, rating, and short summary. 5. User selects a manga from the results to view details or start reading.
Alternative flow	4b. No results found: System displays a message and may suggest popular or similar manga. 3b. Invalid input: System prompts the user to correct empty or invalid search queries.
Special requirements	● Search should be fast and scalable, even with a large database of manga chapters. ● Filters should allow users to narrow effectively (genre, language, popularity, rating). ● Search results should be ranked by relevance, popularity, or user ratings.

Preconditions	● User has access to the platform. ● Manga data (titles, chapters, metadata) exist in the system.
Post conditions	● User receives relevant search results. ● Users may navigate to manga details, read chapters, or apply further filters.

4.2.4. *Use Case 4*

Use case name	Read Manga
Brief description	This use case allows a user (reader) to open and read a manga chapter. The system displays pages, provides automatic translation options, tracks reading progress, and allows interaction through comments, ratings, and sharing.
Basic flow	1. User selects a manga chapter to read from search results, recommendations, or the homepage. 2. System loads the chapter pages. 3. User navigates pages using next/previous controls. 4. System provides translation options: ● If the user chooses a language, the system sends pages to the translation module (OCR + machine translation). ● Translated text is displayed on each page in real time. 5. System tracks user reading progress: current chapter, last page read, and translation status.
Alternative flow	4b. Translation service unavailable: System displays original language or a cached translation. 2b. Pagination errors or missing pages: System notifies user and skips missing pages if possible.
Special requirements	● Translation must be accurate and fast, using OCR + machine translation. ● Comments and ratings must be properly stored and displayed in real time.
Preconditions	● User is registered and logged in. ● Manga chapter exists and is approved/published by admin.
Post conditions	● User's reading progress is updated. ● User can interact with the chapter via comments, ratings, and sharing.

4.2.5. *Use Case 5*

Use case name	Post Manga
Brief description	This use case allows a registered user with the uploader role to submit a new manga chapter to the system. The uploader provides metadata and uploads chapter pages, after which the admin reviews and approves the content for publication.
Basic flow	1. Uploader logs in and navigates to the "Upload Manga" section. 2. Uploader provides descriptive information for the chapter: 3. Uploader uploads chapter pages (images). 4. System validates the input. 5. System submits the chapter for admin approval. 6. Admin reviews the content and either approves or rejects it. 7. If approved, the chapter is published and becomes visible to readers.
Alternative flow	4b. Missing or invalid information: System prompts uploaders to complete or correct the metadata. 7b. Rejected by admin: Admin provides feedback; uploader can revise and resubmit. 3b. File upload errors: System notifies uploaders and allows retry.
Special requirements	● Uploaded files must meet platform format and size requirements. ● Metadata must be accurate and complete for indexing and search.
Preconditions	● User must be registered and have uploader role. ● User must be logged in.
Post conditions	● Manga chapter is submitted to the system. ● Admin receives notification for review. ● If approved, chapter is published and available to readers.

4.2.6. *Use Case 6*

Use case name	Comment and Reviews
Brief description	This use case allows registered users (readers) to provide feedback on manga chapters by posting comments and submitting ratings. These interactions help other users evaluate the content and contribute to community engagement.
Basic flow	1. User opens a manga chapter. 2. System displays the chapter along with existing comments and ratings. 3. User selects the option to leave a comment or rating. 4. User enters a comment and/or selects a rating (e.g., 1–5 stars). 5. System validates the input (e.g., no empty comments, ratings within allowed range). 6. System stores the comment and rating in the database. 7. System updates the chapter's overall rating and displays the new comment to other users in real time.
Alternative flow	4b. Empty or invalid comment/rating: System prompts the user to correct input. 1b. User not logged in: System prompts user to log in or register before submitting comments or ratings.
Special requirements	● Comments must be stored securely and comply with content moderation policies. ● Comments and ratings must be synchronized in real time for all users viewing the chapter. ● Support for sorting/filtering comments (e.g., most recent, highest rated).
Preconditions	● User is registered and logged in. ● Manga chapter exists and is accessible.
Post conditions	● User comment and rating are stored and visible to other users. ● Chapter's average rating is updated. ● Community feedback is enhanced, supporting decisions like highlighting popular chapters.

4.2.7. *Use Case 7*

Use case name	User Authorization
Brief description	This use case manages user roles and access rights within the platform. It ensures that only authorized users can perform specific actions such as uploading manga, approving content, or accessing admin features.
Basic flow	1. User logs in to the platform. 2. System retrieves the user's account information and role(s) (e.g., reader, uploader, admin). 3. System grants or restricts access to platform features based on the user's role. 4. User attempts an action; system verifies if the action is allowed for their role. 5. If authorized, the action proceeds; if not, system denies access and displays an error message.
Alternative flow	5b. Unauthorized action: System displays a message explaining insufficient permissions. 3b. Account locked or suspended: System prevents login or specific actions for security reasons.
Special requirements	● Role-based access control must be enforced consistently across all system features. ● Sensitive actions (e.g., approving content, managing users) must be restricted to admin accounts.
Preconditions	● User is registered and logged in. ● User's role is defined in the system.
Post conditions	● User can only perform actions permitted by their role. ● Unauthorized attempts are blocked and logged.

4.2.8. *Use Case 8*

Use case name	Register for Uploading
Brief description	This use case allows a registered reader to request permission to become an uploader. Once approved by the admin, the user gains the ability to submit manga chapters, manage their uploads, and edit content.
Basic flow	1. User logs in as a registered reader. 2. User navigates to the "Request Uploader Role" or "Register for Uploading" section. 3. User submits a request to become an uploader.

	4. System validates the request (e.g., ensures the user has a verified account). 5. Admin receives the request and reviews the user's eligibility. 6. Admin approves the request. 7. System updates the user's role to uploader, granting access to upload and manage manga chapters. 8. System notifies the user that their uploader role has been approved.
Alternative flow	6b. Request denied: Admin rejects the request, and system notifies the user with a reason. 4b. Incomplete or invalid request: System prompts the user to complete necessary information before submission. 7b. Already an uploader: System informs the user they already have uploading privileges.
Special requirements	● Only verified users can submit a request to become uploaders. ● Notifications to users about approval or rejection should be immediate.
Preconditions	● User is registered and logged in. ● User does not already have the uploader role.
Post conditions	● User's role is updated to uploader if approved. ● User can access uploader functionalities, such as posting manga chapters.

4.2.9. Use Case 9

Use case name	**Payment for Manga**
Brief description	This use case allows a registered user (reader) to pay for access to specific manga chapters using supported payment methods. Payment ensures that the user can read premium content while enabling monetization for the platform and uploaders.
Basic flow	1. User selects a manga chapter marked as premium or requiring payment. 2. System displays the payment options (e.g., credit card, ZaloPay, or other supported methods). 3. User selects a payment method and enters required payment information. 4. System validates the payment information and processes the transaction. 5. Upon successful payment, system grants the user access to the paid manga chapter. 6. System records the transaction in the user's account and updates the chapter access status. 7. User can now read the chapter without restrictions.
Alternative flow	4b. Payment failure: System displays an error message (e.g., insufficient funds, invalid card) and allows the user to retry. 4c. Payment cancellation: User cancels the transaction before completion; access is not granted.
Special requirements	● System must ensure payment confirmation before granting content access. ● Transaction history must be stored and retrievable for both users and admins. ● Support for multiple payment methods and currencies as applicable.
Preconditions	● User is registered and logged in. ● Manga chapter requires payment.
Post conditions	● User is granted access to the paid manga chapter. ● Payment transaction is recorded in the system.

4.2.10. Use Case 10

Use case name	**Users Visit Tracking**
Brief description	This use case allows the system to monitor and record user activity on the platform, including page visits, manga chapters viewed, reading duration, and interactions such as comments or ratings. Admins can analyze this data to understand user behavior and optimize content recommendations.
Basic flow	1. User logs in or accesses the platform as a guest. 2. User navigates through manga chapters, searches for titles, or interacts with content. 3. System records key user activity. 4. System stores visit data in a tracking database. 5. Admin can view aggregated analytics via the "Users Visit Tracking" dashboard, including trends, popular chapters, and engagement statistics.
Alternative flow	1b. Guest user activity: System tracks basic visit data (e.g., pages visited) but cannot associate it with a registered user account. 5b. Data unavailability: If tracking fails temporarily, system logs the error and continues recording future activity.
Special requirements	● Tracking statistics should be accurate. ● Analytics should support filtering by time range, manga title, or user segment. ● System must store historical data for trend analysis.
Preconditions	● User accesses the platform. ● Tracking system is active.
Post conditions	● User accesses the platform. ● Tracking system is active.

4.2.11. Use Case 11

Use case name	Manga Management
Brief description	This use case allows admins to manage manga content on the platform. Admins can approve, modify, highlight, recommend, or remove manga chapters, ensuring content quality, relevance, and proper organization for readers.
Basic flow	1. Admin logs in and navigates to the "Manga Management" section. 2. System displays a list of all manga chapters, including their metadata and status (e.g., pending approval, published, highlighted). 3. Admin selects a manga chapter to manage. 4. Admin can perform actions such as add, update and delete manga. 5. System updates the manga records and notifies relevant users (e.g., uploader notified upon approval or rejection).
Alternative flow	5b. Invalid edits: System prompts admin to correct incomplete or invalid metadata.
Special requirements	● Only authorized admins can access manga management functions. ● Changes must be logged for auditing purposes. ● System must handle bulk actions efficiently (e.g., approving multiple chapters at once). ● Notifications to uploaders should be automated and clear.
Preconditions	● Admin is logged in and authorized. ● Manga chapters exist in the system (uploaded by users).
Post conditions	● Manga chapters are properly reviewed, updated, and published according to admin actions. ● Readers see updated content, highlights, and recommendations.

4.2.12. Use Case 12

Use case name	Manga Approval
Brief description	This use case allows admins to review newly uploaded manga chapters and determine whether they are suitable for publication. Approval ensures that all content on the platform meets quality standards, community guidelines, and legal requirements.
Basic flow	1. Admin logs in and navigates to the "Pending Manga Approval" section. 2. System displays a list of uploaded manga chapters awaiting review, including metadata and uploaded pages. 3. Admin selects a chapter to review. 4. Admin examines the chapter for: accuracy, image quality,.. 5. Admin takes one of the following actions: approve and reject. 6. System records the decision and updates the status of the chapter.
Alternative flow	5b. Incomplete submission: System prompts admin to request corrections from the uploader. 5c. Content violation detected: System flags chapter for further review or automatic rejection.
Special requirements	● Only authorized admins can perform approvals. ● The system must log all approval actions for accountability. ● Notifications to uploaders must be automatic and provide clear feedback. ● The review process should be efficient to minimize delays in content publishing.
Preconditions	● Manga chapter has been uploaded by an uploader and is awaiting review. ● Admin is logged in with sufficient permissions.
Post conditions	● Manga chapter status is updated (approved or rejected). ● Approved chapters are published and visible to readers ● Uploaders are notified of the decision.

4.2.13. *Use Case 13*

Use case name	Share Content
Brief description	This use case allows users (readers) to share manga chapters or pages with external platforms, such as social media, messaging apps, or via direct links. Sharing increases engagement, promotes content, and helps attract new users to the platform.
Basic flow	1. User opens a manga chapter they want to share. 2. User selects the "Share" option. 3. System displays available sharing methods (e.g., Facebook, Twitter, WhatsApp, copy link). 4. User selects a method and confirms the action. System generates the appropriate shareable link or content preview. 5. System sends the content to the selected platform or copies the link to the clipboard. 6. User successfully shares the manga chapter externally.
Alternative flow	4b. Unsupported platform: System notifies the user if the selected sharing method is unavailable. 1b. Guest user: System may prompt the user to log in or register before sharing.
Special requirements	● Sharing should be simple and fast, without interrupting the reading experience. ● Shared content must include correct references (manga title, chapter, platform URL) to avoid broken links. ● System should prevent unauthorized redistribution of premium content (e.g., require login or limit sharing for paid chapters).
Preconditions	● User is registered and logged in (optional if guest sharing is supported). ● Manga chapter exists and is accessible to the user.
Post conditions	● Manga chapter is shared successfully to the selected platform. ● Sharing activity may be tracked for analytics purposes.

4.2.14. *Use Case 14*

Use case name	Revenue Tracking
Brief description	This use case allows admins to monitor and analyze revenue generated from paid manga chapters. It provides insights into transactions, earnings from readers, and payouts to uploaders, helping with financial management and business decisions.
Basic flow	1. Admin logs into the platform with proper authorization. 2. Admin navigates to the “Revenue Tracking” or “Financial Dashboard” section. 3. System retrieves and displays transaction data. 4. Admin can filter or sort revenue data by date, manga title, uploader, or payment method. 5. System generates reports or visual charts to summarize revenue statistics. 6. Admin can export data for accounting or further analysis.
Alternative flow	5b. Partial data load: If some transaction data is unavailable, system displays available data with a warning.
Special requirements	● Sharing should be simple and fast, without interrupting the reading experience. ● Shared content must include correct references (manga title, chapter, platform URL) to avoid broken links. ● System should prevent unauthorized redistribution of premium content (e.g., require login or limit sharing for paid chapters).
Preconditions	● User is registered and logged in (optional if guest sharing is supported). ● Manga chapter exists and is accessible to the user.
Post conditions	● Manga chapter is shared successfully to the selected platform. ● Sharing activity may be tracked for analytics purposes.

4.2.15. *Use Case 15*

Use case name	Favourite Manga Management
Brief description	This use case allows users (readers) to manage their personal list of favorite manga chapters or titles. Users can add, remove, or view favorite manga, helping them quickly access content they enjoy.
Basic flow	1. User logs in to their account. 2. User navigates to a manga chapter or manga title they want to mark as a favorite. 3. User selects the "Add to Favorites" option. 4. System adds the manga to the user's favorite list. 5. User can view their favorite manga list via a dedicated section or dashboard. 6. User can remove manga from favorites if desired. 7. System updates the favorite list and ensures the changes are saved in the database.
Alternative flow	4b. Already in favorites: System notifies the user that the manga is already in their favorite list. 6b. Invalid removal: System prompts the user if trying to remove a manga that is not in the favorite list.
Special requirements	● Favorite list must be persistent across sessions and devices. ● Operations (add/remove) should be fast and synchronized with the user account. ● System should allow sorting or filtering within the favorites list (e.g., by recently added, genre).
Preconditions	● User is registered and logged in. ● Manga exists in the system.
Post conditions	● Manga is added to or removed from the user's favorite list. ● User can easily access favorite manga in future sessions.

4.2.16. Use Case 16

Use case name	Manga Suggestion
Brief description	This use case allows admins to suggest or highlight specific manga chapters on the platform, increasing visibility for selected content. Suggested manga can appear on the homepage, in special sections, or be recommended to readers based on popularity, ratings, or editorial choice.
Basic flow	1. Admin logs in and navigates to the "Manga Suggestion" or "Content Highlighting" section. 2. System displays a list of all published manga chapters with metadata (title, genre, views, ratings). 3. Admin selects manga chapters to suggest or highlight. 4. Admin sets optional parameters: placement, start and end date,.. 5. System updates the display settings and ensures the selected manga appear in the suggested sections. Readers visiting the platform see the suggested manga in the highlighted locations.
Alternative flow	3b. Conflict with existing highlights: System warns admin if multiple manga are competing for limited featured slots. 4b. Invalid dates: System prompts admin to correct invalid start/end dates.
Special requirements	● Only authorized admins can suggest or highlight manga. ● System must ensure that highlighted manga are visible in the intended sections without errors. ● Suggested manga should be displayed consistently across devices and user sessions. ● Admin actions must be logged for accountability.
Preconditions	● Manga chapter is uploaded and approved for publication. ● Admin is logged in and authorized.
Post conditions	● Selected manga chapters are displayed in suggested or highlighted sections. ● Readers are exposed to promoted content.

4.2.17. Use Case 17

Use case name	Manga Auto-Translation
Brief description	This use case allows a reader to view a manga chapter in their chosen language. The system uses OCR to detect text in manga images, sends the extracted text to a translation engine, and displays the translated text on the manga page in real time.
Basic flow	1. Reader opens a manga chapter. 2. Reader selects a translation language from the available options. 3. System loads the manga page. 4. System applies OCR to extract text from the manga page. 5. System sends the extracted text to a machine translation service (internal or third-party API). 6. System receives the translated text. 7. System overlays or replaces the original text with the translated version on the displayed manga page. 8. Reader views the translated manga page.
Alternative flow	5b. Translation Language Not Supported: System displays a message indicating the selected language is unavailable and suggests supported options. 5c. OCR Fails to Recognize Text: System displays the manga page in the original language and optionally allows the reader to request manual translation. 5d. External Translation API Unavailable: System informs the reader of temporary translation issues and allows retry.
Special requirements	● OCR must handle various font styles, handwritten text, and Japanese/Chinese/Korean characters. ● Translation must be reasonably fast to maintain reading experience. ● If third-party translation services are used, secure API integration is required. ● System should cache translated pages to avoid repeated translation requests.
Preconditions	● Reader is logged in. ● Manga chapter exists and contains translatable text. ● Translation engine or API is configured and accessible.
Post conditions	● Readers successfully view the manga in their selected language. ● Optionally, translated results may be stored for reuse to improve performance.

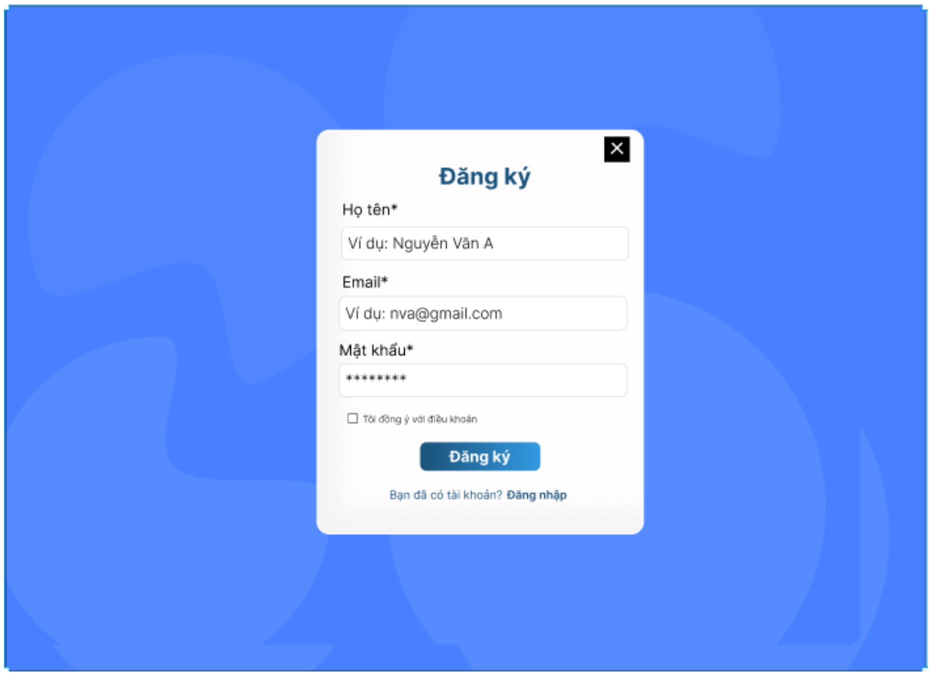
5 Prototype/Mockup

5.1. Figma Prototype Link

[Link Figma]()

5.2. Screenshots

- Register Page



Đăng ký

Họ tên*

Ví dụ: Nguyễn Văn A

Email*

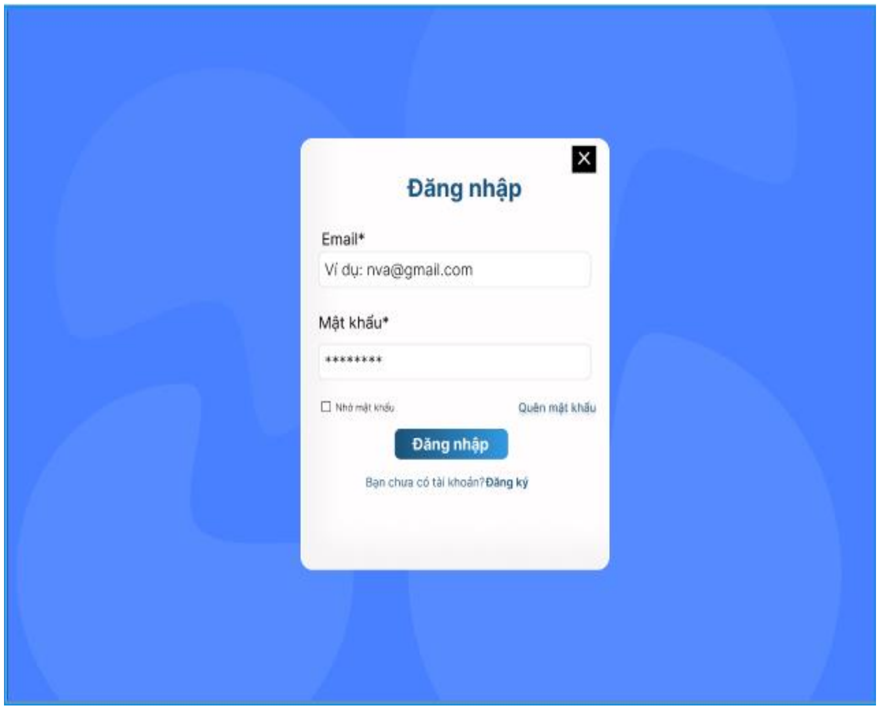
Ví dụ: nva@gmail.com

Mật khẩu*

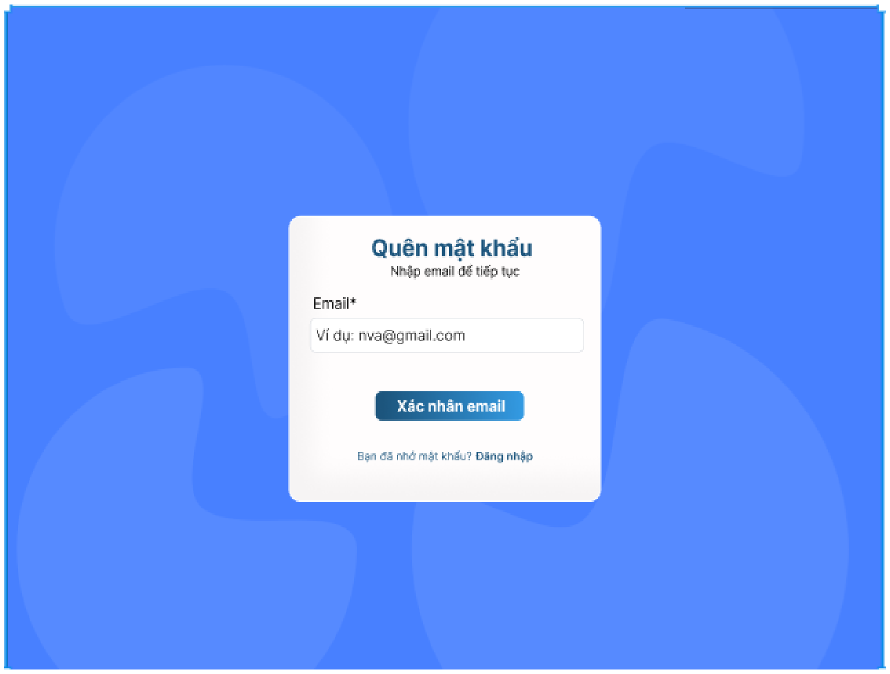
****

☐ Tôi đồng ý với điều khoản

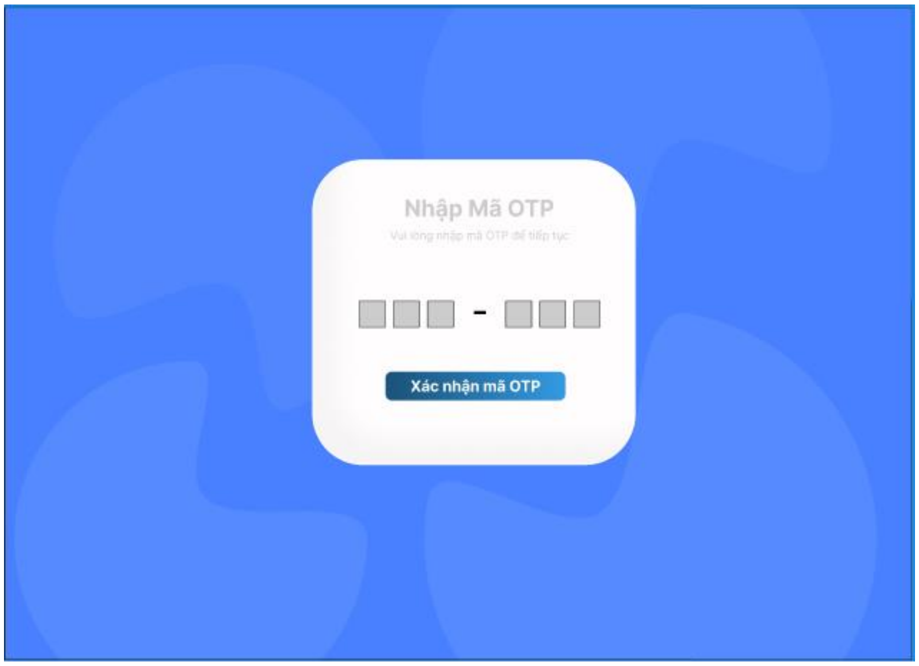
- Login page



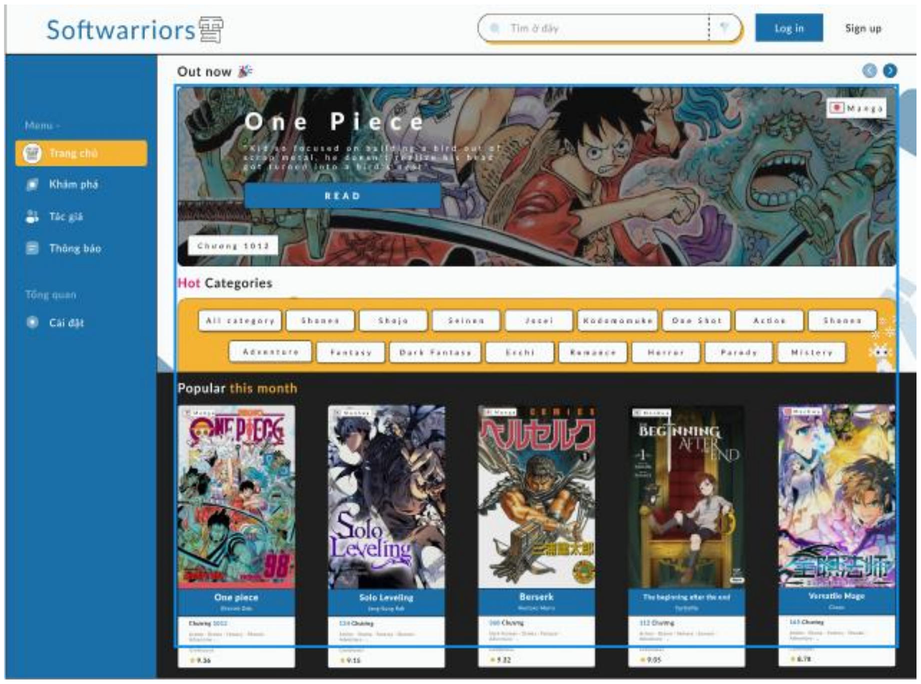
- Forgot-password page



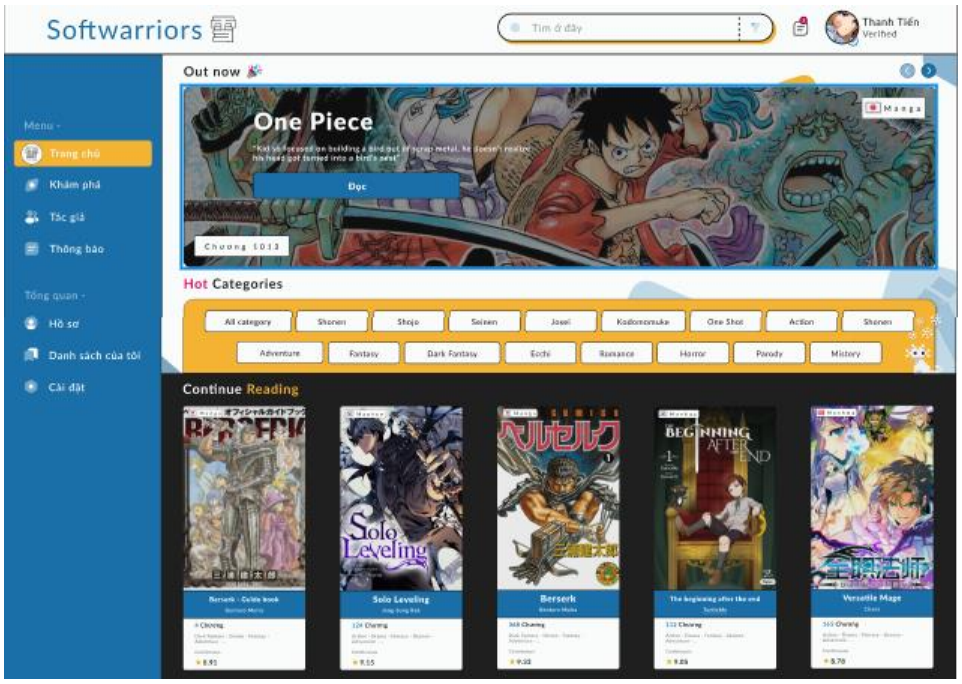
- OTP page



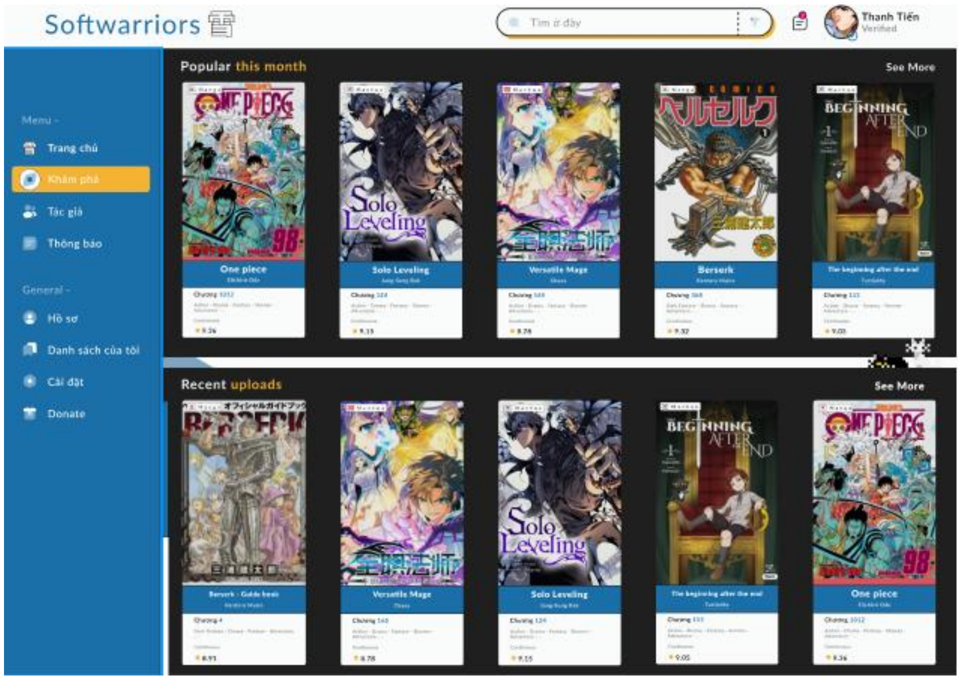
- Home page(when not logged in)

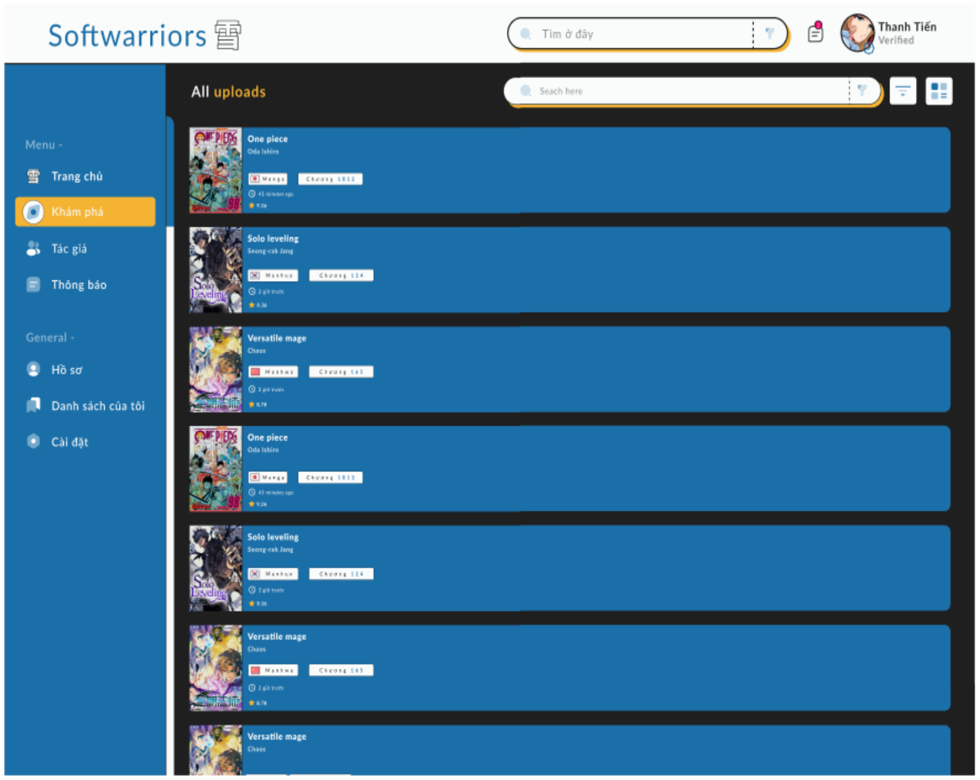


- Home page (when logged in)

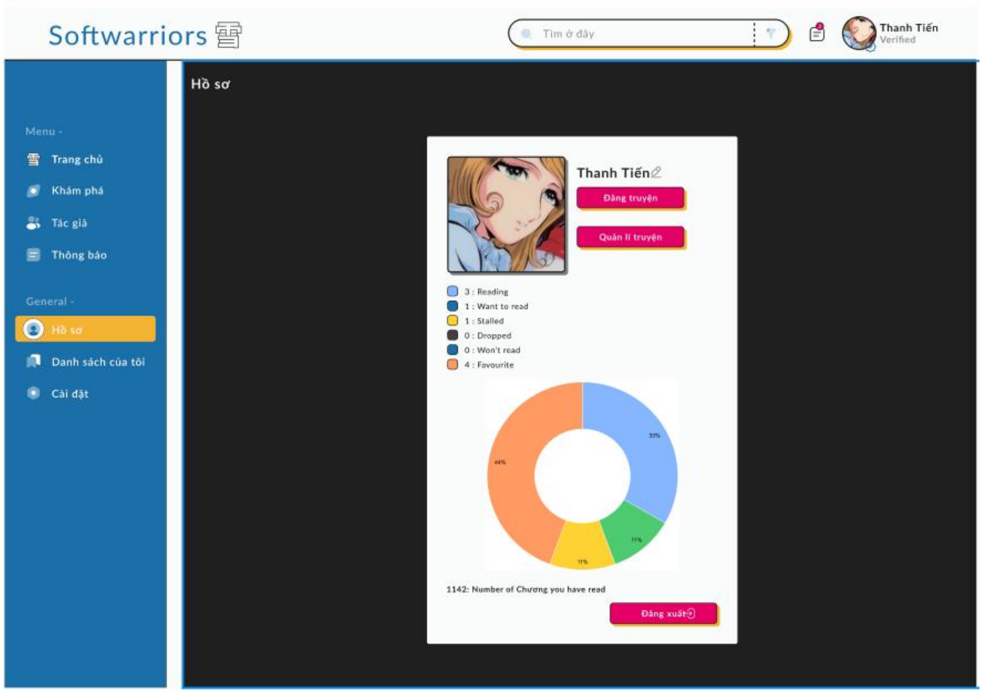


- Discover page

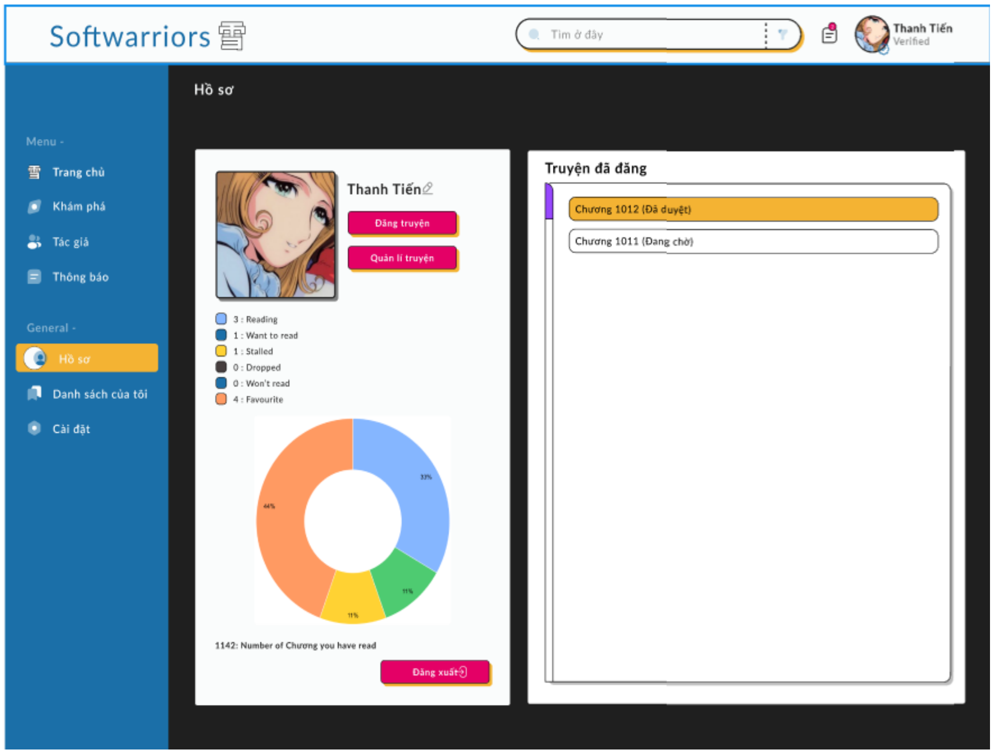




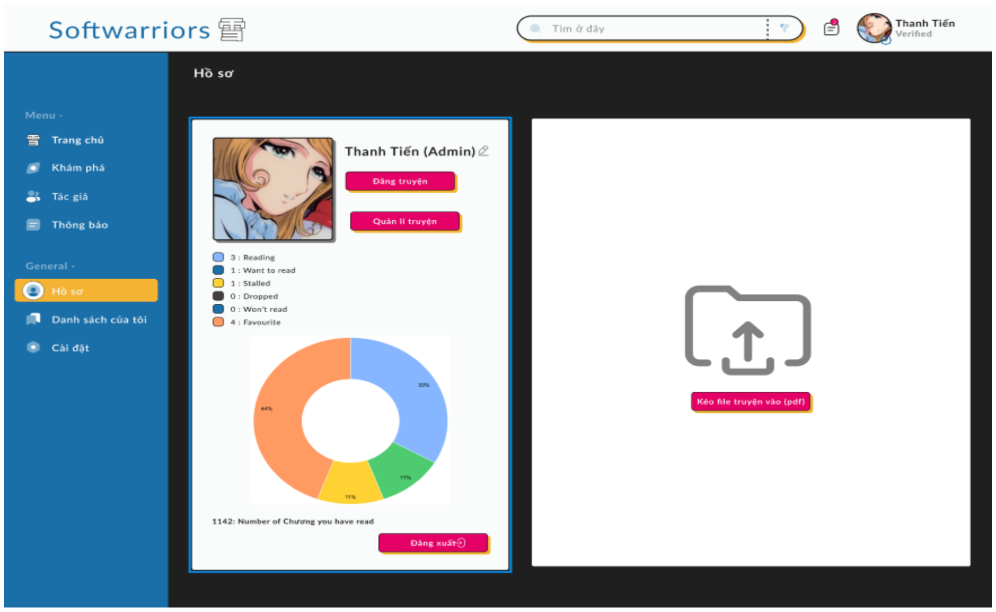
- Profile page



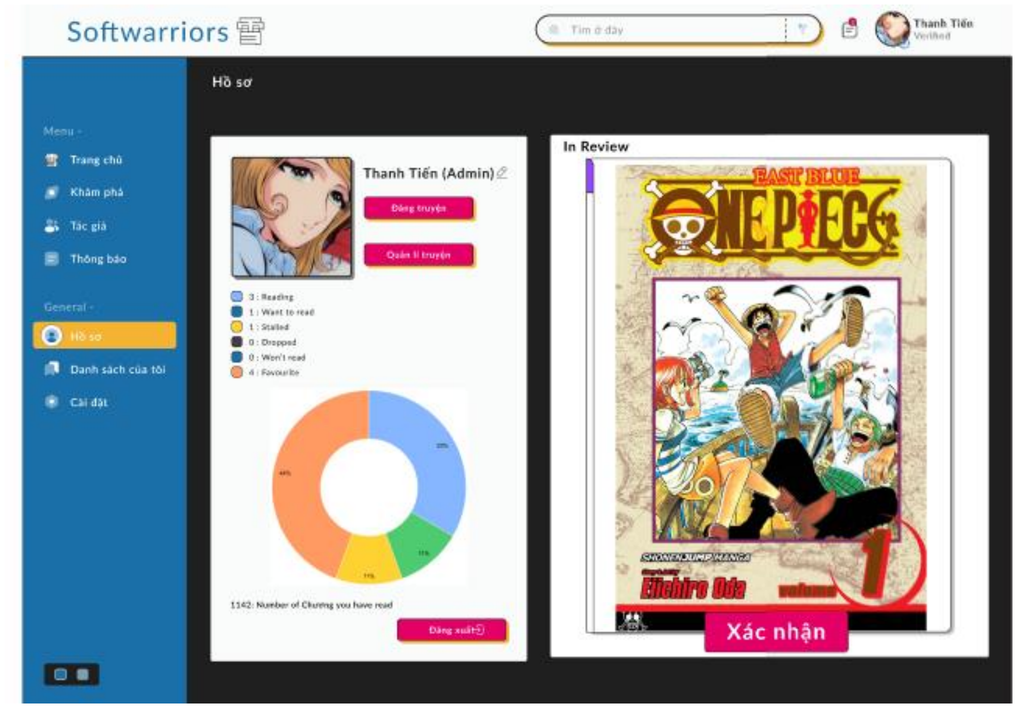
- Profile page (Posted manga)



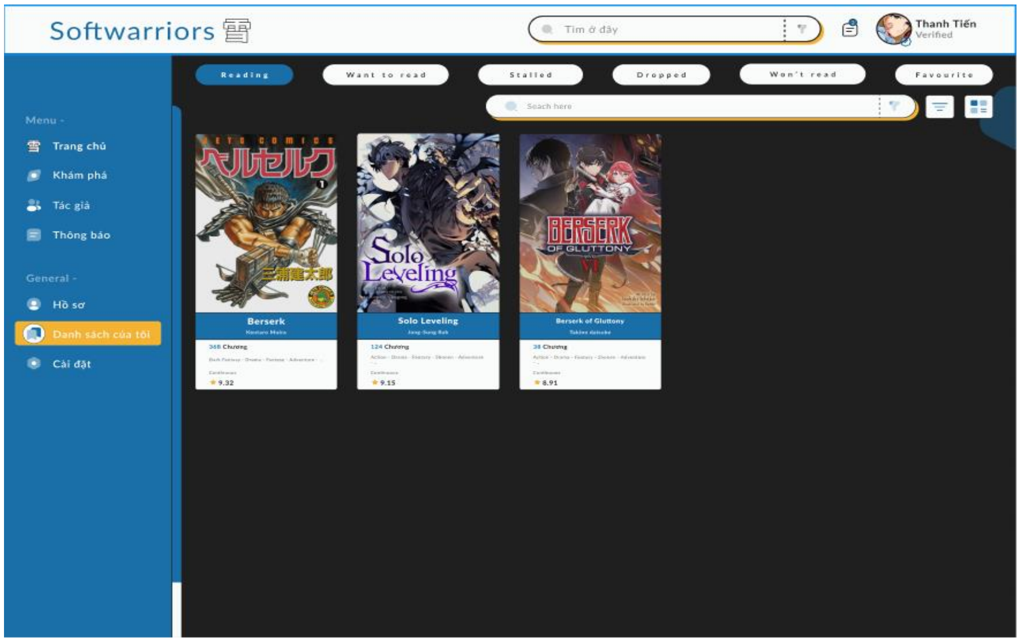
- Profile page (post manga)



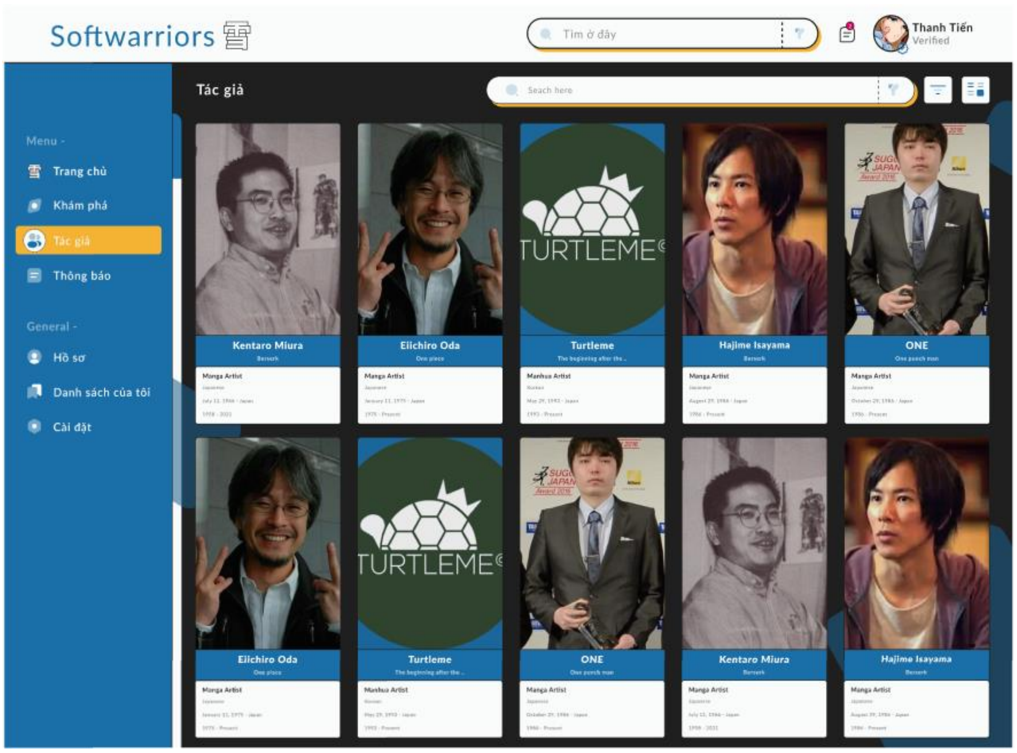
- Profile page (confirm story posting)



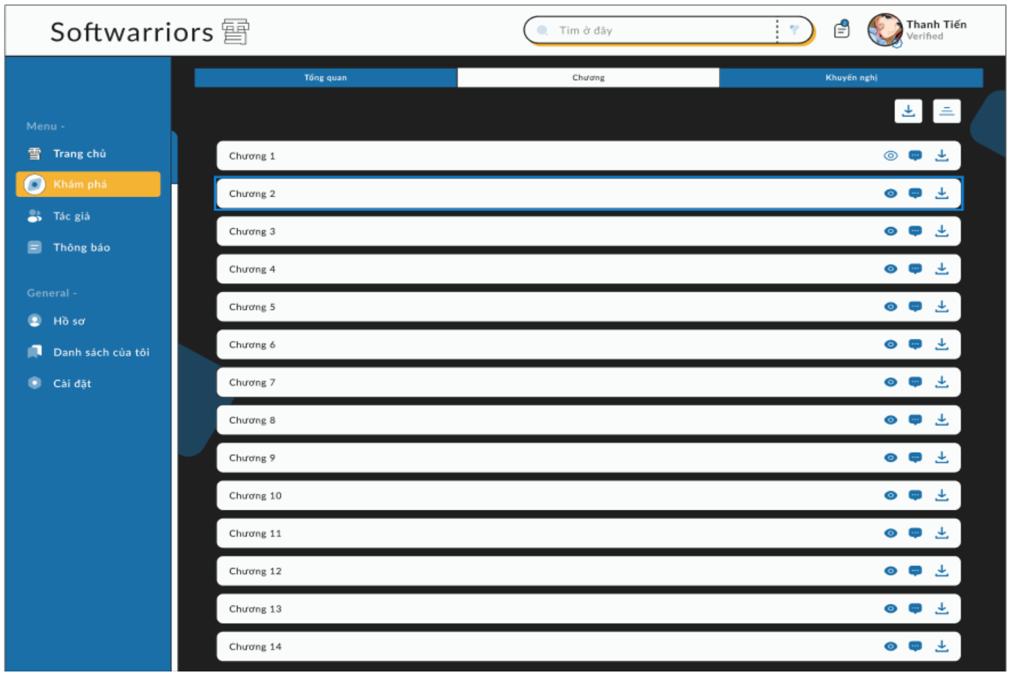
- My-list page



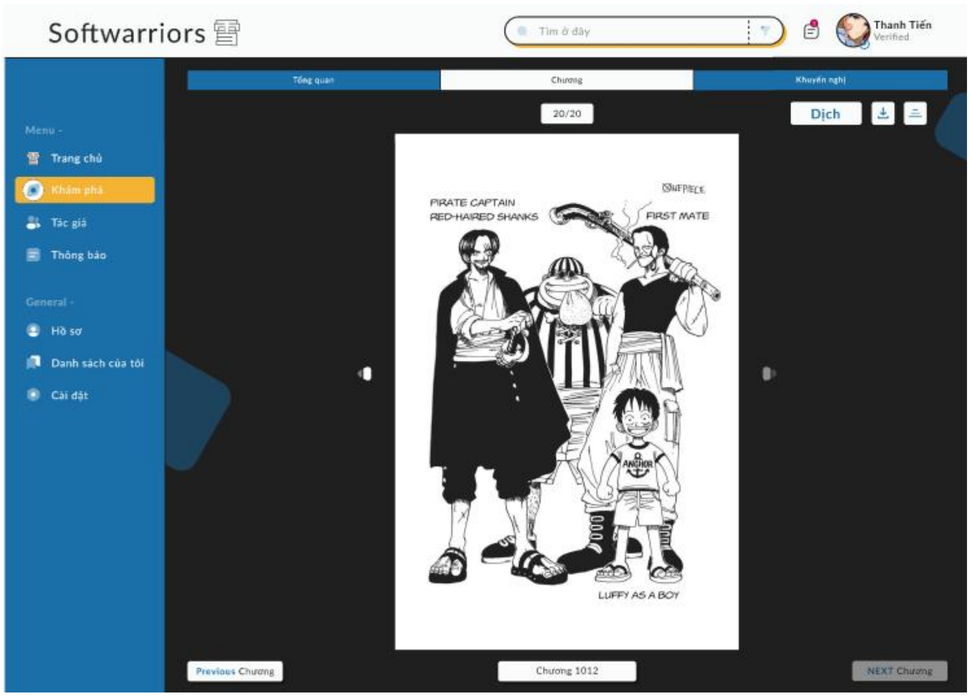
- Author page



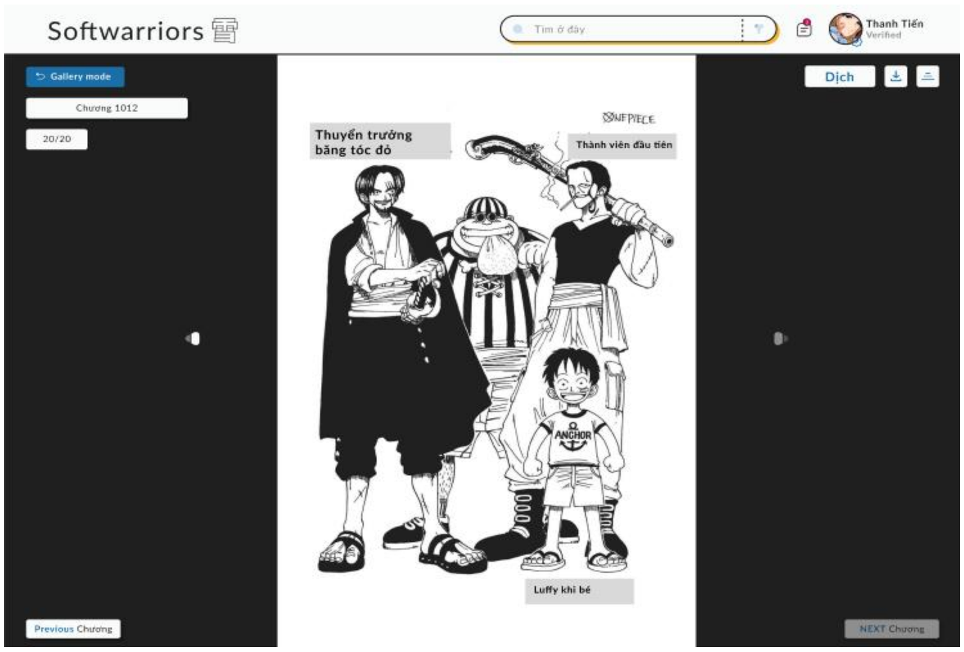
- Page by chapter in the manga



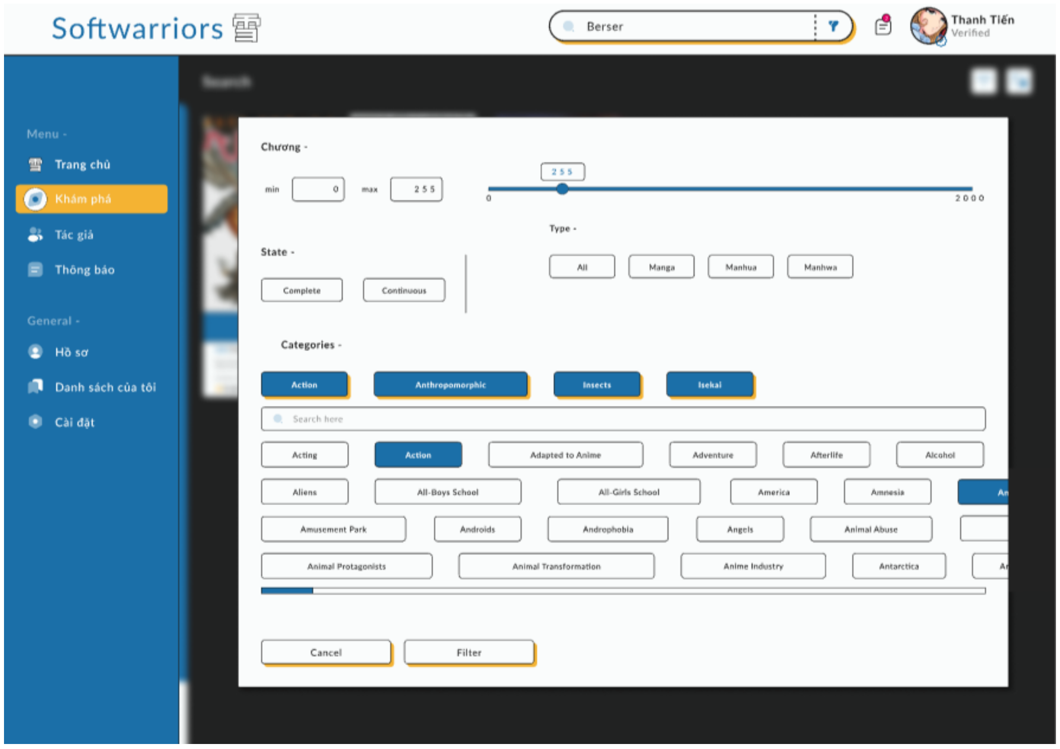
- Original manga page:

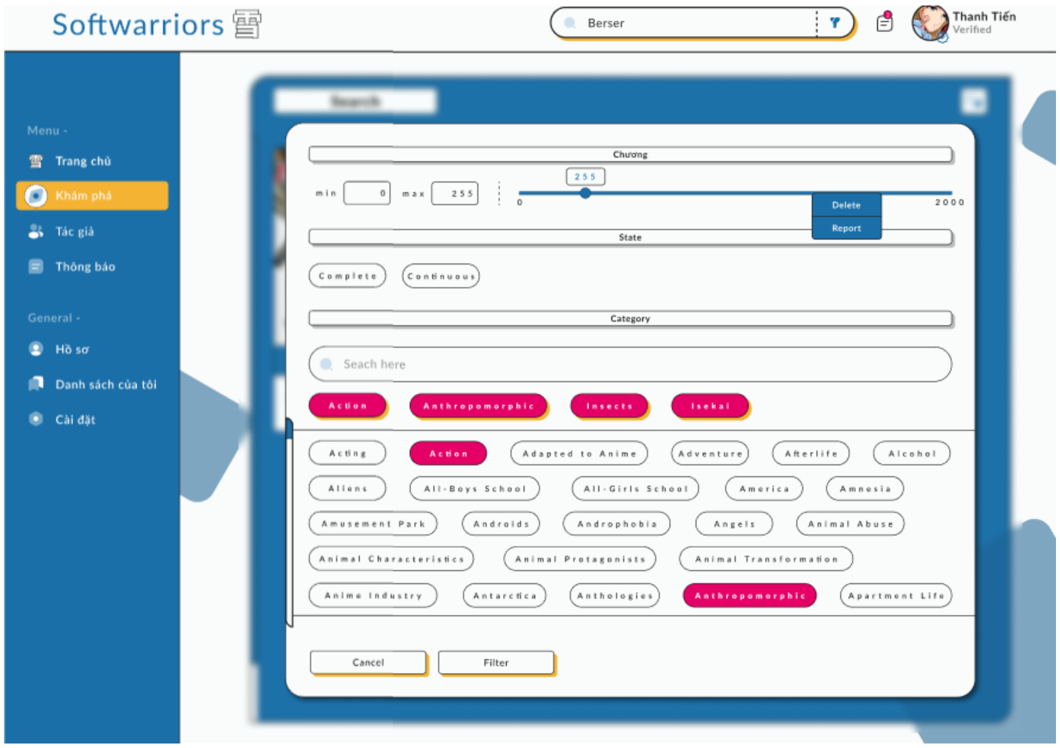


- Translated manga page:



- Filter page


Softwarriors
Berser
Thanh Tiến
Verified
Menu -
Trang chủ
Khám phá
Tác giả
Thông báo
General -
Hồ sơ
Danh sách của tôi
Cài đặt
Chương -
min 0
max 255
255
0
2000
Type -
All
Manga
Manhua
Manhwa
State -
Complete
Continuous
Categories -
Action
Anthropomorphic
Insects
Isekai
Search here
Acting
Action
Adapted to Anime
Adventure
Afterlife
Alcohol
Aliens
All-Boys School
All-Girls School
America
Amnesia
Amusement Park
Androids
Androphobia
Angels
Animal Abuse
Animal Protagonists
Animal Transformation
Anime Industry
Antarctica
Cancel
Filter


Softwarriors
Berser
Thanh Tiến
Verified
Menu -
Trang chủ
Khám phá
Tác giả
Thông báo
General -
Hồ sơ
Danh sách của tôi
Cài đặt
Chương
min 0
max 255
255
0
2000
Delete
Report
State
Complete
Continuous
Category
Seach here
Action
Anthropomorphic
Insects
Isekai
Acting
Action
Adapted to Anime
Adventure
Afterlife
Alcohol
Aliens
All-Boys School
All-Girls School
America
Amnesia
Amusement Park
Androids
Androphobia
Angels
Animal Abuse
Animal Characteristics
Animal Protagonists
Animal Transformation
Anime Industry
Antarctica
Anthologies
Anthropomorphic
Apartment Life
Cancel
Filter