Software Requirements Specification
Marito Multilingual Terminology PWA

Team Name: Velox

August 20, 2025

Contents

1. Introduction

1.1. Purpose

The purpose of this document is to define the software requirements of the Marito system, a multilingual terminology web application developed as part of the COS 301 Capstone project at the University of Pretoria. This document is intended to guide the development team, ensure alignment with client expectations, and serve as a reference for future maintenance, testing, and extension of the platform.

1.2. Scope

Marito is a multilingual, progressive web application (PWA) designed to provide users with access to curated statistical terminology in all 11 official South African languages. The system enables users to search and browse glossary terms, submit suggestions, and engage through a gamified contribution model. Marito supports offline functionality, glossary versioning, contributor uploads, and plugin-based feature extensions including AI-enhanced semantic search and analytics. The system is built using a microkernel-microservices architecture for flexibility, modularity, and future extensibility.

1.3. Intended Audience

This document is primarily intended for:

- **Developers and engineers** responsible for implementing and maintaining the system.
- **Project stakeholders and academic supervisors** overseeing the design, quality, and alignment of Marito with its educational goals.
- **Testers and QA personnel** who require a clear specification of expected system behavior.
- **Future contributors** who may extend the application or onboard new glossaries and features.

1.4. Document Overview

The remainder of this document includes detailed user stories, functional and non-functional requirements, architectural and design patterns, service contracts, constraints, and system models. It is structured to follow industry best practices and academic guidelines for software specification documents, ensuring clarity and completeness for all stakeholders.

1.5. Definitions, Acronyms, and Abbreviations

- **PWA**: Progressive Web Application
- **CI/CD**: Continuous Integration and Continuous Deployment

2. User Stories

This section outlines the high-level user goals and core system interactions that informed the functional requirements. Detailed use case specifications and diagrams can be found in the official project repository.

2.1. User Story #1: Sync Updates

ID: US001 (Marito Project)
Title: Automatic Synchronization of Downloaded Lexicon Data
As a: Marito application user (e.g., language enthusiast, NLP researcher, student) who has downloaded lexicon data for offline use,
I want: the application to automatically check for and apply updates to my downloaded lexicon data whenever I am online,
So that: I can be confident that I am always working with the most current and accurate version of the linguistic information without needing to manually re-download or check for updates.

Acceptance Criteria:

1. **Automatic Update Check:**
    - **Given** I have previously downloaded lexicon data,
    - **And** I open the Marito application with an active internet connection,
    - **Then** the application automatically initiates a check for updates to my downloaded data against the central repository in the background.
2. **Notification of Available Updates (Recommended):**
    - **Given** updates are available for my downloaded data,
    - **Then** the application clearly notifies me that updates are available (e.g., via a subtle in-app notification, a badge on a settings icon).
    - **And** the notification provides an option to view details about the updates (e.g., number of terms changed, lexicon version).
3. **Update Process:**
    - **Given** updates are available and I have an active internet connection,
    - **Then** the application allows me to initiate the download and application of these updates (or this happens automatically, depending on user settings or application design).
    - **And** the application provides clear feedback on the progress of the update (e.g., download progress bar, installation status).
    - **And** the update process is efficient and minimizes data usage (e.g., by only downloading changes/deltas if possible, rather than the entire dataset, for future enhancements).
4. **Successful Update:**

- **Given** the update process completes successfully,
- **Then** my locally stored lexicon data reflects the latest version from the central repository.
- **And** I receive a confirmation message that the data has been updated.

5. **Handling No Updates:**
   - **Given** I am online and no updates are available for my downloaded data,
   - **Then** the application does not interrupt my workflow with unnecessary notifications (or provides a subtle indication that data is "up-to-date").
6. **Offline State Post-Sync:**
   - **Given** my data has been successfully synced,
   - **Then** the newly updated data is fully accessible offline.
7. **Error Handling - Interrupted Connection:**
   - **Given** an update is in progress and my internet connection is lost,
   - **Then** the application gracefully pauses or stops the update process.
   - **And** I am notified of the interruption.
   - **And** the application attempts to resume the update when the connection is re-established, or allows me to manually retry.
   - **And** my previously downloaded (pre-update attempt) data remains intact and usable.
8. **Error Handling - Sync Conflict (Advanced Consideration for Future):**
   - **Given** a conflict occurs during synchronization (e.g., if local modifications were possible vs. server changes),
   - **Then** the system has a defined strategy for conflict resolution (e.g., prioritizes server version for this project's scope, notifies user if manual intervention were ever needed).
9. **User Control (Optional Enhancement):**
   - **Given** I am a user concerned about data usage or update timing,
   - **Then** I may have an option in settings to control sync behavior (e.g., sync only on Wi-Fi, schedule syncs, manual sync only).

Notes/Assumptions:

- This user story assumes that users primarily download data for offline reading and searching. Contributions and comments are made while online and sent to a central repository (as per Functional Requirement FR4.3).
- The complexity of only downloading changes/deltas (Acceptance Criterion 3.4) can be significant. For an initial implementation, syncing entire updated files might be a simpler starting point, with delta updates as a future enhancement.

- Conflict resolution (Acceptance Criterion 8) is likely simplified if the local data is treated as a read-only cache that gets overwritten by server updates, which aligns with the current understanding of the Marito project.

2.2. User Story #2: Download Language Resources for Offline Access

ID: US002 (Marito Project)
Title: Download Select Language Resources for Offline Use
As a: Marito application user (e.g., language enthusiast, NLP researcher, student),
I want: to be able to select and download specific language resources (like individual lexicons, glossaries, or dictionaries) to my device,
So that: I can access and use this information even when I do not have an active internet connection.

Acceptance Criteria:

1. **Discover and Select Resources for Download:**
    - **Given** I am browsing the available language resources within the application (while online),
    - **Then** I can clearly identify which resources are available for download.
    - **And** I can select one or more language resources to download.
    - **And** the application shows the estimated size of the selected resource(s) before initiating the download.
2. **Initiate Download:**
    - **Given** I have selected one or more language resources for download,
    - **Then** I can initiate the download process with a clear action (e.g., a "Download" button).
3. **Download Process Feedback:**
    - **Given** a download is in progress,
    - **Then** the application provides clear visual feedback on the download status (e.g., progress bar, percentage complete, estimated time remaining).
    - **And** I can continue to use other online features of the application while a download is in progress (background download).
    - **And** I have the option to pause and resume a download if needed.
    - **And** I have the option to cancel an ongoing download.
4. **Successful Download and Storage:**
    - **Given** a language resource download completes successfully,
    - **Then** the application notifies me that the download is complete.
    - **And** the downloaded resource is stored locally on my device in a way that the Marito application can access it offline.

- **And** the application clearly indicates which resources have been successfully downloaded and are available offline.

5. **Managing Downloaded Resources:**
   - **Given** I have downloaded language resources,
   - **Then** I can view a list of all my downloaded resources within the application.
   - **And** I can remove/delete downloaded resources from my device to free up storage space.
   - **And** the application shows the amount of local storage space currently used by downloaded Marito resources.
6. **Error Handling - Insufficient Storage:**
   - **Given** I attempt to download a resource,
   - **And** my device has insufficient storage space,
   - **Then** the application informs me of the insufficient storage and the download does not proceed (or pauses until space is freed).
7. **Error Handling - Interrupted Connection During Download:**
   - **Given** a download is in progress and my internet connection is lost,
   - **Then** the application gracefully pauses the download process.
   - **And** I am notified of the interruption.
   - **And** the application attempts to resume the download automatically when the connection is re-established, or allows me to manually resume.
8. **Error Handling - Download Failure:**
   - **Given** a download fails for reasons other than connection loss or insufficient storage (e.g., server error, corrupted file),
   - **Then** the application notifies me of the failure and provides a reason if possible.
   - **And** I have the option to retry the download.
9. **Accessing Downloaded Resources Offline:**
   - **Given** I have successfully downloaded a language resource,
   - **And** I am offline,
   - **Then** I can access and use that downloaded resource within the Marito application.

Notes/Assumptions:

- The user must be online to browse and initiate downloads.
- The application will need appropriate permissions to write to local device storage.
- The format of the downloaded data should be optimized for offline use and efficient storage.

- This user story focuses on the *download* functionality. A separate user story would cover the specifics of *accessing and using* the data offline (e.g., offline search within downloaded resources).

2.3. User Story #3: Search Feature

ID: US003 (Marito Project)
Title: Multilingual Term Search
As a: Marito application user (e.g., casual user, linguist, or contributor),
I want: to search across multiple multilingual glossaries and dictionaries using filters and smart suggestions,
So that: I can quickly find definitions, translations, and related entries in my preferred language.

Acceptance Criteria:

1. **Query Input:**
   - **Given** I am on the main search page,
   - **When** I enter a query into the search bar,
   - **Then** the system searches across all selected data sources and returns matching terms.
2. **Filter Options:**
   - **Given** I want to refine my search,
   - **Then** I can apply filters like language, part of speech, or glossary type before or after submitting the query.
3. **Fuzzy Search (Optional):**
   - **Given** my query has a typo or partial match,
   - **Then** the system offers similar results using fuzzy matching.
4. **AI-Powered Suggestions (Optional):**
   - **Given** I start typing in the search bar,
   - **Then** I see AI-generated autocomplete suggestions based on common terms or semantic matches.
5. **Search History:**
   - **Given** I have searched for terms in the past,
   - **Then** the system shows my recent queries and allows re-searching them.
6. **Sorting and Result Presentation:**
   - **Given** the results are displayed,
   - **Then** I can sort them by relevance, alphabetical order, or popularity,
   - **And** each result shows the term, language, and a brief definition snippet.

Assumptions:

- Offline support will allow previously downloaded glossaries to be searched.
- Filters and sort options persist across sessions.
- AI and fuzzy search are optional enhancements.
- Glossary datasets are already available and preloaded or fetched via sync.

2.4. User Story #4: Feedback Submissions

ID: US004 (Marito Project)
Title: User Contributions and Feedback
As a: logged-in user of the application,
I want: to be able to comment on terms and submit feedback or error reports,
So that: I can contribute to improving the accuracy and usefulness of the data content.

Acceptance Criteria:

1. **Commenting on Terms:**
   - **Given** I am logged into my account,
   - **And** I am viewing a term or entry,
   - **Then** I should see a comment section below the entry,
   - **And** I should be able to submit my own comment.
2. **Submitting Feedback:**
   - **Given** I am logged into my account,
   - **And** I am viewing a specific term or entry,
   - **And** I choose to provide feedback,
   - **Then** I should be presented with a feedback form,
   - **And** I should receive a confirmation message after successfully submitting the feedback.
3. **Voting on User Contributions:**
   - **Given** I am logged into my account,
   - **And** I am viewing a comment or suggestion submitted by another user,
   - **Then** I should see upvote and downvote buttons associated with it,
   - **And** I should be able to cast one vote per contribution,
   - **And** the vote count should update immediately after I vote.
4. **Approval Status of Feedback:**
   - **Given** I have submitted feedback for a term,
   - **And** the feedback has been approved by a moderator,

- **Then** the approved feedback should be integrated into the application content,
- **And** I should receive a notification confirming the integration.

5. **Marking Approved Submissions:**
   - **Given** user feedback or content has been approved and integrated,
   - **Then** the associated entry should be visibly marked as a "User Submission" to distinguish it from original content.
6. **Rewarding User Contributions (Optional Gamification):**
   - **Given** I am logged in and submit a comment, feedback, or report,
   - **When** my contribution meets a predefined threshold (e.g., approved, upvoted),
   - **Then** I should earn points or badges for that action,
   - **And** my contribution stats should be viewable in my profile.

Notes/Assumptions:

- Only logged-in users can comment, submit feedback, vote, or receive contribution rewards.
- Users must have verified accounts to interact with community features (e.g., comments, voting).
- Rate limiting will be implemented to prevent spam submissions.

2.5. User Story #5: Gamification Feature

ID: US005 (Marito Project)
Title: Contribution-Based Rewards and Progress Tracking
As a: frequent Marito contributor (e.g., user who submits suggestions or comments),
I want: to earn points, unlock badges, and track my contribution progress,
So that: I feel motivated to participate and can see recognition for my efforts.

Acceptance Criteria:

1. **Points for Contribution:**
   - **Given** I submit a suggestion, comment, or report an issue on a term,
   - **Then** I earn points based on the type and quality of the contribution.
2. **Achievement Unlocking:**
   - **Given** I reach a milestone,
   - **Then** I unlock a badge or achievement.
3. **Progress Dashboard:**
   - **Given** I navigate to my profile,
   - **Then** I can view my total points, badges earned, and contribution rank.

4. **Real-Time Feedback:**
    - **Given** I perform a gamified action,
    - **Then** I receive instant feedback such as a pop-up message.
5. **Leveling Up:**
    - **Given** I reach predefined thresholds,
    - **Then** my user rank or title updates accordingly.
6. **Offline Support:**
    - **Given** I contribute while offline,
    - **Then** my contributions and points are synced once I reconnect.

2.6. User Story #6: UpVote System

ID: US006 (Marito Project)
Title: Crowdsourced Validation via UpVoting
As a: Marito application user (e.g., casual user, linguist, or academic) who wants to contribute to the accuracy and quality of lexicon entries,
I want: to upvote (or downvote) suggested changes or comments on terms in the lexicon,
So that: the community can collectively validate contributions, and moderators can prioritize high-quality updates for integration into the central repository.

Acceptance Criteria:

1. **Voting Interface:**
    - **Given** I am viewing a term entry with user-submitted comments or suggested changes,
    - **Then** I see an option to upvote or downvote each contribution.
    - **And** the current vote count is displayed next to each contribution.
2. **Vote Submission:**
    - **Given** I am logged in and have not yet voted on a specific contribution,
    - **When** I click the upvote/downvote button,
    - **Then** my vote is recorded immediately (if online) or queued for sync (if offline).
    - **And** the UI reflects my vote and updates the vote count.
3. **Prevent Duplicate Voting:**
    - **Given** I have already voted on a contribution,
    - **Then** the UI prevents me from voting again (unless I undo my vote).
4. **Offline Handling:**
    - **Given** I vote while offline,

- **Then** the vote is stored locally and synced to the central repository when I reconnect.

5. **Moderation Visibility (Optional Enhancement):**
   - **Given** I am a moderator,
   - **Then** I can filter contributions by vote count to prioritize high-quality submissions.
6. **Feedback Transparency:**
   - **Given** a contribution receives significant downvotes,
   - **Then** the system may flag it for review (future enhancement).

Notes/Assumptions:

- Voting requires user authentication to prevent abuse.
- Vote counts are public to encourage transparency.
- Offline votes are treated as "pending" until synced.
- Future enhancements could include:
  - Weighted voting for trusted users (e.g., linguists).

2.7. User Story #7.1: Word Frequency Trends

ID: US007.1 (Marito Project)
Title: Historical Word Usage Visualization
As a: Linguist or language researcher studying lexical evolution,
I want: To view historical trends of word usage frequency across different time periods,
So that: I can identify patterns in language adoption, obsolescence, or cultural influences.

Acceptance Criteria:

1. **Trend Visualization:**
   - **Given** I select a word or phrase,
   - **Then** I see a line chart showing its monthly/quarterly/yearly frequency.
   - **And** can toggle between absolute counts and percentage changes.
2. **Comparative Analysis:**
   - **Given** I select multiple words,
   - **Then** the system overlays their trends with distinct colors.
   - **And** provides a legend identifying each word.
3. **Contextual Data:**

- **Given** I hover over a data point,
- **Then** I see exact usage counts and sample sentences.
- **And** can click to view source documents.

4. **Export Functionality:**
   - **Given** I want to analyze data externally,
   - **Then** I can export charts as PNG or data as CSV/JSON.

Notes/Assumptions:

- Data aggregates nightly for performance.
- Supports all 12 official South African languages.
- Default view shows last 12 months.
- Future enhancements could include:
  - Regional usage heatmaps.
  - Sociolinguistic correlation analysis.

2.8. User Story #7.2: Contribution Analytics

ID: US007.2 (Marito Project)
Title: User Contribution Visualization
As a: Regular contributor to the Marito platform,
I want: To see a breakdown of my edits and comments across languages,
So that: I can track my impact and focus on underrepresented languages.

Acceptance Criteria:

1. **Personal Dashboard:**
   - **Given** I view my profile,
   - **Then** I see a pie chart of my contributions by language.
   - **And** a timeline of my activity.
2. **Progress Metrics:**
   - **Given** I'm an active user,
   - **Then** I see my percentile ranking in the community.
   - **And** suggested languages needing more contributions.

Notes/Assumptions:

- Updates within 5 minutes of contributions.
- Includes all contribution types.
- Future enhancements could include:
  - Team contribution tracking.
  - Contribution quality scoring.

2.9. User Story #7.3: Trending Terms

ID: US007.3 (Marito Project)
Title: Real-Time Popularity Tracking
As a: Language learner or cultural researcher,
I want: To see which words are currently trending in popularity,
So that: I can stay current with evolving language usage.

Acceptance Criteria:

1. **Trending Display:**
    - **Given** I visit the homepage,
    - **Then** I see a "Trending Now" carousel with top terms.
    - **And** percentage change indicators.
2. **Contextual Information:**
    - **Given** I select a trending word,
    - **Then** I see related news/events driving popularity.
    - **And** its historical frequency graph.

Notes/Assumptions:

- Updates every 4 hours.
- Excludes spam/fake trends.
- Future enhancements could include:
    - User-submitted trend explanations.
    - Regional trend variations.

Assumptions:

- Contributions are validated before points are awarded to prevent abuse.
- Gamification rewards are symbolic, not monetary.
- This feature is designed to encourage consistent, high-quality engagement.

2.10. User Story #8: Responsive Design

ID: US008 (Marito Project)
Title: Responsive Layout for Mobile and Desktop Devices
As a: user who may need to access the Marito application while on the go,
I want: the user interface to adapt and remain fully functional on mobile phones, tablets, and desktops,
So that: I can quickly and seamlessly use the application in any situation, regardless of the device I'm using.

Acceptance Criteria:

1. **Mobile Adaptability:**
    - **Given** I am using the Marito application,
    - **And** I am on a mobile device,
    - **Then** the layout should automatically adjust to fit the screen without horizontal scrolling.
2. **Touch-Friendly UI:**
    - **Given** I am interacting with the app on a touchscreen device,
    - **And** I tap on any button or link,
    - **Then** the tapped element should be responsive to touch and have a minimum tap area of 48px $\times$ 48px.
3. **Touchscreen Swiping and Scrolling Support:**
    - **Given** I am using the Marito app on a touchscreen device,
    - **And** I am viewing content that extends beyond the initial screen (e.g., a list or feed),
    - **Then** I should be able to scroll vertically or horizontally (depending on the content) using swipe gestures,
    - **And** the scrolling should be smooth and responsive.
4. **Adaptive Mobile Navigation Elements:**
    - **Given** I am using the Marito app on a mobile device,
    - **Then** the navigation should adapt to a mobile-friendly layout, such as displaying a hamburger menu instead of a full-width navigation bar.
5. **Device Orientation Handling:**
    - **Given** I am using the app on a mobile device,
    - **And** I switch between portrait and landscape mode,
    - **Then** the UI layout should follow without breaking or cutting off content.
6. **Mobile Hardware and OS Compatibility:**
    - **Given** I am using the Marito app on a mobile device running a supported OS (e.g., Android 10+, iOS 13+),
    - **And** the device has at least 2 GB of RAM and a modern mobile browser (e.g., Chrome),
    - **Then** all core features should work smoothly without crashes or performance lags.
7. **Error Handling - Network Errors (Mobile Use):**
    - **Given** I lose internet connectivity due to a poor signal while using the app,

- **Then** the application should seamlessly switch to its offline version without interruption,
- **And** provide an indication informing me that the app is currently in offline mode.

Notes/Assumptions:

- Devices with screen widths less than or equal to 768 pixels are considered mobile, and touch interaction is expected on these devices.

2.11. User Story #9: Dictionary Glossary

ID: US009 (Marito Project)
Title: Glossaries User Stories
As a: terminology researcher
I want: to view detailed information about a specific term,
So that: I can understand its meaning and translations.

Acceptance Criteria:

1. **Definition Display:**
   - **Given** I have selected a term from search results,
   - **And** I should see its complete definition,
   - **Then** its assigned category with clickable link to the full glossary.
2. **Translation Panel**
   - **Given** I am viewing a term,
   - **And** all available translations should be displayed in a structured table,
   - **Then** clear language code labels (e.g., "afr", "zul").
3. **Missing Data Handling:**
   - **Given** a term lacks translation for certain languages,
   - **Then** those fields should display "Translation not available",
   - **And** be visually distinct from complete entries.

Notes/Assumptions:

- Integrates with US003 search results.
- Preserves all existing search filters when navigating from results to term view

2.12. User Story #10: Glossary Category Navigation

ID: US010 (Marito Project)
Title: Glossary Category Navigation
As a: user interested in exploring specific domains (e.g., Agriculture, Legal),
I want: to browse all terms within a particular category,
So that: I can discover related terminology.

Acceptance Criteria:

1. **Category Selection:**
   - **Given** I access the glossary browser,
   - **Then** I should see all available categories.
2. **Term Listing:**
   - **Given** I select a category (e.g., "Agriculture"),
   - **Then** I should see paginated results of all terms,
   - **With** options to filter by language availability.
3. **Cross-Referencing:**
   - **Given** I view a term in a glossary,
   - **Then** I should see related terms from the same category,
   - **And** options to navigate to similar categories.

Notes/Assumptions:

- Categories are predefined based on dataset taxonomy
- Works both online and for downloaded glossaries

2.13. User Story #11: Term Bank Translations

ID: US011 (Marito Project)
Title: Access Multilingual Translations
As a: multilingual user,
I want: to toggle between translations for a term,
So that: I can understand it in my preferred language.

Acceptance Criteria:

1. **Translation Toggle:**
   - **Given** I view a term,
   - **When** I click a language tab (e.g., "Zulu"),
   - **Then** I should see the translation ("Imbewu kawoyela").
2. **Missing Translations:**

- **Given** a term has no translation for a language,
- **Then** display "Translation not available.

Notes/Assumptions:

- Uses the translations field from the dataset.

2.14. User Story #12: Submit Feedback

ID: US012 (FeedbackHub Project)
Title: Multi-Category Feedback Submission
As a: Application user (customer, stakeholder, or end-user) who wants to communicate with the organization about their experience,
I want: to submit feedback through different categories (suggestions, complaints, or compliments) with optional contact information,
So that: the organization can understand my needs, address issues, and improve their services based on user input.

Acceptance Criteria:

1. **Category Selection:**
   - **Given** I am on the feedback submission page,
   - **Then** I can choose between three feedback types: Suggestion, Complaint, or Compliment.
   - **And** each category has distinct visual indicators (icons and colors).
2. **Form Completion:**
   - **Given** I have selected a feedback category,
   - **When** I fill out the feedback form,
   - **Then** I can optionally provide my name and email address.
   - **And** I must provide a message describing my feedback (required field).
3. **Contextual Guidance:**
   - **Given** I select a specific feedback type,
   - **Then** the form updates with appropriate placeholder text and messaging.
   - **And** the submit button reflects the selected category.
4. **Form Validation:**
   - **Given** I attempt to submit feedback,
   - **When** the required message field is empty,
   - **Then** the submit button remains disabled.
   - **And** form validation prevents submission until requirements are met.
5. **Submission Confirmation:**

- **Given** I successfully submit feedback,
- **Then** I receive immediate visual confirmation of submission.
- **And** the form resets after a brief thank you message.

6. **Anonymous Submissions:**
    - **Given** I choose not to provide contact information,
    - **Then** my feedback is still accepted and processed.
    - **And** my submission is marked as "Anonymous" in the system.

Notes/Assumptions:

- All feedback types use the same form structure for consistency.
- Contact information is optional to encourage participation.
- System logs all submissions with timestamps.
- Future enhancements could include file attachments.

2.15. User Story #13: Admin Dashboard Management

ID: US013 (FeedbackHub Project)
Title: Comprehensive Feedback Administration
As a: System administrator who needs to monitor and respond to user feedback,
I want: to view, filter, search, and manage all submitted feedback through a centralized dashboard,
So that: I can efficiently track feedback trends, prioritize responses, and ensure all user concerns are addressed appropriately.

Acceptance Criteria:

1. **Dashboard Overview:**
    - **Given** I access the admin dashboard,
    - **Then** I see summary statistics for total feedback, pending items, in-progress items, and resolved items.
    - **And** statistics include trend indicators showing changes over time.
2. **Feedback Listing:**
    - **Given** I am viewing the feedback list,
    - **Then** each item displays type, status, priority, user information, and submission date.
    - **And** items are color-coded by category and status for quick visual identification.
3. **Filtering Capabilities:**
    - **Given** I want to focus on specific feedback,

- **When** I use the filter controls,
- **Then** I can filter by feedback type (all, suggestions, complaints, compliments).
- **And** I can filter by status (all, pending, in-progress, resolved).

4. **Search Functionality:**
   - **Given** I need to find specific feedback,
   - **When** I use the search bar,
   - **Then** the system searches across feedback content, user names, and email addresses.
   - **And** results update in real-time as I type.
5. **Detail Management:**
   - **Given** I click on a feedback item,
   - **Then** I see full details in a dedicated panel.
   - **And** I can update the status using a dropdown menu.
   - **And** changes are immediately reflected in the system.
6. **Status Workflow:**
   - **Given** I am managing feedback items,
   - **Then** I can move items through the workflow: Pending → In Progress → Resolved.
   - **And** status changes are tracked with timestamps.

Notes/Assumptions:

- Admin access requires authentication and appropriate permissions.
- All actions are logged for audit purposes.
- System supports real-time updates when multiple admins are working.
- Future enhancements could include automated assignment.

2.16. User Story #14: Profile Management

ID: US014 (Marito)
Title: User Profile Management
As a: logged-in user of the application,
I want: to be able to manage my profile information including picture, username, email address, and password,
So that: I can maintain accurate personal information and secure access to my account.

Acceptance Criteria:

1. **Setting Profile Picture:**

- **Given** I am logged into my account,
- **And** I am viewing my profile settings,
- **Then** I should see an option to upload or change my profile picture,
- **And** I should be able to select an image file and save it successfully.

2. **Changing Username:**
    - **Given** I am logged into my account,
    - **And** I am viewing my profile settings,
    - **Then** I should see a field to update my username,
    - **And** I should receive confirmation when the username is successfully changed.
3. **Changing Email Address:**
    - **Given** I am logged into my account,
    - **And** I want to update my email address,
    - **Then** I should be required to validate my existing password before making the change,
    - **And** I should receive a confirmation email at the new address to verify the change.
4. **Changing Password:**
    - **Given** I am logged into my account,
    - **And** I want to change my password,
    - **Then** I should be required to validate my existing password,
    - **And** I should be able to enter and confirm a new password,
    - **And** I should receive confirmation when the password is successfully updated.
5. **Logging Out:**
    - **Given** I am logged into my account,
    - **When** I choose to log out,
    - **Then** I should be securely signed out of the application,
    - **And** I should be redirected to the login page or home page.

Notes/Assumptions:

- Users must be authenticated to access profile management features.
- Password changes require validation of the current password for security.
- Profile picture uploads should have file size and format restrictions.

2.17. User Story #15: View Saved Languages

ID: US015 (Marito)
Title: Language Collection Management
As a: User who wants to track my language learning progress,
I want: to view all languages I have saved in my workspace,
So that: I can easily access and manage my language learning collection and see my overall language portfolio.

Acceptance Criteria:

1. **Language Display:**
   - **Given** I am logged into the workspace system,
   - **When** I navigate to the saved languages section,
   - **Then** I see a list of all languages I have previously saved.
   - **And** each language entry shows relevant details like name and date saved.
2. **Collection Overview:**
   - **Given** I am viewing my saved languages,
   - **Then** I can see the total count of languages in my collection.
   - **And** languages are organized in a clear, readable format.
3. **Access Integration:**
   - **Given** I have saved languages in my collection,
   - **Then** I can easily navigate to related saved terms and glossary items.
   - **And** the system provides seamless navigation between language components.

Notes/Assumptions:

- Languages are persistently stored in user's workspace.
- System maintains relationship between languages and their associated content.
- Future enhancements could include language categorization and sorting options.

2.18. User Story #16: View Saved Terms

ID: US016 (Marito)
Title: Terminology Management and Review
As a: User building vocabulary in multiple languages,
I want: to view and manage all terms I have saved across different languages,
So that: I can review my vocabulary progress and access saved terminology for study and reference.

Acceptance Criteria:

1. **Term Collection Display:**

- **Given** I access the saved terms section,
- **Then** I see all terms I have previously saved.
- **And** each term displays relevant information like definition, language, and date saved.

2. **Language Association:**
   - **Given** I am viewing saved terms,
   - **Then** I can see which language each term belongs to.
   - **And** I can filter or group terms by their associated languages.
3. **Progress Tracking:**
   - **Given** I have been saving terms over time,
   - **Then** I can track my vocabulary building progress.
   - **And** I can see submission progress for terms I'm working on.
4. **Search and Filter:**
   - **Given** I have multiple saved terms,
   - **When** I use the search functionality,
   - **Then** I can quickly locate specific terms or filter by criteria.

Notes/Assumptions:

- Terms are linked to their source languages.
- System tracks submission progress for learning workflows.
- Integration with external systems for enhanced term management.

2.19. User Story #17: View Saved Glossary

ID: US017 (Marito)
Title: Glossary Access and Management
As a: User who needs quick access to specialized terminology and definitions,
I want: to view and manage my saved glossary entries,
So that: I can maintain a personal reference collection and quickly look up important terms and concepts.

Acceptance Criteria:

1. **Glossary Navigation:**
   - **Given** I access the saved glossary section,
   - **Then** I see all glossary entries I have saved.
   - **And** entries are organized in a logical, searchable format.
2. **Content Display:**

- **Given** I am viewing the glossary,
- **Then** each entry shows the term, definition, and relevant context.
- **And** I can easily read and reference the content.

3. **Cross-Reference Integration:**
   - **Given** I am using the glossary,
   - **Then** I can navigate to related saved languages and terms.
   - **And** the system maintains connections between glossary entries and other workspace items.

Notes/Assumptions:

- Glossary entries can be linked to multiple languages.
- System supports both user-created and imported glossary content.
- Integration with search functionality across all saved items.

2.20. User Story #18: Track Submitted Term Progress

ID: US018 (Marito)
Title: Term Submission Progress Monitoring
As a: User who submits terms for review or processing,
I want: to track the progress of my submitted terms through the system workflow,
So that: I can monitor the status of my contributions and know when they have been processed or approved.

Acceptance Criteria:

1. **Progress Visualization:**
   - **Given** I have submitted terms for processing,
   - **When** I access the progress tracking section,
   - **Then** I see the current status of each submitted term.
   - **And** progress is clearly indicated through status indicators or progress bars.
2. **Status Updates:**
   - **Given** my terms are being processed,
   - **Then** I receive updates when status changes occur.
   - **And** I can see timestamp information for each status change.
3. **Submission History:**
   - **Given** I have submitted multiple terms over time,
   - **Then** I can view a complete history of my submissions.
   - **And** I can filter by status, date, or other relevant criteria.

4. **Feedback Integration:**
    - **Given** my submitted terms require revisions,
    - **Then** I can see any feedback or comments from reviewers.
    - **And** I can resubmit updated versions when needed.

Notes/Assumptions:

- System maintains workflow states for submitted content.
- External review processes may impact term progression.
- Notifications may be implemented for status changes.

2.21. User Story #19: Organize Saved Items into Groups

ID: US019 (Marito)
Title: Workspace Organization and Categorization
As a: User with multiple saved languages, terms, and glossary entries,
I want: to organize my saved items into custom groups and categories,
So that: I can efficiently manage my workspace content and quickly find related items for specific projects or learning goals.

Acceptance Criteria:

1. **Group Creation:**
    - **Given** I want to organize my saved items,
    - **When** I access the organization tools,
    - **Then** I can create custom groups with descriptive names.
    - **And** I can assign colors or icons to distinguish different groups.
2. **Item Assignment:**
    - **Given** I have created groups,
    - **When** I view my saved languages, terms, or glossary entries,
    - **Then** I can assign items to one or more groups.
    - **And** I can move items between groups as needed.
3. **Group Management:**
    - **Given** I have organized items into groups,
    - **Then** I can view items filtered by group membership.
    - **And** I can edit group properties like names and descriptions.
    - **And** I can delete groups while preserving the underlying saved items.
4. **Cross-Category Organization:**
    - **Given** I want comprehensive organization,

- **Then** I can create groups that span multiple item types.
- **And** I can organize languages, terms, and glossary entries together based on themes or projects.

Notes/Assumptions:

- Groups are flexible containers that don't restrict item functionality.
- Items can belong to multiple groups simultaneously.
- Organization preferences are saved to user's workspace.
- System maintains referential integrity when groups are modified.

2.22. User Story #20: Search and Filter Saved Items

ID: US020 (Marito)
Title: Advanced Search and Filtering Capabilities
As a: User with extensive saved content in my workspace,
I want: to search and filter across all my saved items using various criteria,
So that: I can quickly locate specific content regardless of volume and efficiently work with targeted subsets of my saved materials.

Acceptance Criteria:

1. **Universal Search:**
   - **Given** I want to find specific content,
   - **When** I use the search functionality,
   - **Then** the system searches across all saved languages, terms, and glossary entries.
   - **And** search results highlight matching content and provide context.
2. **Advanced Filtering:**
   - **Given** I need to narrow down my saved items,
   - **When** I apply filters,
   - **Then** I can filter by item type, language, date saved, or custom groups.
   - **And** multiple filters can be combined for precise results.
3. **Search Performance:**
   - **Given** I perform searches on my saved content,
   - **Then** results appear quickly without significant delay.
   - **And** the search function works efficiently even with large collections.
4. **Filter Persistence:**
   - **Given** I have applied specific filters,

- **Then** my filter preferences are maintained during my session.
- **And** I can easily clear or modify filters as needed.

Notes/Assumptions:

- Search functionality extends across all workspace components.
- System provides relevant and ranked search results.
- Extension points allow for enhanced search capabilities.
- Integration with external systems may provide additional search features.

2.23. User Story #21: Multilingual Interface Support

ID: US021 (Marito)
Title: Multilingual Interface Support
As a: User who speaks a language other than English,
I want: to access and use the Marito platform in my preferred language,
So that: I can navigate and interact with the system more comfortably and efficiently without language barriers.

Acceptance Criteria:

1. **Language Selection:**
   - **Given** I am a new or returning user,
   - **When** I access the application settings or profile,
   - **Then** I can select my preferred interface language from a list of supported languages.
   - **And** my language preference is saved for future sessions.
2. **Dynamic Language Switching:**
   - **Given** I am using the application,
   - **When** I change my language preference,
   - **Then** the interface language changes immediately without requiring a page reload.
   - **And** all interface elements adapt to the selected language.
3. **Localized Content Display:**
   - **Given** I have selected my preferred language,
   - **When** I navigate through the application,
   - **Then** all UI elements, menus, buttons, and system messages are displayed in my chosen language.
   - **And** dates, numbers, and other locale-specific content follow my language's conventions.

4. **Fallback for Untranslated Content:**
    - **Given** I am using a language that has partial translation coverage,
    - **When** I encounter content without translation in my selected language,
    - **Then** the system displays that content in the default language (English).
    - **And** the transition between translated and untranslated content is seamless.
5. **Offline Language Support:**
    - **Given** I have set my language preference,
    - **When** I use the application offline,
    - **Then** my selected language continues to be applied.
    - **And** I don't need an internet connection to use the interface in my language.
6. **User Contribution to Translations (Optional Enhancement):**
    - **Given** I notice missing or incorrect translations in my language,
    - **When** I use the translation contribution feature,
    - **Then** I can suggest improved translations for specific interface elements.
    - **And** my suggestions are reviewed by moderators before implementation.

Notes/Assumptions:

- Initial release supports at least 5 major languages (English, Spanish, French, Mandarin, Arabic).
- Language selection does not affect user-generated content or terminology data.
- The system uses internationalization (i18n) best practices for language implementation.
- Right-to-left (RTL) languages like Arabic are properly supported with appropriate UI adjustments.

2.24. User Story #22: Accessibility and Customization

ID: US022 (Marito)
Title: Accessibility and Visual Customization
As a: User with accessibility needs or visual preferences,
I want: to be able to customize the visual presentation of the application including text size, spacing, contrast, and color themes,
So that: I can have a comfortable and accessible reading experience that meets my specific needs.

Acceptance Criteria:

1. **Adjusting Text Size:**
    - **Given** I am using the application,

- **And** I am viewing the accessibility settings,
- **Then** I should see an option to adjust text size,
- **And** I should be able to increase or decrease the text size with real-time preview,
- **And** I should see the current size displayed (e.g., 16px).

2. **Modifying Text Spacing:**
   - **Given** I am using the application,
   - **And** I am viewing the accessibility settings,
   - **Then** I should see an option to adjust text spacing,
   - **And** I should be able to change the spacing between lines and letters,
   - **And** I should see the current spacing multiplier displayed (e.g., 1x).
3. **Enabling High Contrast Mode:**
   - **Given** I am using the application,
   - **And** I am viewing the accessibility settings,
   - **Then** I should see an option to enable High Contrast Mode,
   - **And** I should be able to toggle it on or off,
   - **And** When enabled, all colors should have increased contrast for better visibility.
4. **Switching to Dark Mode:**
   - **Given** I am using the application,
   - **And** I am viewing the accessibility settings,
   - **Then** I should see an option to enable Dark Mode,
   - **And** I should be able to toggle it on or off,
   - **And** When enabled, the interface should display dark backgrounds with light text.
5. **Persistence of Settings:**
   - **Given** I have customized my accessibility preferences,
   - **When** I log out and log back in,
   - **Then** all my accessibility settings should be preserved,
   - **And** The application should immediately apply my saved preferences.

Notes/Assumptions:

- Users must be authenticated to save accessibility preferences.
- All accessibility changes should take effect immediately without page refresh.
- Settings should work together without conflicting (e.g., dark mode + high contrast).
- Visual customizations should maintain readability at all adjustment levels.

3. Use Case Diagrams

This section presents the key use case diagrams developed for the Marito application. Each diagram visualizes the interactions between user roles and the system, based on the user stories defined earlier in this document. These diagrams serve as a visual blueprint for understanding the system's expected behavior.

3.1. Registration and Login Use Case Diagram

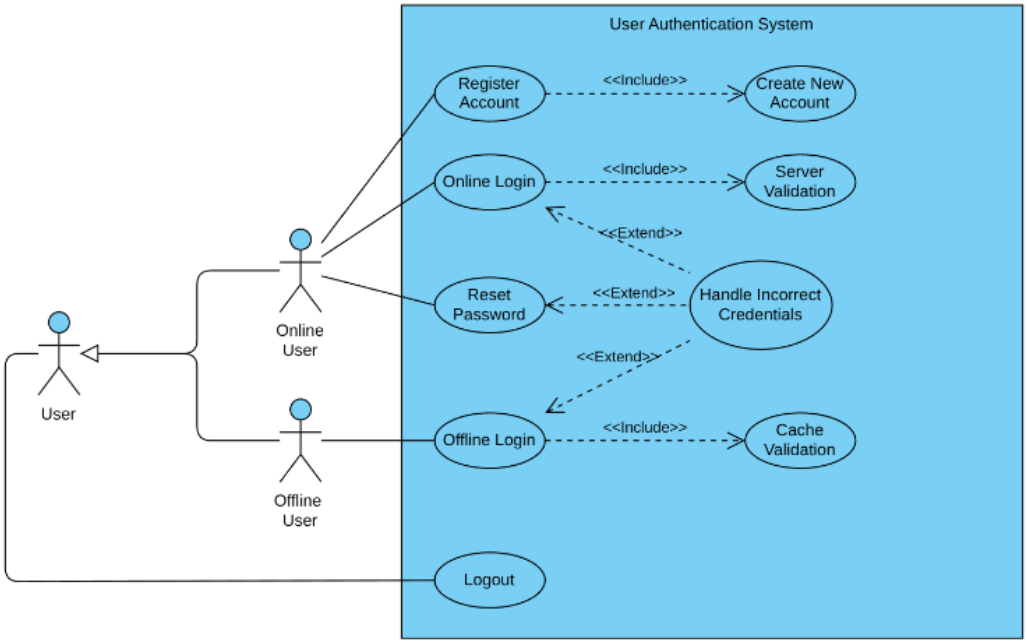

Figure 1: Registration and Login Use Case Diagram

3.2. Accessibility and Customization Use Case Diagram

Figure 2: Accessibility and Customization Use Case Diagram for Marito

3.3. Search Term Use Case Diagram

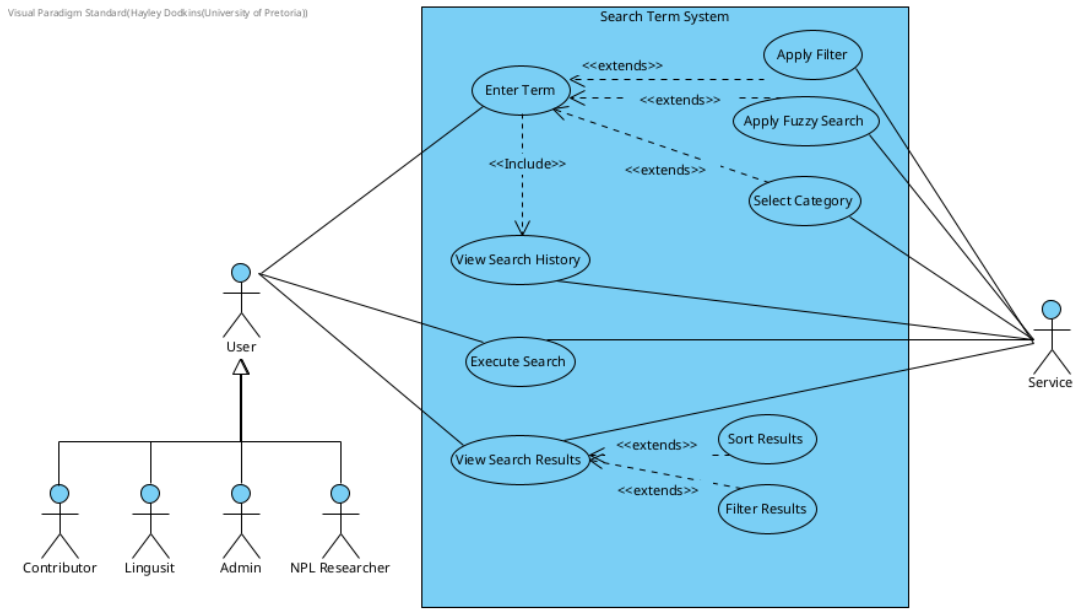

Figure 3: Search Term Use Case Diagram for Marito

3.4. Gamification Use Case Diagram

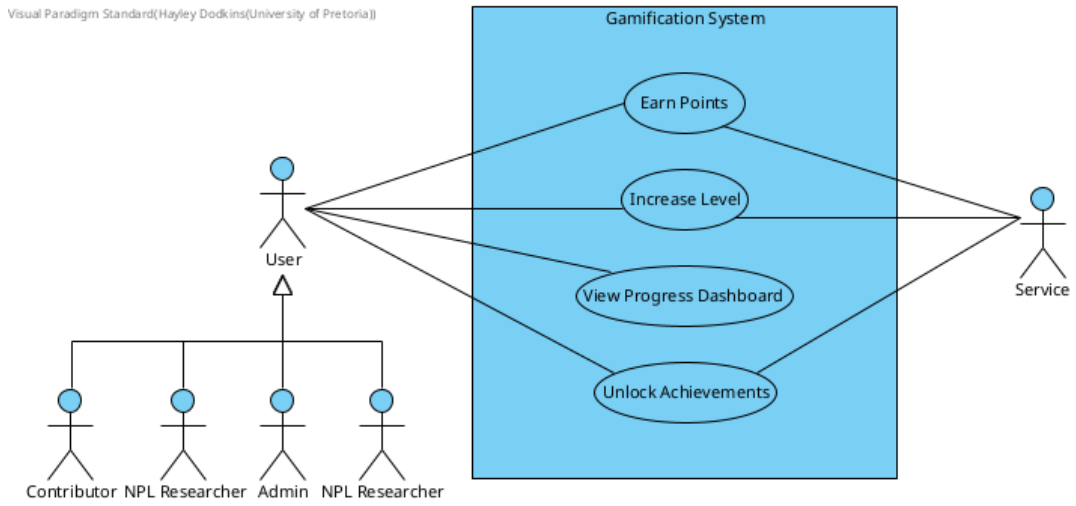

Figure 4: Gamification Use Case Diagram

3.5. Visualization Use Case Diagram

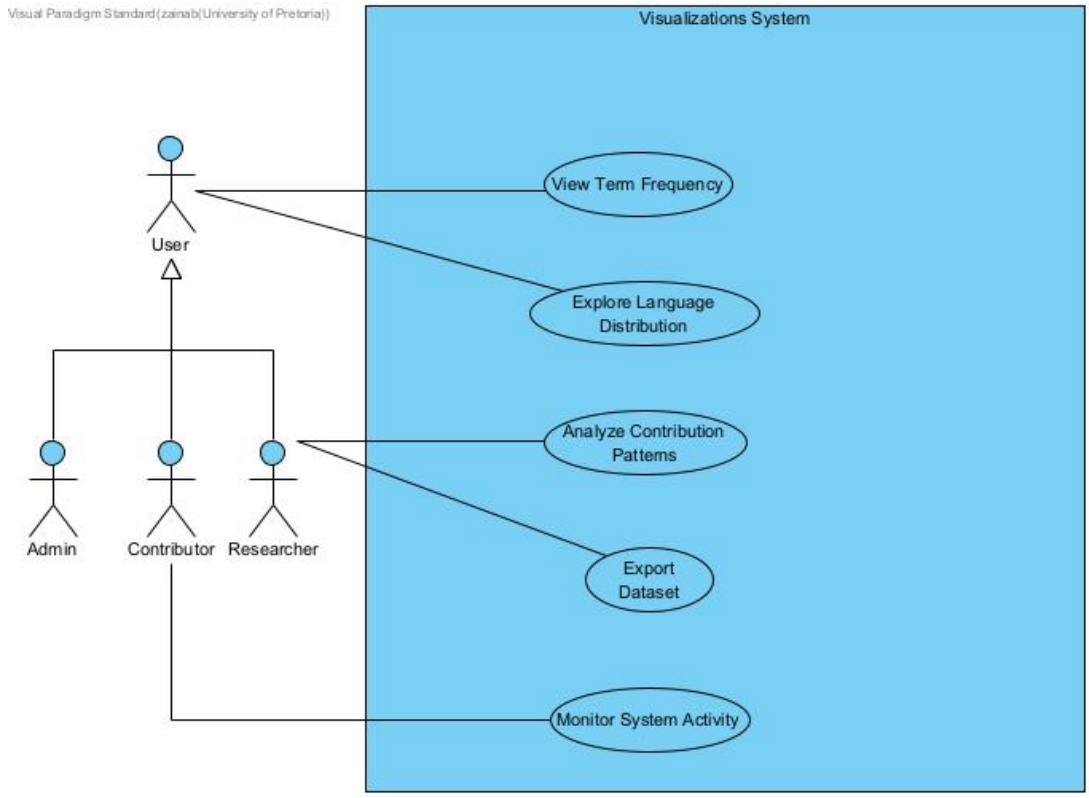

Figure 5: Visualization Use Case Diagram

3.6. Contributions Use Case Diagram

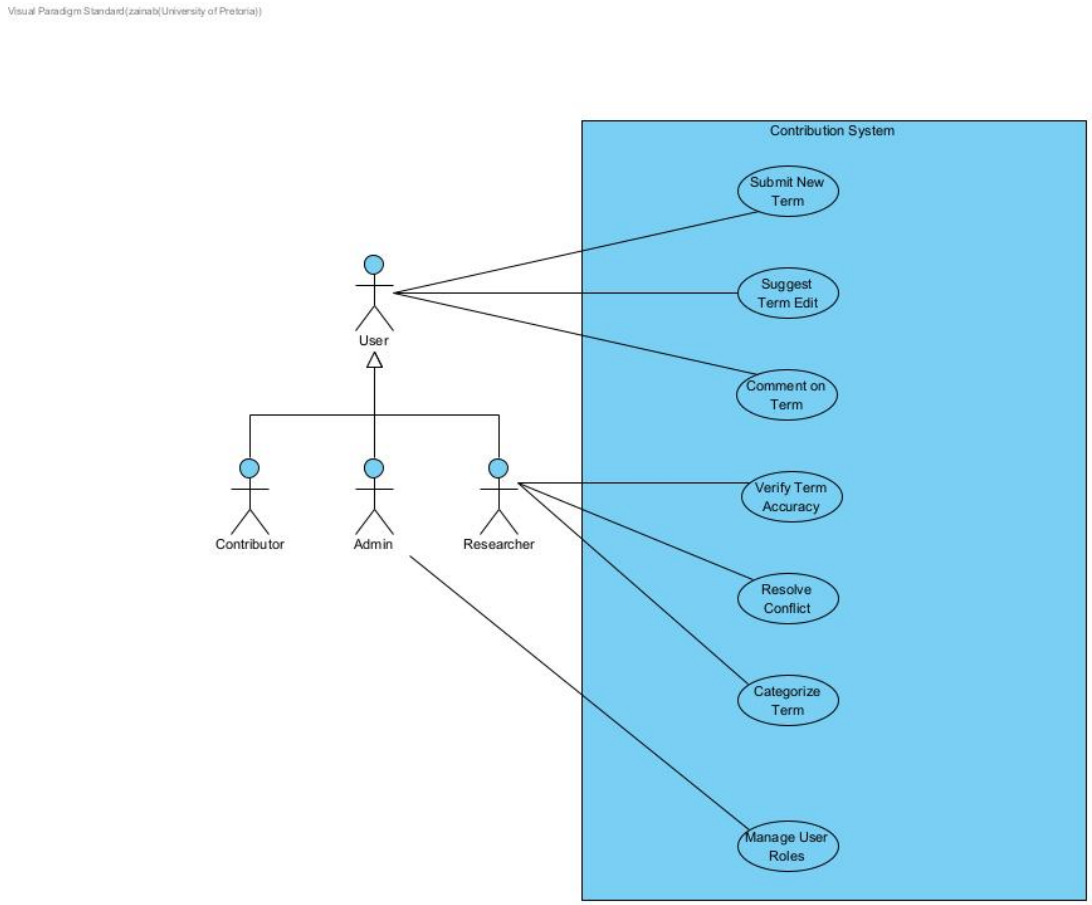

Figure 6: Contributions Use Case Diagram

3.7. Dictionary Use Case Diagram

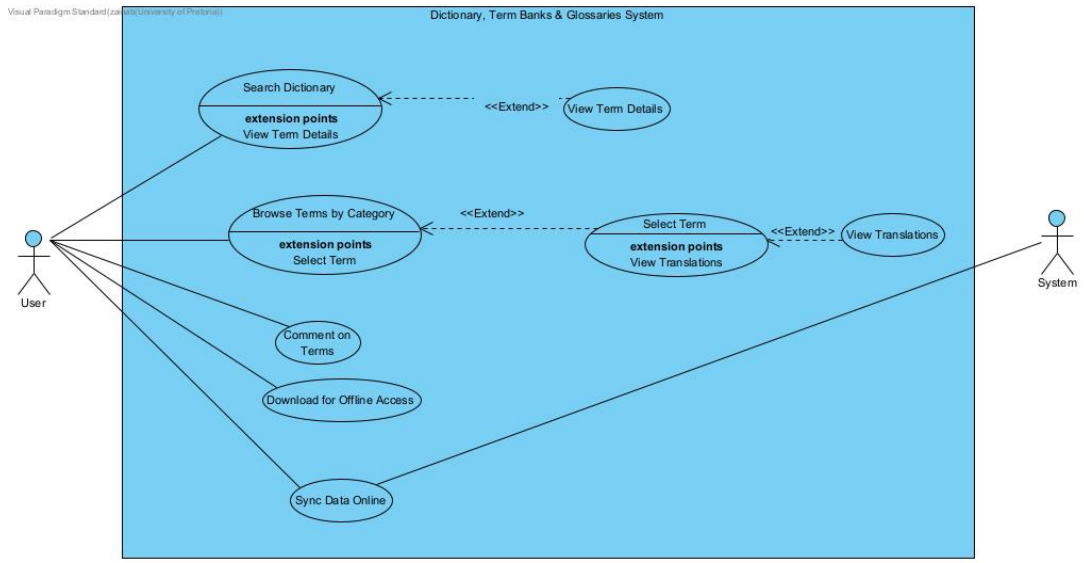

Figure 7: Dictionary and Glossary Use Case Diagram

3.8. Admin Use Case Diagram

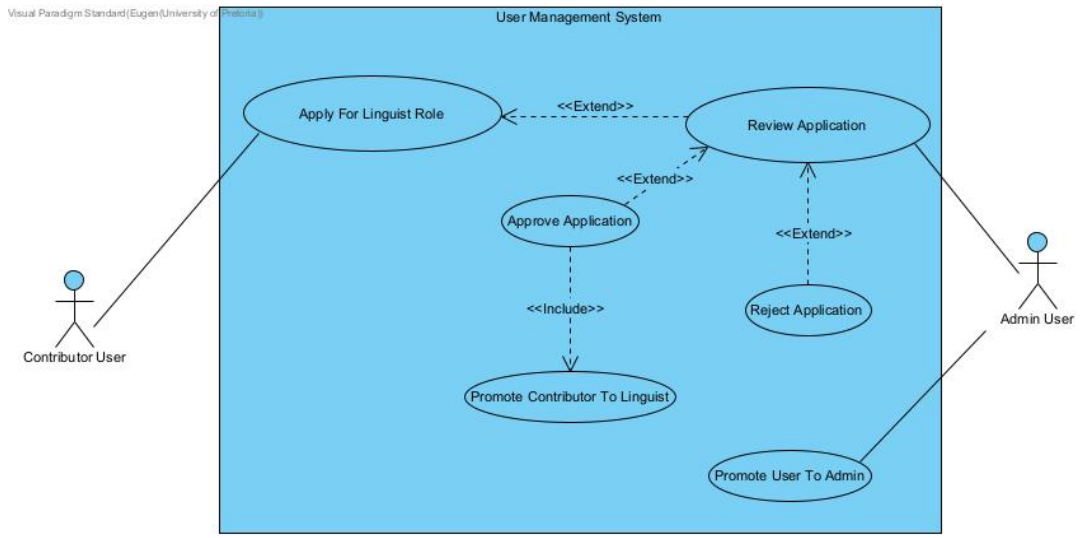

Figure 8: Admin User Management Use Case Diagram

3.9. Export Use Case Diagram

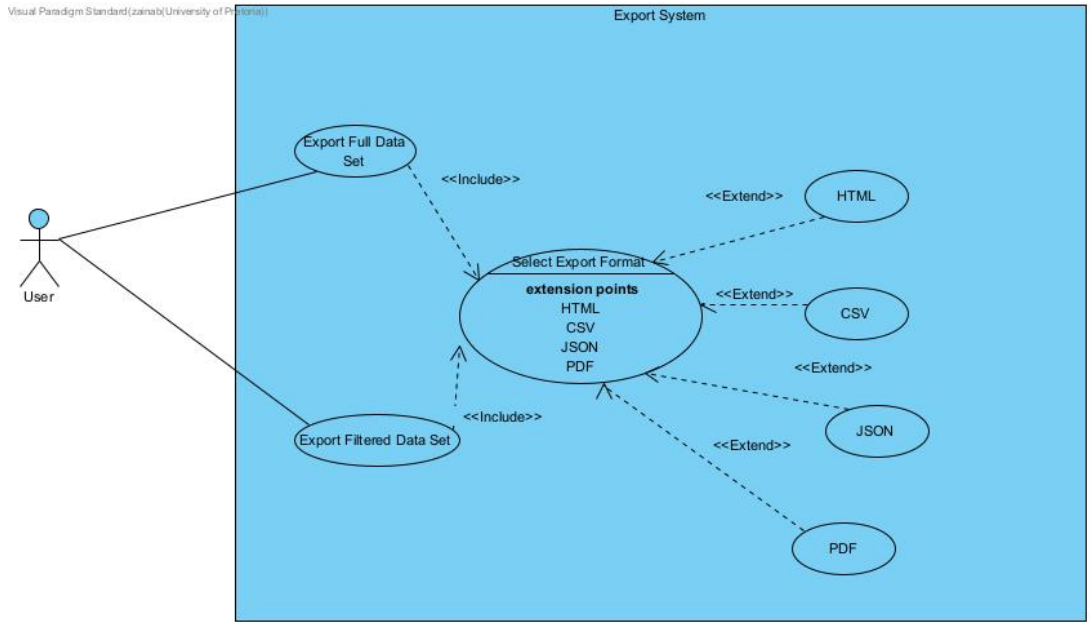

Figure 9: Export System Use Case Diagram

3.10. Feedback System Use Case Diagram

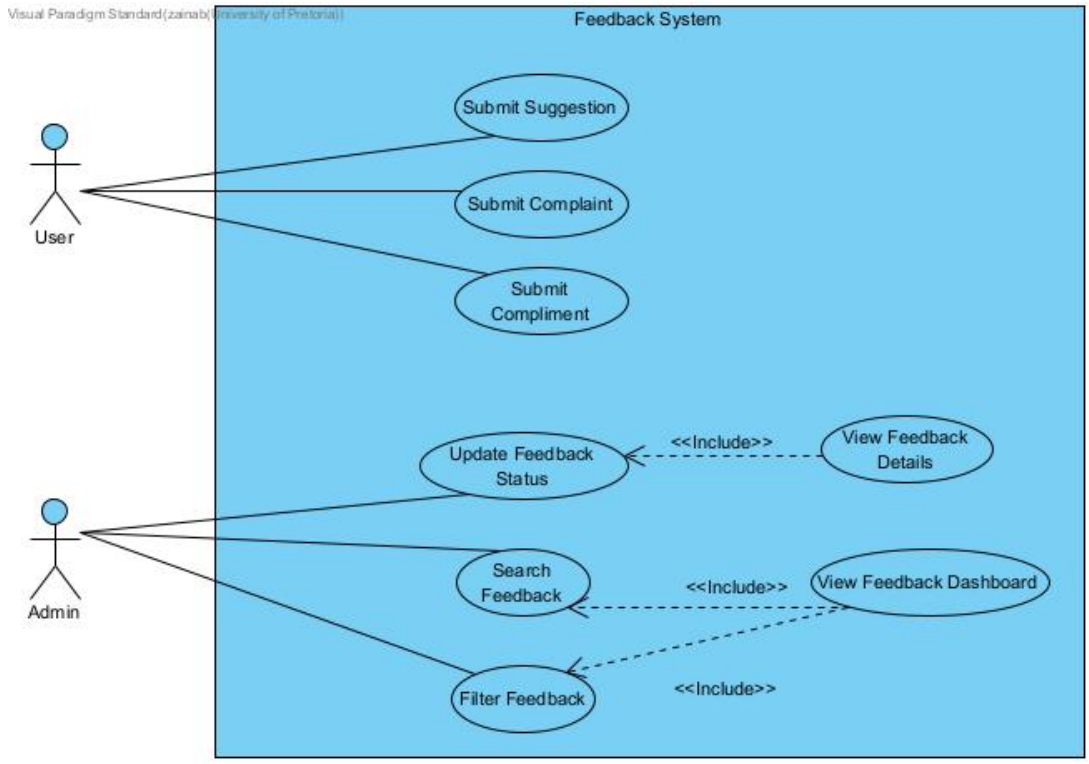

Figure 10: Feedback System Use Case Diagram

3.11. Workspace Use Case Diagram

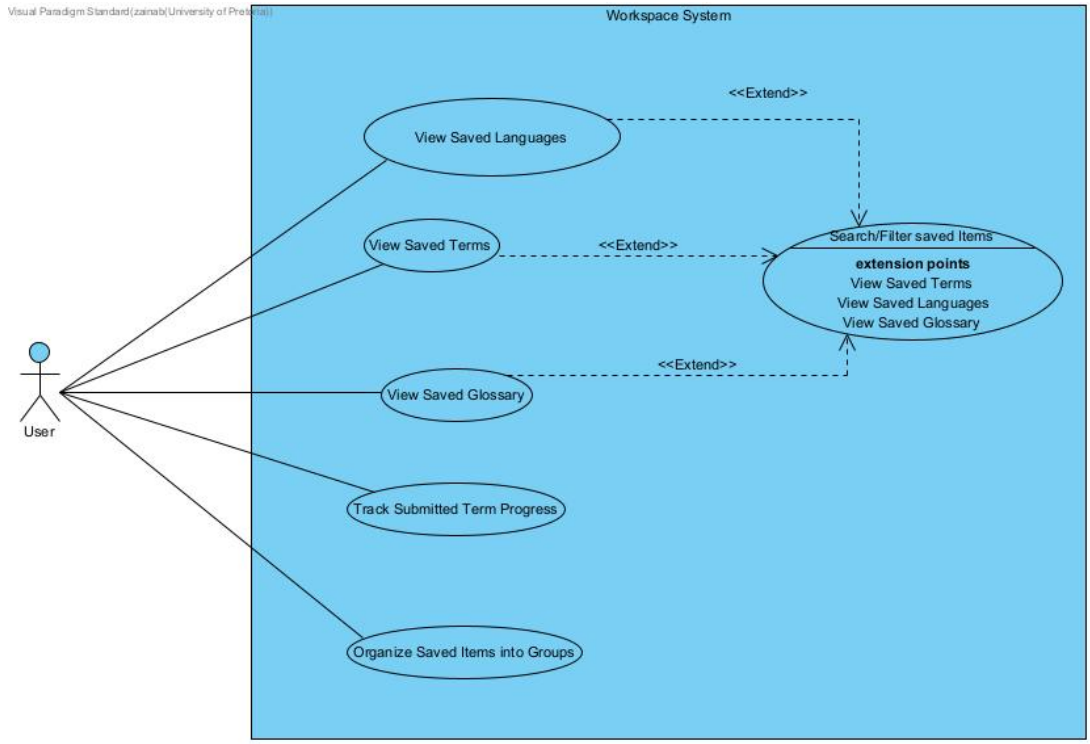

Figure 11: Workspace Management Use Case Diagram

3.12. Overview

The main use cases covered include:

- Browsing and searching multilingual terms
- Accessibility and customization
- Earning contribution points through gamified interactions
- Registration and login interactions
- User Contributions
- Data visualizations
- Dictionary Use Case
- Admin Use case
- Export Use Case
- Feedback System Use Case
- Workspace Management Use Case

4. Functional Requirements

FR1: **Glossary Browsing and Search**

- FR1.1: The system shall allow users to browse available glossaries and term banks.
- FR1.2: The system shall provide a unified search interface across all multilingual glossaries.
- FR1.3: The system shall allow users to search for terms using exact matches, partial strings, or semantic similarity.
- FR1.4: The search results shall be ranked by relevance and display key metadata (definition, part of speech, language).
- FR1.5: The system shall highlight related terms and translations in the term view.

FR2: **Accessibility and Customization**

- FR2.1: Profile & Account Management
  - FR2.1.1: The system shall allow users to upload and change profile pictures.
  - FR2.1.2: The system shall enable username updates with real-time validation for uniqueness.
  - FR2.1.3: The system shall require password validation before email changes and send verification emails.
  - FR2.1.4: The system shall enforce secure password changes requiring current password validation.
  - FR2.1.5: The system shall provide secure logout that terminates all sessions.
- FR2.2: Multilingual Support
  - FR2.2.1: The system shall support interface localization in all eleven official South African languages including English, Afrikaans, isiZulu, isiXhosa, Sesotho, Setswana, Sepedi, isiSwati, Tshivenda, Xitsonga, and isiNdebele.
  - FR2.2.2: The system shall allow users to switch UI language anytime through settings.
  - FR2.2.3: The system shall correctly render special characters for all supported languages.
  - FR2.2.4: The system shall persist language preferences across sessions.
  - FR2.2.5: The system shall apply language changes immediately without page refresh.
- FR2.3: Text & Visual Presentation
  - FR2.3.1: The system shall allow users to adjust text size and display current size in pixels (e.g., 16px).
  - FR2.3.2: The system shall provide real-time text size adjustment.

  - FR2.3.3: The system shall show current text size value as users make changes.
  - FR2.3.4: The system shall maintain text size preferences across sessions.
  - FR2.3.5: The system shall ensure text scaling maintains proper layout.
- FR2.4: Text Spacing Control
  - FR2.4.1: The system shall allow users to adjust text spacing and display current multiplier (e.g., 1x).
  - FR2.4.2: The system shall provide real-time spacing adjustment.
  - FR2.4.3: The system shall show current spacing value as users make changes.
  - FR2.4.4: The system shall apply spacing changes to all text elements consistently.
  - FR2.4.5: The system shall preserve spacing preferences across sessions.
- FR2.5: Colour & Contrast Controls
  - FR2.5.1: The system shall allow users to enable/disable High Contrast Mode.
  - FR2.5.2: The system shall show current mode state (on/off).
  - FR2.5.3: The system shall increase color contrast when high contrast mode is enabled.
  - FR2.5.4: The system shall maintain high contrast settings across sessions.
  - FR2.5.5: The system shall ensure interactive elements remain distinguishable in high contrast mode.
- FR2.6: Dark Mode
  - FR2.6.1: The system shall allow users to enable/disable Dark Mode.
  - FR2.6.2: The system shall show current mode state (on/off).
  - FR2.6.3: The system shall apply dark backgrounds with light text when enabled.
  - FR2.6.4: The system shall maintain proper contrast in dark mode.
  - FR2.6.5: The system shall preserve dark mode preferences across sessions.
  - FR2.6.6: The system shall provide smooth transitions when switching modes.

FR3: **User Contributions and Feedback**

- FR3.1: The system shall allow users to submit comments.
- FR3.2: The system shall allow users to submit corrections or suggestions for terms.
- FR3.3: The system shall send submitted feedback to a backend repository for moderation.
- FR3.4: The system shall support version control for submitted glossary data and prevent overwriting of validated entries.
- FR3.5: The system should be able to allow users to upvote/downvote term changes.

FR4: **AI-Enhanced Functionality**

- FR4.1: The system shall support AI-powered semantic search to retrieve conceptually related terms.
- FR4.2: The system may provide suggested definitions or translations using AI models.
- FR4.3: The system may automatically cluster terms based on meaning or domain.
- FR4.4: The system may auto-tag glossary entries with linguistic metadata (e.g., part-of-speech).
- FR4.5: The system may integrate with external NLP APIs for research purposes.

FR5: **Gamification**

- FR5.1: The system shall assign points to users when they submit valid suggestions, comments, or issue reports on glossary terms.
- FR5.2: The system shall track milestones based on user activity and unlock achievements or badges when predefined thresholds are reached.
- FR5.3: The system shall display a progress dashboard on the user's profile page, including total points, badges earned, and contribution rank.
- FR5.4: The system shall provide real-time feedback when users perform actions that affect their gamification status.
- FR5.5: The system shall support user rank or title progression as users accumulate points and reach specific contribution levels.
- FR5.6: The system shall queue gamified contributions made offline and synchronize them with the server once the user is reconnected.
- FR5.7: The system shall validate contributions before assigning gamification points to prevent abuse.

FR6: **Progressive Web Application Functionality**

- FR6.1: The system shall function as a PWA and be installable on mobile and desktop devices.
- FR6.2: The frontend shall use service workers to cache glossary data, enabling offline access to core features such as term lookup and browsing.
- FR6.3: The system shall store user feedback or contributions made offline and queue them for submission once connectivity is restored.
- FR6.4: The application shall automatically synchronize cached content with the server when an internet connection is detected.
- FR6.5: The system shall provide visual feedback or indicators when the app is operating in offline mode.
- FR6.6: The system shall support background updates of cached glossary data to maintain consistency with the server.

- FR6.8: The system shall allow users to download selected glossaries for offline access.

FR7: **Data Visualization and Analytics**

- FR7.1: The system should be able to display stats on word frequency and usage trends.
- FR7.2: The system should be able to visualize a user's contribution to different languages.
- FR7.3: The system should highlight trending words and new entries to the database.
- FR7.4: The system should be able to provide interactive charts or graphs for linguistic data.

FR8: **Data Import and Export**

- FR8.1: The system shall allow users to export glossary data in JSON format.
- FR8.2: The system shall allow users to export glossary data in CSV format.
- FR8.3: The system may allow authorized users to import glossaries in JSON or CSV format.

FR9: **Feedback System**

- FR9.1: **Feedback Submission**
  - FR9.1.1: The system shall allow users to submit feedback through multiple categories: Suggestion, Complaint, and Compliment.
  - FR9.1.2: The system shall allow users to optionally provide their name and email address with feedback submissions.
  - FR9.1.3: The system shall require a descriptive message for all feedback submissions.
  - FR9.1.4: The system shall validate that required fields are completed before allowing submission.
  - FR9.1.5: The system shall provide appropriate guidance text based on the selected feedback category.
  - FR9.1.6: The system shall accept anonymous feedback submissions when contact information is not provided.
- FR9.2: **Feedback Administration**
  - FR9.2.1: The system shall provide administrators with a dashboard showing feedback statistics and trends.
  - FR9.2.2: The system shall display all feedback items with category, status, priority, user information, and submission date.
  - FR9.2.3: The system shall allow administrators to filter feedback by type (suggestions, complaints, compliments) and status (pending, in-progress, resolved).
  - FR9.2.4: The system shall provide search functionality across feedback content and user information.

- FR9.2.5: The system shall allow administrators to update the status of feedback items through a defined workflow.
- FR9.2.6: The system shall track all status changes with timestamps for audit purposes.

FR10: **Workspace Management**

- FR10.1: The system shall provide a personalized workspace where users can organize, access, and manage their saved language resources and contributions.
- FR10.2: **Saved Terms**
  - FR10.2.1: The system shall allow users to bookmark individual terms from dictionaries/glossaries for quick access.
  - FR10.2.2: The system shall display saved terms in a searchable, filterable list (e.g., by language, tags, or date saved).
  - FR10.2.3: The system shall enable users to add custom notes or annotations to saved terms.
  - FR10.2.4: The system shall sync saved terms across devices when online and when offline.
- FR10.3: **Saved Glossaries**
  - FR10.3.1: The system shall allow users to download entire glossaries/dictionaries for offline access.
  - FR10.3.2: The system shall let users mark glossaries as "favorites" for quick filtering.
  - FR10.3.3: The system shall notify users when a saved glossary has updates (e.g., new terms or corrections).
  - FR10.3.4: The system shall provide storage management (e.g., "Delete Glossary" to free space).
- FR10.4: **Contributed Terms**
  - FR10.4.1: The system shall maintain a history of user-submitted terms, definitions, or edits.
  - FR10.4.2: The system shall display the moderation status of contributions (e.g., "Pending," "Approved," "Rejected").
- FR10.5: **Workspace Organization**
  - FR10.5.1: The system shall allow users to create folders/tags to group saved terms or glossaries (e.g., "Thesis Research").
  - FR10.5.2: The system shall support bulk actions (e.g., "Move 5 terms to Folder X").

5. Non-Functional Requirements

NFR1: **Offline Accessibility**

- NFR1.1 The system shall support offline access to previously downloaded language resources.

- NFR1.2 The frontend shall use service workers and caching to allow uninterrupted use of core features without an active internet connection.
- NFR1.3 Synchronization of updated resources shall automatically occur once the device is back online.

NFR2: **Performance and Responsiveness**

- NFR2.1 The system shall deliver fast search responses, with results displayed within 2 seconds for standard queries.
- NFR2.2 The frontend application shall load and become interactive within 3 seconds on devices with moderate hardware and average bandwidth.
- NFR2.3 UI interactions shall be smooth and not exceed 100ms latency where possible.

NFR3: **Scalability**

- NFR3.1 The backend shall be able to handle simultaneous requests from a growing user base, including researchers and contributors.
- NFR3.2 The data architecture must accommodate the addition of new glossaries, languages, and APIs without the need for major refactoring.

NFR4: **Security and Privacy**

- NFR4.1 All data transmissions between client and server shall be encrypted using HTTPS.
- NFR4.2 User-submitted feedback shall be sanitized and validated to prevent injection attacks.
- NFR4.3 The backend shall implement basic access control to restrict sensitive actions to authorized roles.

NFR5: **Usability and Accessibility**

- NFR5.1 The user interface shall support all 11 official South African languages through dynamic localization.
- NFR5.2 The application shall conform to WCAG 2.1 Level AA accessibility guidelines to accommodate visually impaired users.
- NFR5.3 The system shall provide a clean and intuitive interface that requires no more than three clicks to reach key features.

NFR6: **Maintainability**

- NFR6.1 The codebase shall follow modular and well-documented design practices to enable ease of maintenance.
- NFR6.2 The frontend and backend shall use consistent code formatting enforced via linting tools.
- NFR6.3 New contributors shall be able to understand and modify the codebase with minimal onboarding effort.

NFR7: **Extensibility**

- NFR7.1 The system shall support the addition of new term banks and glossaries via a modular import system.
- NFR7.2 New frontend features shall be integrable without affecting the core application features.
- NFR7.3 The backend shall expose extensible REST API routes following OpenAPI standards.

NFR8: **Reliability and Fault Tolerance**

- NFR8.1 The system shall gracefully handle failed API requests and notify the user when a feature is temporarily unavailable.
- NFR8.2 The frontend shall include fallback mechanisms for critical resources, ensuring minimal disruption in case of partial data loss.
- NFR8.3 The backend shall log all failed operations and expose logs for future debugging or auditing.

NFR9: **Portability and Cross-Platform Compatibility**

- NFR9.1 The PWA shall work on major browsers (Chrome, Firefox, Edge, Safari) and platforms (Windows, Android, iOS, Linux).
- NFR9.2 The UI shall be fully responsive and usable on screen sizes ranging from smartphones to desktop monitors.
- NFR9.3 No feature shall be dependent on platform-specific behavior.

NFR10: **Deployment and DevOps Readiness**

- NFR10.1 The backend shall be containerized using Docker to support consistent deployment across environments.
- NFR10.2 CI/CD pipelines shall be configured to run tests, build artifacts, and deploy the application to the cloud or GitHub Pages.

6. Constraints

6.1. Overview

This section outlines the constraints that govern the design, development, deployment, and maintenance of the **Marito** application. These constraints stem from ethical considerations, client requirements, project context, budgetary limitations, and architectural guidelines. Adherence to these constraints is critical to the project's success and sustainability.

6.2. Constraints

2.1 **Privacy and Data Minimization**

- The application must adopt a privacy-first approach.
- No personal user information may be collected unless voluntarily submitted.
- No personal data should be stored without explicit user consent.

- Any user data collected must comply with ethical research standards and university data policies.
- All stored data must follow ethical and data protection principles.

2.2 Maintainability and Sustainability

- The project must use a maintainable and reliable technology stack that future students or stakeholders can easily understand and extend.
- All source code must be well-documented, modular, and follow clean architecture principles.
- Technologies selected must have active community support and comprehensive documentation to minimize onboarding effort.

2.3 Use of Open Source Technologies

- Wherever possible, the project must use open-source frameworks and libraries.
- External packages or modules must have permissive licenses (e.g., MIT, Apache 2.0) and be cited appropriately.

2.4 Budget Constraints

- The project must be developed with minimal to zero cost.
- The team is not permitted to incur costs unless explicitly approved by the client.
- Hosting should utilize free tiers.

2.5 Offline Functionality

- The application must support offline access as a core feature.
- Core features must function without an active internet connection, using Progressive Web App (PWA) technologies such as service workers and local caching.
- Synchronization with the central repository must occur seamlessly once connectivity is restored.

2.6 Version Control of Glossary Data

- Glossary data and language resources must be version-controlled to prevent loss or corruption of validated content.
- User feedback must be stored separately and reviewed before merging into the main dataset.
- No glossary data may be overwritten without proper validation and versioning mechanisms.

2.7 Ethical and Legal Considerations

- No proprietary or restricted datasets may be used without explicit authorization.
- Contributions and authorship must be tracked where possible to preserve academic and community recognition.

2.8 **Performance and Efficiency**

- The application must be responsive and performant on low-resource or mobile devices.
- Heavy assets (e.g., large datasets or images) must be lazy-loaded or paginated.
- Search and filtering mechanisms must be optimized for performance on slower devices and large datasets.

2.9 **Accessibility Requirements**

- The application must adhere to WCAG 2.1 AA accessibility standards.
- Features such as text-to-speech, keyboard navigation, color contrast, and screen reader support should be considered where feasible.
- Dark mode and adjustable text size are encouraged to support visual diversity in users.

2.10 **Deployment and Portability**

- The system must be deployable using containerization to support ease of testing and reproducibility.
- CI/CD pipelines must be used for automated builds and testing.
- Both frontend and backend should be deployable without manual intervention.

2.11 **Multi-Language Support**

- The interface must support multiple South African languages.
- The system must be internationalization-ready using industry-standard practices (e.g., translation JSON files, locale selectors).
- UI elements, glossaries, and feedback must render correctly regardless of language.

6.3. Summary

These constraints serve as the foundation for all architectural, design, and implementation decisions made during the development of **Marito**. They ensure the solution is ethical, sustainable, inclusive, affordable, and aligned with the client's goals of making multilingual resources accessible to all.

7. Domain Model

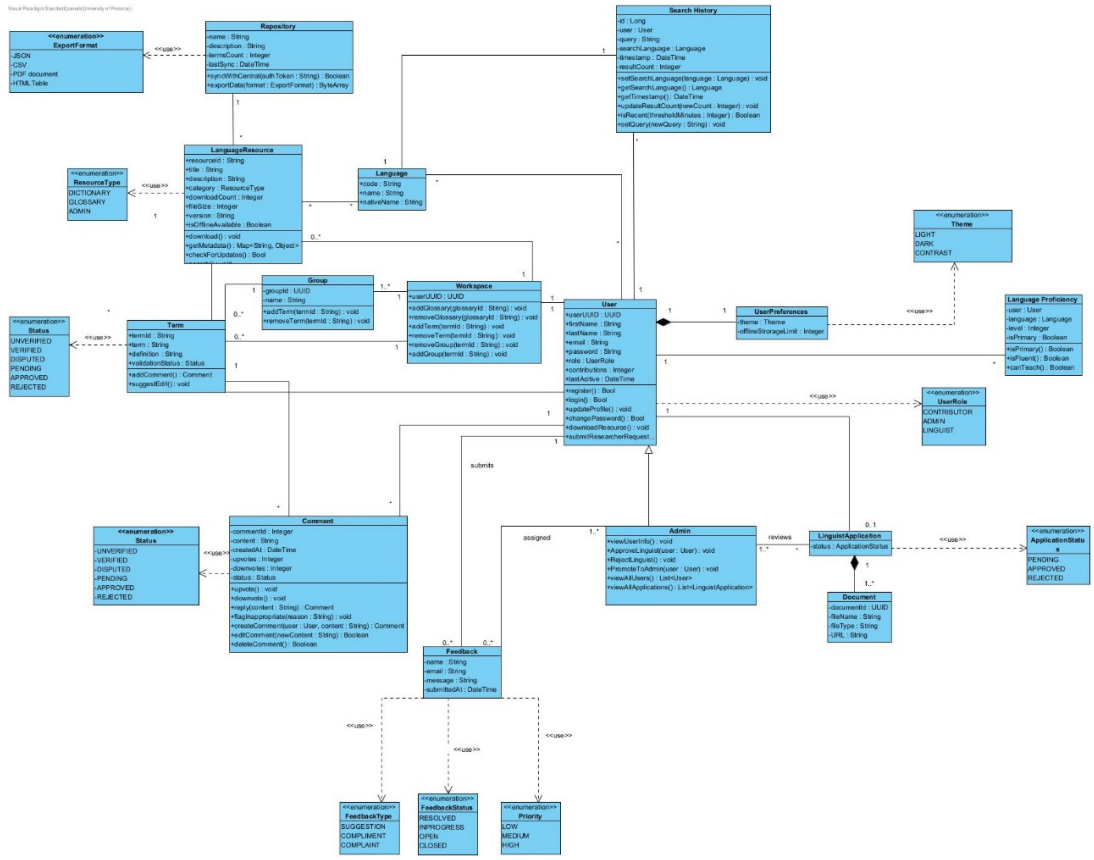

Figure 12: Domain Model

Architectural Specifications & Service Contract