ITLingoCloud Usability Assessment Guide (2026)

Vasco Marques, Eduardo Guerra (IST, Universidade de Lisboa, MSc Students)

Alberto Rodrigues da Silva (Supervisor),

v 1.1, August 2026

(PDF available at https://cloud.itlingo.pt/public-workspaces/242/documents/138 or from public workspace https://cloud.itlingo.pt/public-workspaces/242/documents)

User Assessment Goals

The purpose of this evaluation is to assess the usability and relevance of the ITLingoCloud platform, with emphasis on its Chatbot (ITLingo-Chatbot), Specifications (ITLingo-ITOI), and Documents (Document Automation) components in supporting requirements engineering activities based on domain-specific languages (DSLs), in particular the RSL and ASL languages.

This is a task guide organized into five parts (Parts 1 to 5), each containing user tasks (Tasks 1 to 14). Each task defines a Goal and numbered subtasks. Follow the tasks in order and perform each action on the platform. For this evaluation, you have been invited as a member of a demo organization that already contains shared documents you will use along the way. In particular, the demo organization's name in this experiment is **"LibraryTech (Demo)"**.

During this evaluation, you will:

- Join the demo organization and create your own workspace;
- Browse the organization's shared documents and download the ones you may need;
- use the **Chatbot** component to briefly learn about the DSLs, generate specifications and UML-like diagrams, and transform RSL specifications into ASL specifications;
- Edit and validate an RSL specification with the **ITOI** (ITLingo Online IDE) component;
- Generate requirements documents in several formats (e.g., Word, Markdown, JSON) using the **Documents** (templating engine) component.

This experience is divided into tasks grouped into the following parts:

P1 — Joining the Organization and Creating a Workspace (Task1–T3)

P2 — Organization Documents (T4)

P3 — Chatbot and AI Capabilities (T5 -T11)

P4 — Specifications with ITOI (T12–T13)

P5 — Document Templating (T14)

Complete the tasks in order whenever possible, and at the end of your experience, submit your assessment and feedback in the provided questionnaire.

If a subtask produces unexpected results, continue with the next subtask whenever possible and note your observations. You may pause the evaluation at any time and continue later: your workspaces, documents, and chats are automatically saved.

Part1. Joining the Organization and Creating your Workspace

Task1. Access the platform and sign in

Goal: Access ITLingoCloud with your user account.

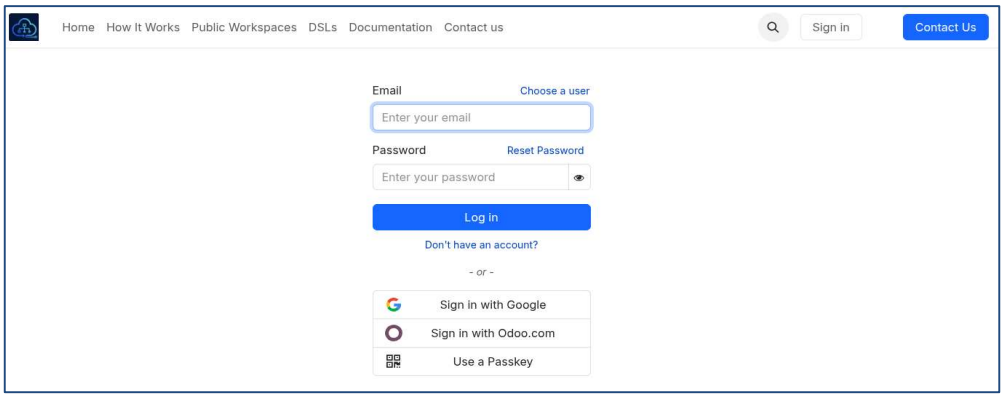
Task1.1. Open your browser and navigate to https://cloud.itlingo.pt.

Task1.2. Click **Sign in** in the top-right corner.

If you do not have an account yet, choose **Don't have an account?** and register:
- fill in your **Name**, **Email,** and **Password** (and confirm the password);
- read and accept the **Terms and Conditions** checkbox, then click **Sign up**.

Task1.3. Sign in with your account.

Use the **same email address** you used to receive the invitation to the demo organization; this is how the platform links the invitation to your account.



Sign-in / sign-up page

Task2. Open the demo organization and explore it

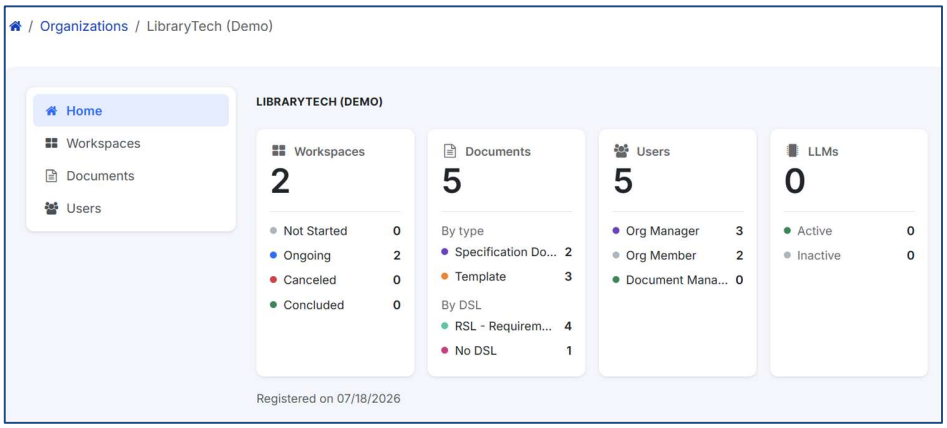
Goal: Get familiar with how an organization is structured. An organization is the top-level container that groups workspaces, members, and shared documents.

Because you were **invited before you registered**, your membership was created at the moment of the invitation: as soon as you sign up with the invited email, you are **already an "Organization Member"** of the demo organization, i.e., there is no invitation to accept.

Task2.1. Open the top menu Organizations. The demo organization (named "**LibraryTech (Demo)**") is already listed under your Org Member role. Open it.

If you already had an account when you were invited, you would instead have received a notification (via the bell at the top of the page and by email) to accept; the acceptance step applies only to existing users.

Task2.2. Explore the sections available in the organization's left sidebar. As an **Org Member,** you shall see:
- **Home** — overview of the organization;
- **Workspaces** — the organization's workspaces (you shall see only the Workspaces created by you or to which you were invited);
- **Documents** — documents shared at organization level (you will use these in P2);



Organization page with the sidebar visible

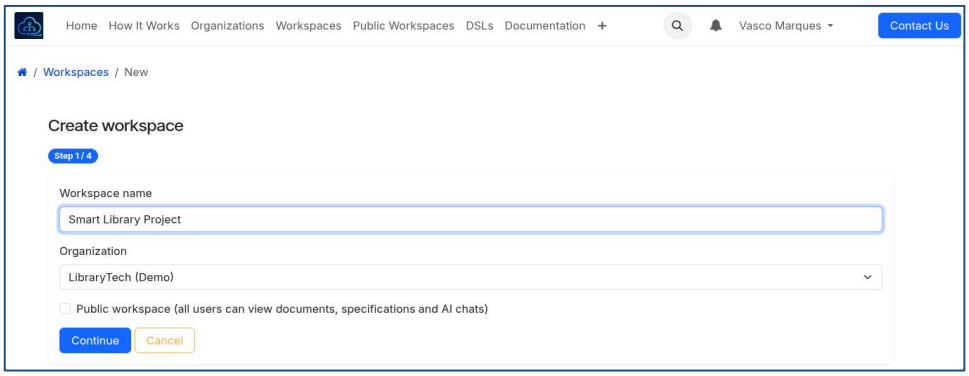
Task2. Create a workspace

Goal: Create your own workspace, i.e., the working area where you keep the specifications and documents of your experience or project.

Task3.1. In the organization sidebar, open **Workspaces** and click **New workspace**.

The creation wizard has 4 steps:

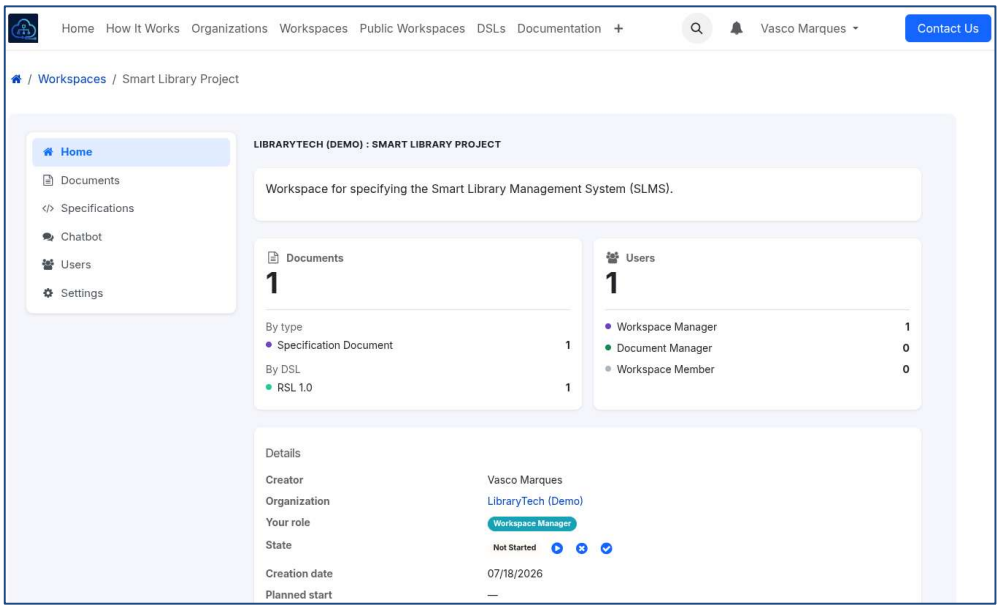
1. **Details** — enter the **Workspace name** and confirm the organization. Please use the naming pattern **`SLMS - <your first and last name>`** (e.g., `SLMS - Maria Santos`). Leave the **Public workspace** checkbox **unchecked,** so your workspace stays private (only you and invited members can see its documents, specifications, and AI chats);
2. **Planning** — optionally set planned/actual start and end dates (all dates are optional); click **Continue**;
3. **Description** — optionally describe the workspace; click **Continue**;
4. **Review** — check the summary and click **Create workspace**.



Workspace creation wizard, step 1

Task3.2. When you are on the workspace page, explore the sections available from the left sidebar menu:

- **Home** — workspace overview;
- **Documents** — the workspace's documents;
- **Specifications** — access to the ITOI online IDE to rigorously write and validate specifications;
- **Chatbot** — access to the AI assistant for this workspace (used in the Chatbot guide);
- **Users** — to manage workspace members and roles;
- **Settings** — workspace details (visible to workspace managers only).



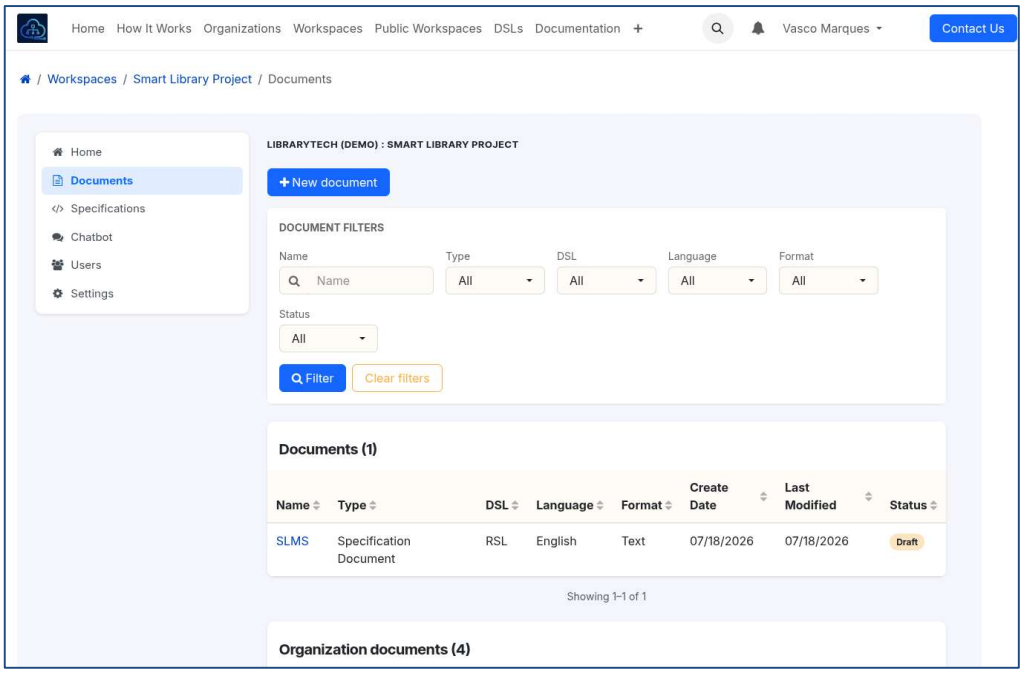
Workspace page with the sidebar visible

Part2. Organization Documents

Task3. Explore and download the organization's documents

Goal: See how documents published at organization level are available inside your workspace and download the material you will use in the chatbot and later tasks.

Task4.1. In your **workspace** sidebar, open **Documents**. Besides your own (empty) library, you can see the documents **published at the organization** level — they are shared with every workspace of the organization without being copied.

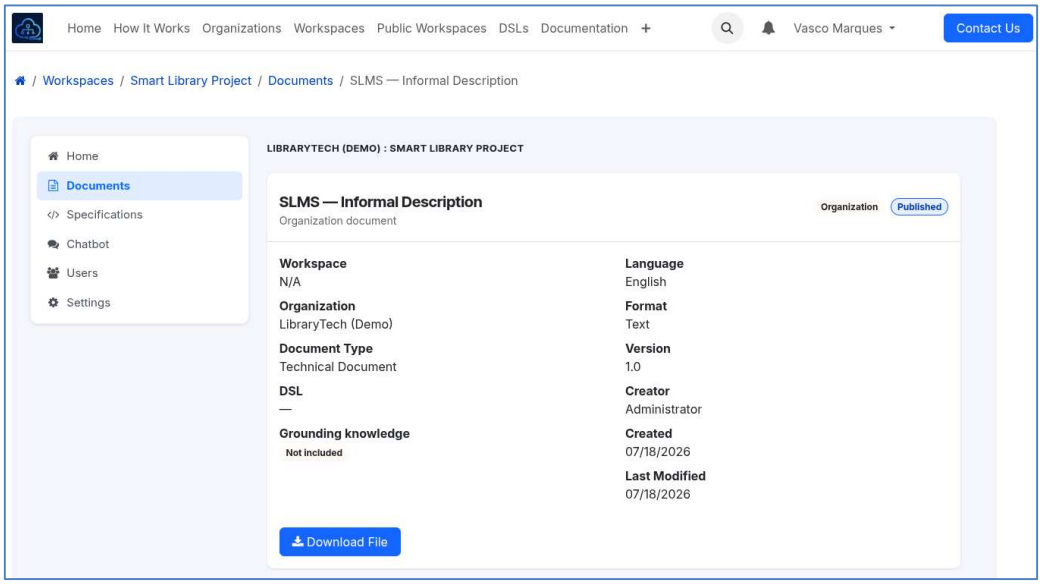


Workspace Documents page showing the organization's published documents.

Task4.2. Review the shared documents. They include:

- **SLMS informal specification**: a PDF describing the SLMS system in natural language — the starting point for the specification you will draft with the chatbot;
- **SLMS RSL specification (Complete)**: a complete, working RSL specification of the system, provided as a backup in case you cannot produce a good one yourself (used in P4);
- **RSL Requirements Document Template (...)**: three RSL requirements document templates (Word, Markdown, JSON) that you will use to generate requirements documents directly in P5 (no need to download them).

Task4.3. Download the document "**SLMS informal specification**" (use the download action on each document, or open its detail page and download from there). The downloaded file is named SLMS-CaseStudy-Definition-v1.1-1.pdf. Keep it handy — you will use it in the Chatbot and AI Capabilities part.



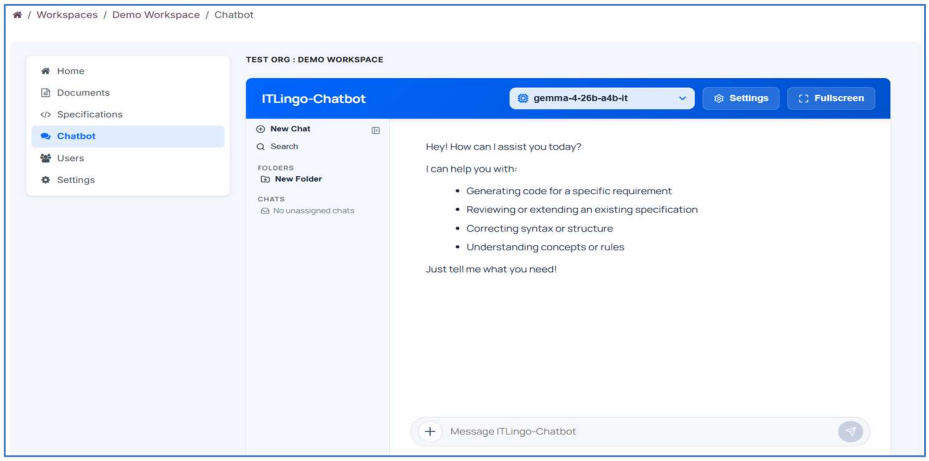
Shared organization document with the Download File action

Part3. Chatbot and AI Capabilities

Task5. Accessing the Chatbot

Goal: Open the workspace chatbot and become familiar with its location.

Task5.1. Open the Chatbot inside your workspace.



Chatbot page showing the ITLingo-Chatbot Interface

Task6. Exploring the ITLingo DSL Knowledge Base

Goal: Explore the Chatbot's built-in knowledge base to understand the syntax, concepts and modelling constructs supported by the Requirements Specification Language (RSL) and Application Specification Language (ASL).

Task6.1. Create a **new folder** inside the chatbot named "**Language Understanding**".

Task6.2. Create a **new chat** inside the previously created folder with the Prompt T6.1.

```
Prompt T6.1. Understanding the RSL Language
Explain the main elements supported by the Requirements Specification Language (RSL).
For each key element—such as stakeholders, actors, data entities, data enumerations, state machines,
use cases, user stories, functional requirements, quality requirements, and constraints—provide:
1. A concise definition;
2. Its main purpose in a requirements specification;
3. A small, syntactically valid RSL snippet illustrating its use.
Organize the answer by element. Use the official RSL terminology and syntax, and do not invent
constructs that are not supported by the language.
```

Task6.3. Create **another chat** inside the Language Understanding folder with Prompt T6.2.

```
Prompt T6.2. Understanding the ASL Language
Explain the main elements supported by the Application Specification Language (ASL).
For each key element, provide:
1. A concise definition;
2. Its main purpose in an application specification;
3. A small, syntactically valid ASL snippet illustrating its use.
Conclude with a small but complete and syntactically valid ASL specification for a simple
application. Use the official ASL terminology and syntax, and do not invent unsupported constructs.
```

Task6.4. (Optional) Create a **new chat** and ask your own questions about the RSL or ASL languages.

Task7. Preparing the Chatbot Context

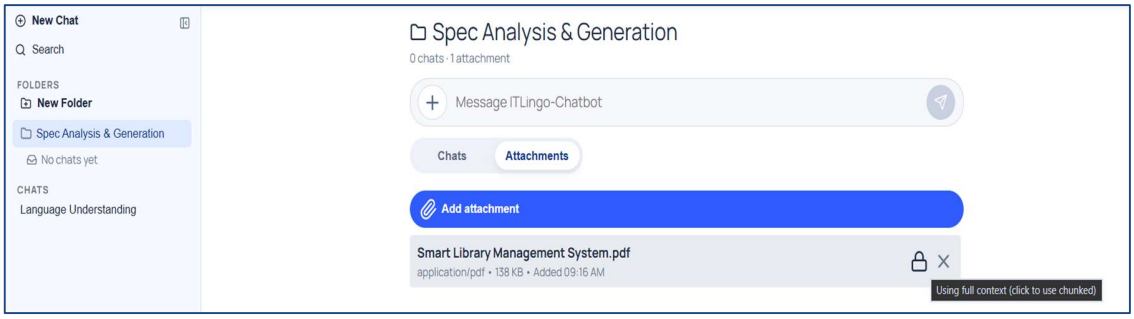
Goal: Prepare the chatbot with the SLMS specification.

Task7.1. Create a new folder inside the chatbot named "RSL Specification".

Task7.2. Click on "RSL Specification" folder then **open Attachments**

Task7.3. Upload the file you downloaded in T4.3 ("SLMS informal specification", whose file name is SLMS-CaseStudy-Definition-v1.1-1.pdf or similar).

Task7.4. Enable Preserve Full Context (**lock icon**), which ensures it is fully included as context in subsequent conversations within the folder.



Chatbot folder interface showing folder-level attachments

Task8. Initial RSL Requirements Specification

Goal: Generate an initial RSL requirements specification containing the high-level requirements elements derived from the natural-language specification.

Task8.1. Create a **New Chat** inside the "**RSL Specification**" folder (Previously created on T7.p.1).

Send Prompt T8.1.

```
Prompt T8.1 – RSL Specification of Requirements
Consider the informal natural-language description of the Smart Library Management System (SLMS)
case study available in the current folder.
Produce a syntactically valid RSL specification containing the following types of elements:
* Package and System;
* Stakeholders;
* Functional requirements;
* Quality requirements;
* Constraints.
Derive every specification element from the provided SLMS description. Preserve the source's
intended meaning and avoid introducing unsupported business rules or assumptions.
Do not include actors, data entities, data enumerations, state machines, use cases, user stories,
scenarios, or any other RSL elements.
Return only the complete RSL specification in a single code block, without explanatory text.
```

Task8.2. Review the generated specification.

Task8.3. If necessary, use the **inline editor** to correct syntax errors or improve the generated specification directly within the chatbot response and **save**.

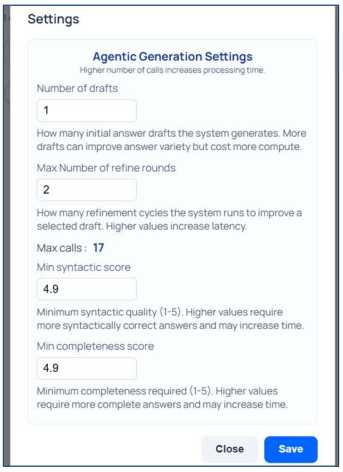
Task9. Complete RSL Domain Specification

Goal: Generate a complete domain-oriented RSL specification using Agentic Generation with controlled quality settings.

Task9.1. Create a **new Chat** inside the "RSL Specification" folder (Previously created in Task 7.1).

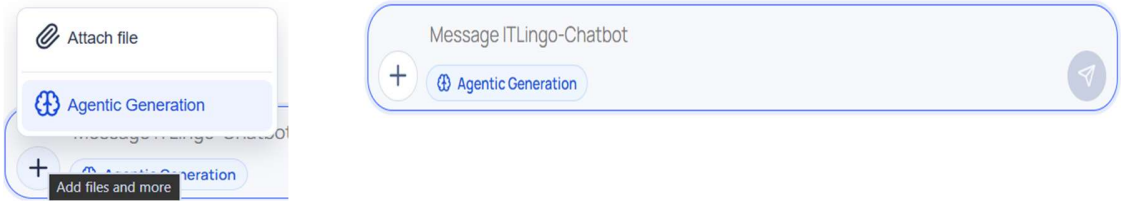
Task9.2. Open Settings in the top right and adjust the values to improve generation, then **save**. **Configure** with the following settings:

- Number of Drafts: 1;
- Max Number of refine rounds: 2;
- Min syntactic score: 4.9;
- Min completeness score: 4.9;



Chatbot Agentic generation quality settings

Task9.3. Inside the Prompt input area, enable Agentic Generation (to ensure that it is activated, there is a tag named "Agentic Generation" at the bottom of the input area).



Chatbot Input bar showing Agentic generation activated

Task9.4. Send **Prompt T9.1** (this task is expected to take longer to generate).

```
Prompt T9.1 – RSL Domain and Use-Case Specification
Consider the informal natural-language description of the Smart Library Management System (SLMS)
case study available in the current folder.
Produce a syntactically valid RSL specification containing the following types of elements:
* Package and System;
* Data entities;
* Data enumerations;
* State machines;
* Actors;
* Use cases (without scenarios).
Represent the main domain concepts, relationships, attribute types, entity lifecycle (Books,
Members, Borrowings, Fines), system actors, and relevant use cases described in the source.
Do not include scenarios within the use cases. Do not include stakeholders, functional requirements,
quality requirements, constraints, user stories, or any other RSL elements.
Derive every element from the SLMS description and avoid introducing unsupported business rules or
concepts.
```

Task9.5. Review the generated specification.

Task9.6. If necessary, use the **inline editor** to correct syntax errors or improve the generated specification directly within the chatbot response and **save**.

Task9.7. Export the final specification to the Workspace Documents. Name it, for example, "SLMS-Chatbot-Generated.rsl".

Task10. Diagrams Generation

Goal: Generate UML-like diagrams from the RSL specification and evaluate the chatbot's ability to visualize system behavior and domain models.

Task10.1. In the **same chat** as the previous task (previously defined on T9.p.1), send **Prompt T10.1** to generate the Use-Case Diagram.

```
Prompt T10.1 – Use-Case Diagram Generated from RSL
Consider the RSL specification produced previously.
Generate a use-case diagram representing all actors and use cases defined in that specification.
Represent actor generalizations and use-case relationships only when they are explicitly defined or
can be directly derived from the RSL specification.
Place the use cases inside a system boundary named SLMS and the human or external-system actors
outside that boundary. Use clear labels, avoid crossed connectors whenever possible, and produce a
readable, well-balanced layout.
Do not introduce actors, use cases, or relationships that are absent from the RSL specification.
```

Task10.2. Review the generated diagram.

Task10.3. If necessary, use the **inline editor** to adjust the generated Mermaid/UML diagram and correct syntax or layout issues (by switching back to preview mode, the changes get automatically saved).

Task10.4. In the **same chat** as the previous step (p.6), send **Prompt T10.2** to generate the Domain Model Diagrams.

```
Prompt T10.2 – Domain-Model Diagrams Generated from RSL
Consider the RSL specification produced previously.
Generate two separate UML class-style diagrams representing all data entities, data enumerations,
and relationships defined in that specification.
The first diagram must show the data entities, data enumerations, and their relationships without
displaying entity attributes. The second diagram must show the same model, including all entity
attributes and their respective data types.
In both diagrams, represent data enumerations with a background color that is visibly distinct from
the one used for data entities. Show association names and multiplicities whenever they are defined
in the RSL specification.
Use consistent names and relationships across both diagrams. Do not introduce elements, attributes,
or relationships that are absent from the source specification.
Return the two diagrams as separate images, clearly identified as **Domain Model without
Attributes** and **Domain Model with Attributes**.
```

Task10.5. Review both generated diagrams. If required, use the **inline editor** to correct modeling, Mermaid syntax, or layout issues before proceeding.

Task11. ASL Specification Generation

Goal: Transform the generated RSL specification into an equivalent ASL specification and generate corresponding domain diagrams.

Task11.1. Download the previously generated file "SLMS-Chatbot-Generated.rsl" (Created on T9.p.4) from the **Workspace Documents**.

Task11.2. Return to the chatbot.

Task11.3. Create a new folder named "ASL Specification".

Task11.4. Repeat the steps described in T7 (p.2–p.4) inside the newly created folder: **upload "SLMS-Chatbot-Generated.rsl" as a folder attachment**, enable Preserve Full Context, and, in the prompt input area, **enable Agentic Generation**.

Task11.5. Send Prompt T11.1.

```
Prompt T11.1 – ASL Specification Derived from RSL
Consider the RSL specification of the Smart Library Management System (SLMS), as attached in the current folder.
Produce an equivalent and syntactically valid ASL specification containing the following types of elements:
* Package and System;
* Data entities;
* Data enumerations;
* Actors;
* User-interface elements, including UI containers, components, and parts.
Preserve the domain entities, enumerations, relationships, and actors defined in the source RSL specification. Define an appropriate user interface for the principal SLMS activities, including navigation and the management of members, books, borrowings, fines, reports, and system settings, whenever supported by the source specification.
Do not include requirements, state machines, use cases, user stories, or other elements outside the requested scope. Do not introduce domain concepts or business rules that are not present in the source RSL specification.
```

Task11.6. Review the generated specification. If necessary, use the **inline editor** to correct syntax errors or improve the generated specification directly within the chatbot response and **save**.

Task11.7. Send **Prompt T11.2** to generate the diagrams.

```
Prompt T11.2 – Domain-Model Diagrams Generated from ASL
Consider the ASL specification produced previously.
Generate two separate UML class-style diagrams representing all data entities, data enumerations, and relationships defined in that specification.
The first diagram must show the data entities, data enumerations, and their relationships without displaying entity attributes. The second diagram must show the same model, including all entity attributes and their respective data types.
In both diagrams, represent data enumerations with a background color that is visibly different from that used for data entities. Show association names and multiplicities whenever they are defined in the ASL specification.
Use consistent names and relationships across both diagrams. Do not introduce elements, attributes, or relationships that are absent from the source specification.
Return the two diagrams as separate images, clearly identified as **Domain Model without Attributes** and **Domain Model with Attributes**.
```

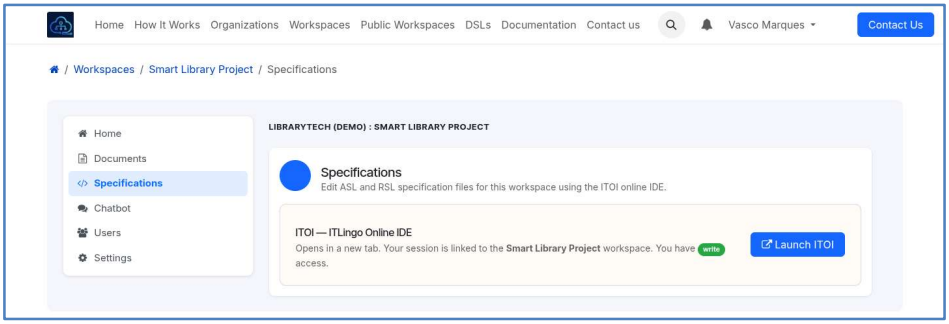
Task11.8. Review both generated diagrams. If required, use the **inline editor** to correct Mermaid syntax, layout, or modeling issues.

Part4. Specifications with ITOI

Task12. Edit and validate the specification in ITOI

Goal: Open the workspace in the ITOI online IDE and experience language-aware editing of RSL, completing the three validation activities (p.5–p.7).

Task12.1. In the workspace sidebar, open **Specifications**.



Specifications page with the Launch ITOI button

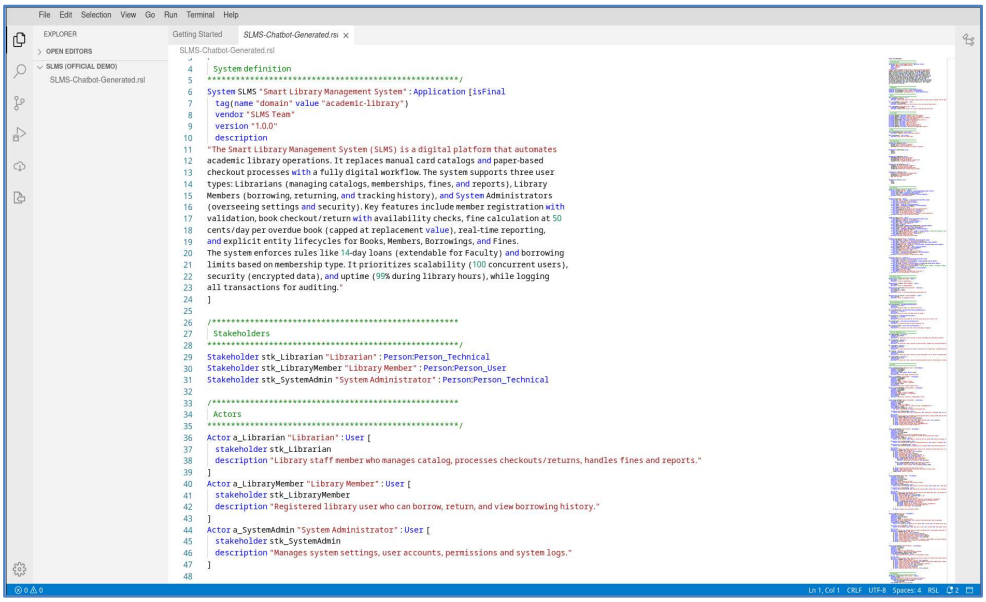
Task12.2. Read the access information: your session will be linked to this workspace, and your badge indicates whether you have **read or write-only** access.

Task12.3. Click **Launch ITOI**. The IDE opens in a new browser tab.

ITOI Main Page

Task12.4. Click on the **Explorer** button. You should see the file you exported in Part 3 Task 9.7. If the file is not present, click on the **Import files** button, click **Import itlingo cloud documents,** and tick `SLMS-Chatbot-Generated.rsl`. The file will appear under projects.

If you previously had trouble generating a correct specification, you can choose to import the Organization-level specification provided for you. Repeat the previous **Import files** step, and tick `SLMS RSL specification (Complete)`. In the next steps, when `SLMS-Chatbot-Generated.rsl` is mentioned, open the last imported file.

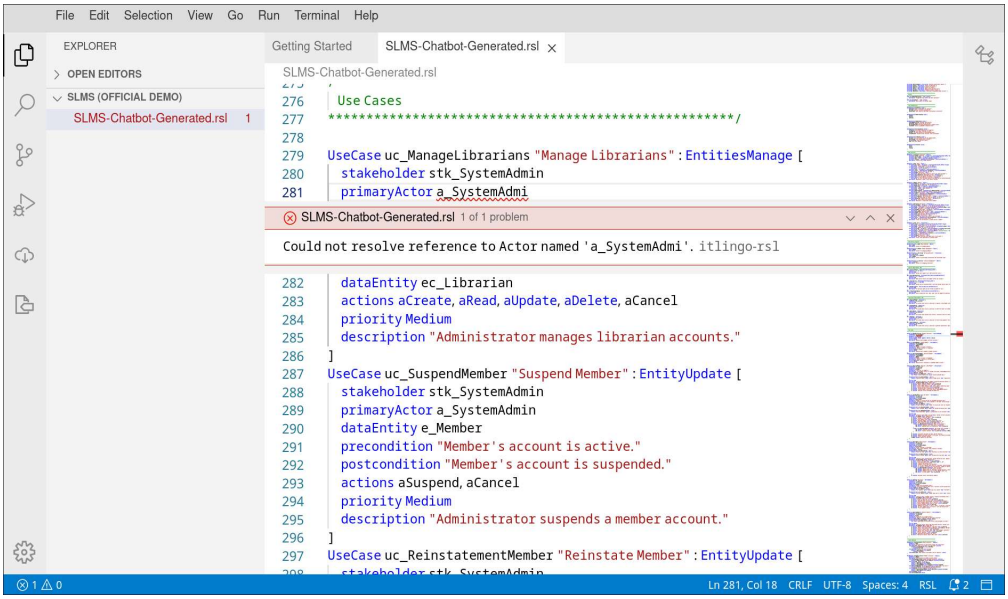


SLMS-Chatbot-Generated.rsl opens in ITOI with RSL syntax highlighting

Task12.5. *Observe the syntax highlighting.* Open `SLMS-Chatbot-Generated.rsl` and observe the **syntax highlighting** of the RSL language. Use the **outline view** to navigate the specification's structure, and hover over an element to view its details.

Task12.6. *Observe the live validation and auto-completion*. Introduce a broken reference and fix it; e.g.:

- In the use case `uc_ManageLibrarians`, change `primaryActor a_SystemAdmin` to `primaryActor a_SystemAdmi`. A red **error marker** appears under the broken reference; hovering it shows *"Could not resolve reference to Actor named 'a_SystemAdmi'"*;
- Delete the wrong name, press `Ctrl+Space` and select `a_SystemAdmin` from the **auto-completion** suggestions. The error disappears.



Live validation error on the broken actor reference

Task12.7. *Add a new RSL element.* Add a new use case at the end of the file, using auto-completion to help you write it, and use `F12` (*Go to Definition*) on a reference such as `de_Book`. For example:

```
UseCase uc_6_ReserveBook "Reserve Book": EntityCreate [
  primaryActor a_LibraryMember
  dataEntity de_Book
  description "Reserve a book that is currently on loan"]
```

Your changes are saved automatically in the ITOI workspace. When you are done, you may close the ITOI tab.

Task12.8. On the left side of the screen, in the **Explorer** tab, right-click the open file and select Download. The file will now be in your **Downloads** folder.

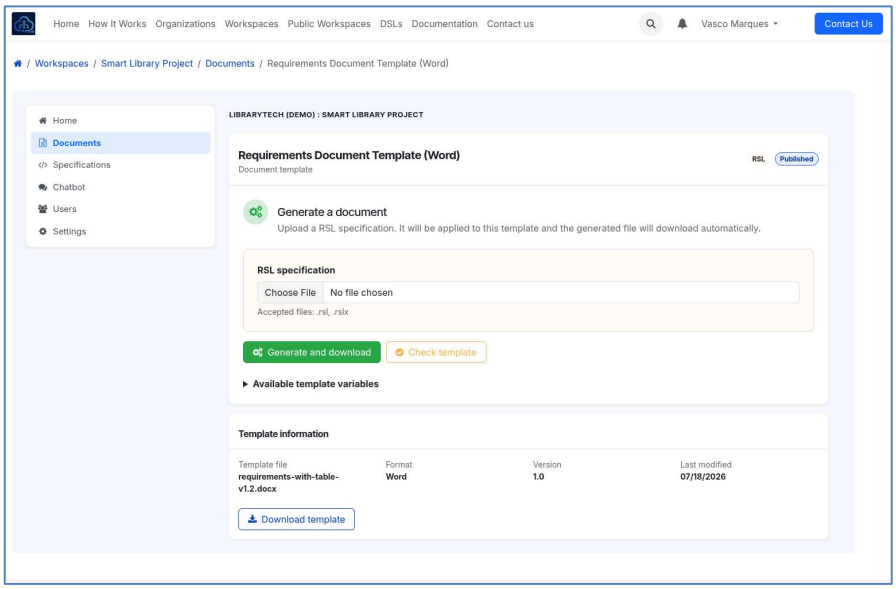
Part5. Document Templating

The templating engine generates human-readable documents (Word, Markdown, JSON, HTML, …) from DSL specifications: a *document template* with placeholders is combined with a *specification file* to produce the final document. The templates are already **published in the organization's Documents**, so you only need to provide your specification.

Task13. Generate documents from the provided templates

Goal: Use the specification you edited in ITOI and generate requirements documents from the organization's ready-made templates, in different formats (Word, Markdown, JSON).

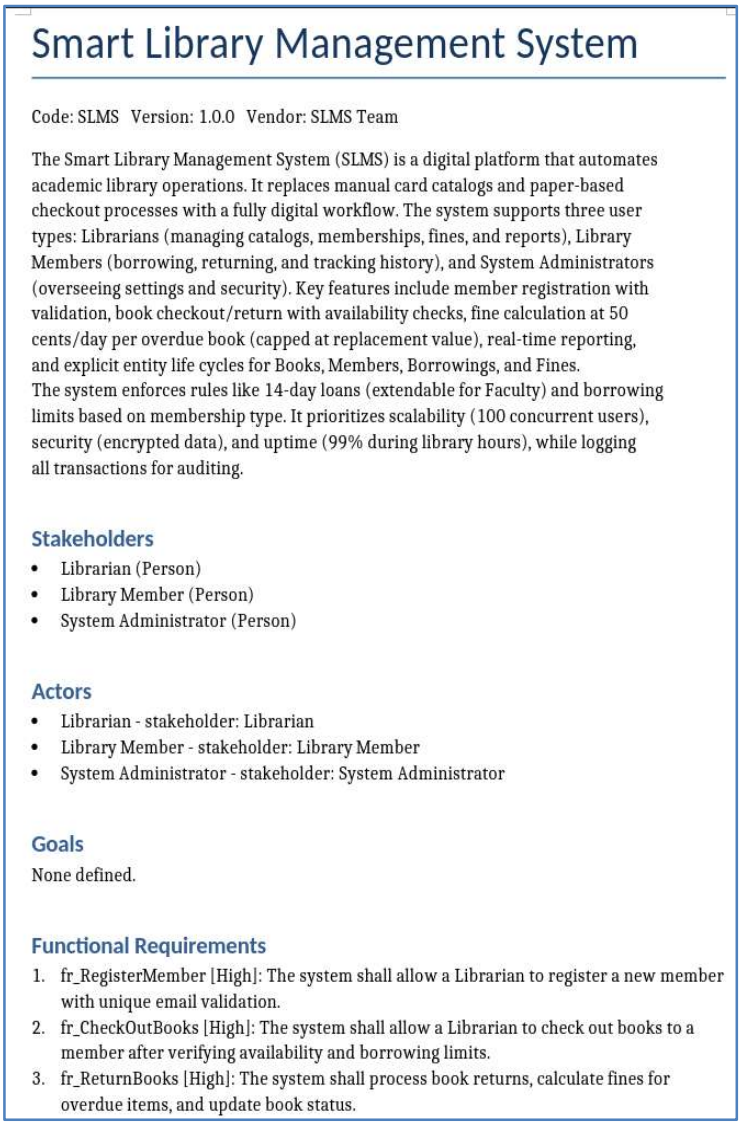
Task13.1. Open a provided template. In your workspace **Documents**, look at the templates published by the organization. Open one of them — because its type is *Document Template*, its detail page already shows the **Generate a document** panel.



Template document detail page with the Generate a document panel

Task13.2. In that panel, **upload** the RSL specification you just downloaded in P4 T12.8 as the source specification, then click **Generate** and download the result.

Task13.3. Open the downloaded document and verify that it contains the actors, use cases, and data entities of the Smart Library Management System — including the changes you made in ITOI.



Smart Library Management System

Code: SLMS Version: 1.0.0 Vendor: SLMS Team

The Smart Library Management System (SLMS) is a digital platform that automates academic library operations. It replaces manual card catalogs and paper-based checkout processes with a fully digital workflow. The system supports three user types: Librarians (managing catalogs, memberships, fines, and reports), Library Members (borrowing, returning, and tracking history), and System Administrators (overseeing settings and security). Key features include member registration with validation, book checkout/return with availability checks, fine calculation at 50 cents/day per overdue book (capped at replacement value), real-time reporting, and explicit entity life cycles for Books, Members, Borrowings, and Fines.
The system enforces rules like 14-day loans (extendable for Faculty) and borrowing limits based on membership type. It prioritizes scalability (100 concurrent users), security (encrypted data), and uptime (99% during library hours), while logging all transactions for auditing.

Stakeholders

- Librarian (Person)
- Library Member (Person)
- System Administrator (Person)

Actors

- Librarian - stakeholder: Librarian
- Library Member - stakeholder: Library Member
- System Administrator - stakeholder: System Administrator

Goals

None defined.

Functional Requirements

1. fr_RegisterMember [High]: The system shall allow a Librarian to register a new member with unique email validation.
2. fr_CheckOutBooks [High]: The system shall allow a Librarian to check out books to a member after verifying availability and borrowing limits.
3. fr_ReturnBooks [High]: The system shall process book returns, calculate fines for overdue items, and update book status.

Word document output generated from the document-template process

Task13.4 (Optional). If you want to explore the templating engine further, repeat the process for the other templates provided to obtain the Word, Markdown, and JSON versions of the requirements document. Notice that the same specification produces different formats — the engine is format-agnostic.

User Questionnaire

You have now completed the guided exploration of ITLingoCloud.

Your feedback is extremely valuable and will help us improve the platform's usability and overall user experience. Please complete the evaluation questionnaire using the link below:

https://forms.gle/9NstqUSuaMynqyA89

The questionnaire follows the same structure as this guide, with sections corresponding to Parts 1–5, followed by a final section covering your overall evaluation of the interface and user experience.

Thank you for your interest, participation, and valuable feedback.