ADA University
School of Information Technologies and Engineering
CSCI 3509: Intro to Software Engineering
Fall 2023

Homework 2: **Software Requirements Specification**

Imba Learn

Project team: Imba gang (Team 8)
Instructor: Kamila Ismayilova

Submitted in partial fulfillment of the requirements of the CSCI 3509: Intro to Software Engineering course project

Version date	Version information
13 Nov 2025	Initial draft
18 Nov 2025	Features revised and models added
20 Nov 2025	Final version is prepared

Team member	Contribution to the homework
Leyla Aliyeva	Worked on introduction, scope and objectives
Diana Kuanyshkyzy	Worked on definitions and target users characteristics
Emin Dabakhov	Worked on assumptions and constraints
Dariya Kairkhanova	Worked on project features, specific requirements, analysis models

Table of Contents

1. Introduction

This is a Software Requirements Specification for project Imba Learn proposed for partial fulfillment of the requirements of the *Introduction to Software Engineering* course in the School of Information Technologies and Engineering at ADA University, Baku, Azerbaijan.

We propose the development of a web-based educational platform inspired by Quizlet, designed to help students and educators with learning through interactive study tools. The platform will allow users to create different modules on different topics. The platform will enable users to create customizable study modules and flashcards on various topics featuring three primary modes: flashcards, tests and quizzes.

Additionally, we are considering the potential to make a mobile application version of this project to complement the web version. Our goal is to create an intuitive, user-friendly tool that optimizes the learning experience and makes it more interesting and convenient.

Scope and Objectives

Potential audience:

Our project targets two primary user categories. The first group are the students who will use the website or application as a learning tool to memorize new terms and assess their knowledge independently. The second category is teachers who may use it as a tool for teaching, to make the lessons more interactive and interesting, and help their students to memorize new terms effectively. While both groups share the common objective of leveraging the platform for learning, their motivations differ: students focus on personal learning, whereas teachers aim to enhance their instructional methods.

Aspects/mechanics that should be covered in the system:

The system must support core learning functionalities that enable users to study, review, and assess terms efficiently. It shall provide multiple learning modes, including flashcards, quizzes, and tests, each offering structured interaction and automated result evaluation. Flashcards must allow sequential or randomized presentation of terms, while quizzes and tests shall generate questions based on the content of a selected module and present users with scores and feedback upon completion.

The system shall include user authentication to allow users to create and manage accounts, create and store personalized modules, and retain their learning results. All mechanics must be implemented through a clear, intuitive interface that supports straightforward navigation and interaction.

Main objectives of the project:

The main objectives of the project are creating a platform (web and mobile) that helps to memorize new terms in different modes, and providing users with the opportunity to personalize their modules and use the platform as the tool to learn or teach effectively.

Major functionalities and features:

The platform should provide a set of core features that support learning, and self-assessment for students and teachers. Key features include interactive flashcards for term memorization, automatically generated quizzes, and structured practice tests that allow users to evaluate their understanding and track their performance.

The platform must ensure reliable operation of these learning modes and support the creation, editing, organization, and storage of user defined modules. Together these features enable users to efficiently study course material, assess progress and use the application as an effective learning tool.

List of features:

1. **User account:** Users must be able to create, edit and manage their account. Register and log in, save changes, create personalized modules and save them. Set and change password and username. View personal statistics and progress. (Optional)
2. **Modules:** Create modules, flashcards; edit and save them; delete modules; organize, filter and search for specific modules.
3. **Flashcards Mode**: Flip flashcards, shuffle and make them appear in random order; track not studied and learned flash cards.
4. **Quiz mode:** Automatically generated quizzes based on the flashcards; Provide scores, time of completion and feedback; different modes of quizzes: multiple choice, true/false, fill-in.
5. **Test mode:** Automatically created practice test that will consist of the majority of terms in the module. The score and the feedback will be provided at the end of the test.
6. **Learning progress (Optional):** Progress of completion of learning the module; history of quizzes and tests; visual progress bar showing the status of each term: not studied, learning, learned.
7. **User interface:** Responsive, convenient design and simple navigation and instructions.

Data collection and Confidentiality

The website will require users to sign up and log in before using it. During the registration, the user must provide their email address to enable authentication, account creation, and password recovery if needed. Also, their names and usernames to personalize their experience and associate their created modules with their accounts. In addition, we will optionally ask users to fill out their personal information as age, gender, and purpose of using the website for statistical analysis. All collected data will be securely stored and used in accordance with privacy and data protection policies.

Definitions

Term	Definition
Flashcard	**A digital card containing information on both sides — one side typically holds a question, term, or prompt, and the other side**

	contains the answer or definition. Used as a learning tool for memorization and self-testing.
Module	**A collection of related flashcards grouped under a specific topic or subject (e.g., “Biology Terms” or “Spanish Vocabulary”). Each module can be created, edited, and shared by users.**
User Account	**A digital identity that allows a person to log in to the system, access their saved modules, and personalize their learning experience. Each user account stores data such as name, email, and progress.**
Authentication	**The process of verifying a user’s identity when logging in, usually through credentials like an email and password. Ensures that only authorized users can access their personal data.**
SaaS (Software as a Service)	**A software distribution model in which applications are hosted by a service provider and made available to users over the internet, rather than installed on local computers.**
Web Application	**A software application that runs in a web browser and can be accessed on any device with an internet connection. It doesn’t require installation like traditional desktop software.**
Mobile Application (App)	**A version of the software designed to run on mobile devices, such as smartphones and tablets. In this case, it refers to the iOS app developed for iPhone users.**
Database	**A structured collection of data that stores user information, flashcards, and modules. It allows the system to efficiently retrieve, modify, and manage data.**
Cloud Hosting	**A hosting method that uses remote servers connected to the internet to store and manage data and applications. It provides scalability, reliability, and accessibility from anywhere.**
API (Application Programming Interface)	**A set of rules and endpoints that allows different software systems (e.g., web and mobile apps) to communicate with the backend server and retrieve or update data.**
Backend	**The part of the application that runs on the server and handles data processing, authentication, and business logic. It communicates with the database and provides information to the frontend.**
Frontend	**The part of the application that users interact with directly — the visual interface built using web technologies like HTML, CSS, and JavaScript.**
Vercel	**A cloud platform used for hosting web applications. It supports modern frameworks like Next.js and provides tools for easy deployment, scaling, and performance optimization.**

Swift	**Swift is a programming language developed by Apple for building iOS applications.**
UIKit	**UIKit is a framework that provides the tools and components for creating the graphical user interface of iOS apps.**

2. Overall Description

Project Features

The project provides an interactive web-based learning platform designed to help users memorize terms and evaluate their understanding through multiple study modes. The system is intended for both students and teachers, enabling them to create learning materials, review content, and measure progress in a structured and intuitive environment.

The main features of the system are described below. Each feature group represents a major functional capability available to users.

List of features:

1. User Account.

This feature allows user to create and maintain their personal accounts. With their accounts users can access other features of our platform such as modules management, learning modes (description of these features are listed below). Account management ensures that each user has a personalized and persistent learning environment.

Includes: registration, login, logout, profile editing, and viewing optional personal statistics.

2. Modules and Flashcards

Modules serve as the main organizational structure for learning content. Each module contains a set of terms where each of them has a definition. Users can create and modify modules based on their study needs.

Includes: creating modules, adding/editing flashcards, deleting content, and searching or organizing modules.

3. Flashcards Learning Mode

Flashcards offer a simple and interactive way to study terms. Users can view cards one by one, flip them to reveal definitions, and reorder them to support memorization strategies.

Includes: viewing, flipping, shuffling the flashcards.

4. Quiz Mode

Quizzes provide a structured method for checking knowledge. The system generates questions automatically from the selected module and evaluates user responses to provide feedback.

Includes: multiple-choice, true/false, and fill-in-the-blank question types, as well as displaying scores and progress.

5. Test Mode

The test mode allows users to assess their understanding of most or all terms in a module. The system presents a more comprehensive evaluation compared to quizzes and provides detailed results at the end.

Includes: automatically generated tests, final scoring, and feedback for incorrect answers.

6. Learning Progress Tracking (Optional Feature)

This feature helps users monitor their study progress across modules. Progress indicators assist users in identifying which terms they have mastered and which need further review.

Includes: progress bars, history of completed quizzes/tests, and term-level learning status.

7. User Interface

The platform provides a clean, intuitive, and responsive interface that supports simple navigation and makes all learning modes easily accessible. The design focuses on clarity and a smooth learning experience.

Target User Characteristics

The target audience is students and teachers on different educational levels.

Students or General Public

- School, college, university level students, or self learners. Typically around 11-25 years old.
- Familiar with basic computer and/or smartphone use;
- Motivated and self-disciplined for the efficient learning process. The platform will not require any specific technical skills and will be free;

Teachers

- Educators on different levels (school, university, tutoring, etc);
- Familiar with basic computer, or optionally smartphone, use;
- Motivated in using online teaching tools.

The users are expected to know the basic level skills of using computers and/or smartphones. Hence, the user interface must be simple to use, responsive, visually appealing and with simple instructions where necessary.

Constraints

Performance and Platform Constraints:

- The web app will be responsive, supporting the latest versions of Chrome, Firefox, and Safari, with page load times under 2 seconds.
- The iOS app will support iOS 16+ with core features like flashcards and modules, aiming for load times under 1 second(Optional).

- Only free development and design tools will be used, including Figma, VS Code, GitHub, and Vercel (Subject to change).
- The backend must support approximately 500 concurrent users.
- The mobile application may not include all features of the web version due to time, resources and scope limitations and thus is optional to implement.

Assumptions and Dependencies

Assumptions: The application targets high school, college students, and lifelong learners familiar with digital learning tools like Quizlet. Users will have reliable internet access. Team members possess the necessary skills for web and mobile development. The backend will support up to 500 concurrent users initially. UI designs will be inspired by Quizlet's layout, created in Figma, ensuring no proprietary assets are reused.

Schedule and Budget: The project will be completed within a 4-month semester timeline (September–December 2025). No budget is required, free tools and platforms will be used (Figma free plan, Vercel hosting, Xcode, VS Code). Minor timeline adjustments may occur, but all work is expected to be completed by the end of the semester.

Resources and Software: Development will use personal computers and GitHub for version control. Reusable software includes Figma templates and Quizlet-inspired UI/UX (no proprietary assets). All APIs, libraries, and frameworks (React.js, Next.js, Node.js/NestJS, Swift/UIKit) are open-source or freely available. For the backend and frontend the the boilerplate might be used for improve the development process.

Techniques and Interfaces: REST APIs with JWT authentication will enable secure communication between frontend and backend. The shared backend will use PostgreSQL with a normalized schema for data consistency across web and mobile apps.

Intended Use: The application will primarily serve as an educational study tool, allowing users to create, study, and share learning modules with flashcards. It will support language learning, term revision, and preparation for school or university assignments requiring memorization and review.

Hardware required: The only hardware requirement is a device capable of running a modern web browser (e.g., Google Chrome, Safari, or Firefox) and, for mobile access, an iPhone compatible with the mobile app. The internet access is required as well.

Target platforms:

- Web application (main): All features of the system will be available here. Accessible on desktop and mobile browsers.
- Mobile application (optional): iOS (built with Swift/UIKit), including most core features; some functionality may be limited compared to the web version.

3. Specific Requirements

Functional Requirements

1. **User account:**

FR-1.1: The system shall allow users to register with their emails and passwords

FR-1.2: The system shall allow users to log in using their registered email and password.

FR-1.3: The system should allow users to log out.

FR-1.4: The system shall allow users to edit their profile information, including username and avatar.

FR-1.5(Optional): The system shall allow users to change the password if the user is logged in.

FR-1.6(Optional): The system may allow users to reset their password via email if user is not logged in.

2. **Modules/flashcards:**

FR-2.1: The system shall let users create modules and delete modules

FR-2.2: The system shall allow users to add, edit, and delete flashcards within modules

FR-2.3: The system shall allow users to search for modules by name

FR-2.4: The system shall allow users to mark favorite or important flashcards for focused study.

FR-2.5: The system shall show the progress of the module based on the flashcards learning progress

FR-2.6: The system shall allow to sort the flashcards within the module by creation date and name.

FR-2.7: The system shall mark flashcards as not started, in progress or completed based on quiz history.

3. **Flashcard mode:**

FR-3.1: The system shall allow users to view flashcards one by one.

FR-3.2: The system shall allow users to flip a flashcard to reveal the reverse side

FR-3.3: The system shall allow flashcards to be shuffled upon user request

4. **Quiz mode:**

FR-4.1: The system shall automatically generate quizzes based on flashcards of the modules

FR-4.2: The questions should be of different types: multiple choice, true/false, fill in the gap, etc

FR-4.3: The system shall provide feedback indicating whether each answer was correct or incorrect.

FR-4.4: The system shall store quiz results for the user.

FR-4.5: The system shall calculate results, save them in the module, and provide feedback

5. **Test mode:**

FR-5.1: The system shall generate tests containing most or all terms from a selected module.

FR-5.2: The system shall automatically grade a completed test.

FR-5.3: The system shall include multiple choice questions in tests

FR-5.4: The system shall include most/all terms from the module in the generated test

FR-5.5: The system should give an overall result at the end of the test with the score, number of mistakes and the feedback.

6. **Mobile application:**

FR-6.1: The mobile application shall include the majority of features available on the web application

FR-6.2: The system shall synchronize data between the web application and mobile app in near real-time.

Nonfunctional Requirements

7. **Usability:**

NFR-7.1: User interface shall be simple and intuitive.

NFR-7.2: Web design shall be responsive.

8. **Performance:**

NFR-8.1: Web page should load in less than 2 seconds

NFR-8.2: Mobile application should load in less than 1 second

NFR-8.3: The system shall handle at least 500 concurrent users without performance degradation.

9. **Security:**

NFR-9.1: Authentication shall use JWT

NFR-9.2: Users' personal data shall remain confidential

10. **Compatibility:**

NFR-10.1: Website shall support Chrome, Safari, Firefox

NFR-10.2: Mobile application shall support IOS 16+

4. Analysis Models (Use Cases)

External Actor Descriptions

Human actors:
Learner

- The main users that register, login, create modules, and learn them by different modes like flashcards, quizzes, tests.

Computer actors:
Authentication system

- The system that validates user's credentials and manages authentication process

Database management system (DBMS)

- The system that stores user accounts, flashcards, study sets, statistics, progress, etc. And handles retrieval and update of all data.

Use Case Descriptions

No	User story name	Description
1	**Learner registration**	Users sign up as learners by providing their personal contact information and unique login credentials.
2	**Login**	After registration provided data will be stored in the database, which will be used for logging into the website
3	**Module creation**	Learners create their own module by specifying the name, short description and adding words to be learned in the form of flashcards
4	**Account editing**	Learners edit their personal accounts by changing the username, password, or personal information. The changes must be saved.
5	**Flashcards studying**	Learners study the flashcards by flipping them and memorizing definitions or translations. Flashcards appear in random order and the progress is tracked by the system.
6	**Taking quiz**	Learners take quizzes that are automatically generated based on the flashcards in the module. The questions are such as multiple choice, matching terms, filling in the gap, etc. Results and time of completion are tracked and the feedback is given after completion. Results added to the history of quizzes.
7	**Taking test**	Learners take tests that are automatically generated based on the flashcards in the module. THe questions are multiple choice. The results and time of completion are tracked and the feedback is provided. Results added to the history of tests.

Use Case Diagram

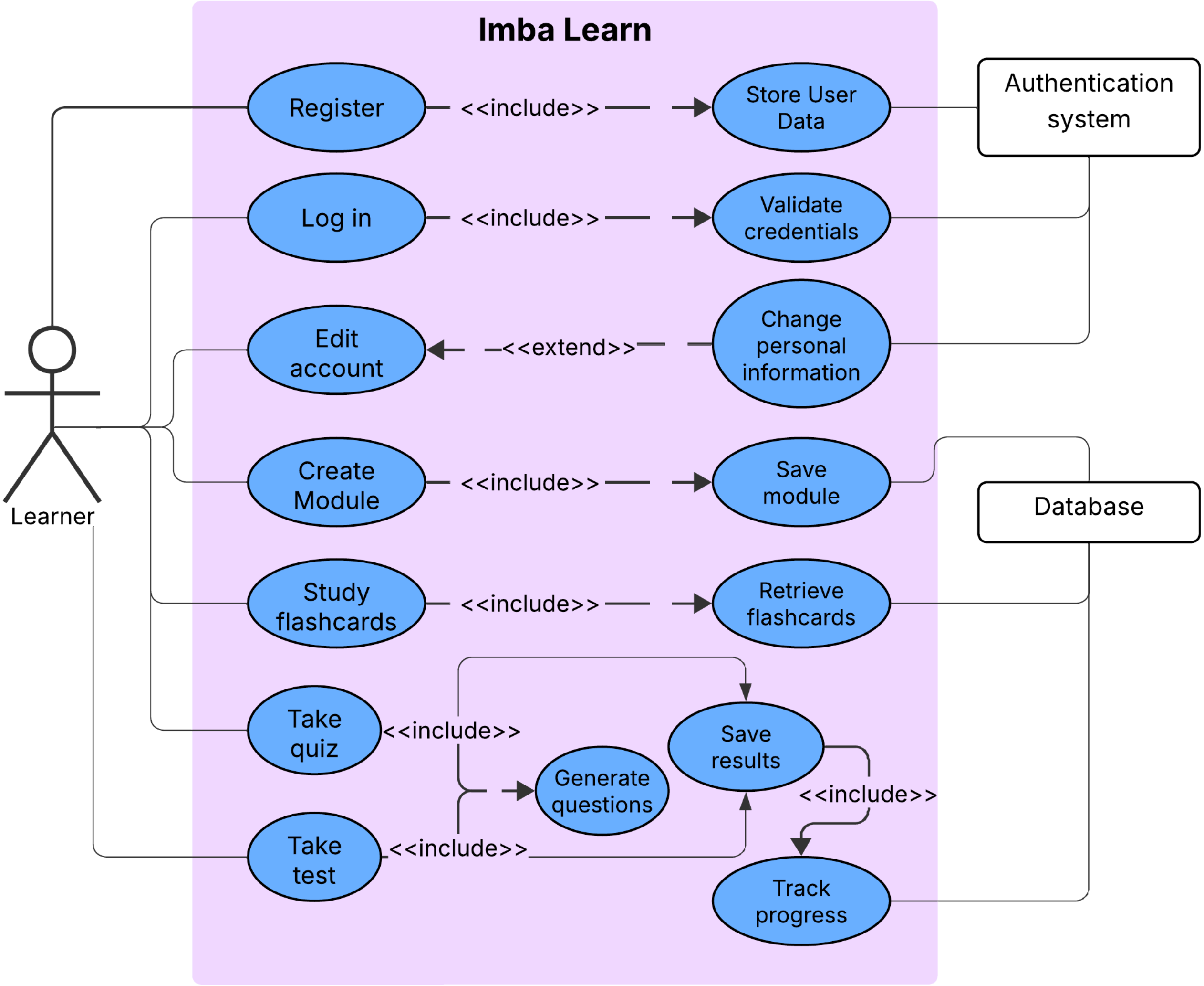

Figure 2. Use case diagram for Imba Learn project

Use Case 1

Use Case Number:	UC-01
Use Case Name:	Registration
Actor(s):	Registrant
Description:	Registrant shall provide personal and contact information to the System upon registering and becoming a User.
Trigger:	External - Registrant registering as a User.
Pre-condition(s):	- System has been setup and configured. - System is running and open for registrations. - Registrant has accessed website via URL
Scenario Flow:	Main (success) Flow: 1. Registrant selects option to register 2. System requests personal information 3. Registrant provided personal information. 4. System verifies required information is provided. ● If information is invalid System displays message. Return to Step 2 5. System requests login information 6. Registrant provides login information 7. System verifies required information is provided ○ If information is invalid System displays message. Return to Step 5 15. System displays confirmation of registration
Alternate Flows:	Alternate Flows: Alternate Flow #1: After Step 2 in success scenario System will display the option to Cancel the registration process. The following steps would occur: 1. Registrant selects option to cancel during registration 2. System requests confirmation to cancel 3. Registrant confirms intent 4. System returns to main screen
Post Condition:	Registrant completed registration. System stored Registrant's information.

Use Case 2

Use Case Number:	UC-02
Use Case Name:	Login
Actor(s):	Registered learners
Description:	Used log into the website using the usernames and passwords provided at the registration step
Trigger:	External - Learner logs in to the system as a user
Pre-condition(s):	- System has been setup and configured - System is running and open for logging in - Learner has accessed the website via URL
Scenario Flow:	Main (success) Flow: 1. 1.Learner selects option to login 2. System requests username and password 3. Learner provides their username and password 4. System verifies required information is provided ● If information is invalid System displays a message. Return to Step 2 5. Learner successfully logs in
Alternate Flows:	Alternate Flows:

	Alternate Flow #1: After Step 1 in success scenario System will display the option to register if user has not done that yet. The following steps would occur: 1. Learner chooses the option to register 2. Learner undergoes UC-01 scenario for registration Alternate Flow #2 (optional) : After Step 4 in unsuccess scenario, System will display the message asking if the user forgot their password. The following steps would occur: 1. Learner chooses the option “I forgot the password” 2. Learner will verify his identity by email and then change password 3. Password shall be updated in the database
Post Condition:	Learner successfully logs in to the website

Use Case 3

Use Case Number:	UC-03
Use Case Name:	Module creation
Actor(s):	Learner
Description:	Learners create their own module by specifying the name, short description and adding words to be learned in the form of flashcards
Trigger:	External - learner creates the module
Pre-condition(s):	- System has been setup and configured - System is running - Learner has accessed the website via URL - Module creation page is open
Scenario Flow:	Main (success) Flow: 1. Learner selects option to create a module 2. Learner enters the name and short description of the module 3. Learner enters flashcards (terms with definitions) (not required) 4. System validates the input data ● If there is empty input then the system displays a message. Return to Step 2 5. Data is saved in the database 6. Module is created
Alternate Flows:	Alternate Flows: Alternate Flow #1: After Step 1 in success scenario System will display the option to Cancel the module creation. The following steps would occur: 1. Learner selects option to cancel during creating module 2. System requests confirmation to cancel 3. Learner confirms intent 4. System returns to main page
Post Condition:	Learner successfully created the module and is redirected to the page with created module

Sequence diagram for creating a module:

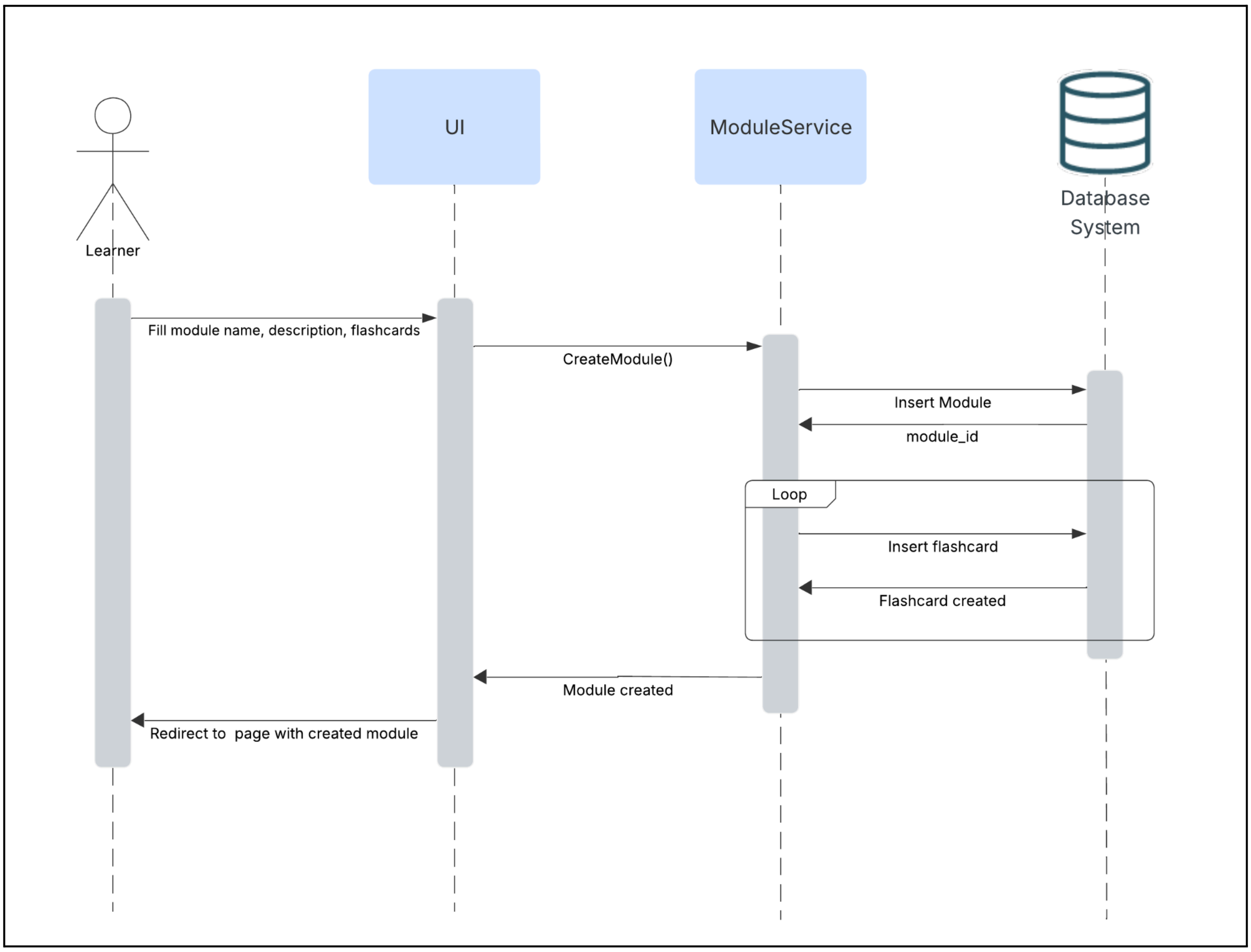

Figure 1. Sequence diagram for module creation

Use Case 4

Use Case Number:	UC-04
Use Case Name:	**Account editing**
Actor(s):	**Learner**
Description:	**Learners edit their personal accounts by changing the username, avatar, password, or any other personal information**
Trigger:	**External - user edits the profile information**
Pre-condition(s):	- **System has been setup and configured** - **System is running** - **Learner has accessed the website via URL** - **Learner logged in to the account and pressed the edit button**
Scenario Flow:	**Main (success) Flow:** 1. **Learner selects option to edit account** 2. **Learner is redirected to the personal profile page** 3. **Learner edits the username, password, and/or other personal information** ● **If the learner inputs invalid data then the system displays a message. Return to Step 3** 4. **Learner confirms the changes and saves them**
Alternate Flows:	**Alternate Flows:**

	Alternate Flow #1: After Step 2 in success scenario System will display the option to Cancel account editing. The following steps would occur: 1. Learner selects option to cancel during editing account 2. System requests confirmation to cancel 3. Learner confirms intent 4. System returns to main page
Post Condition:	Personal information is changed and changes are saved in the database

Use Case 5

Use Case Number:	UC-05
Use Case Name:	Flashcards studying
Actor(s):	Learner
Description:	Learners study the flashcards by flipping and memorizing them. Flashcards appear in random order and the progress is tracked by the system.
Trigger:	External - user learns flashcards
Pre-condition(s):	- System has been setup and configured - System is running - Learner has accessed the website via URL - Learner has an account - Learner created module and flashcards
Scenario Flow:	Main (success) Flow: 1. Learner selects option to study flashcards 2. System redirects leaner to the flashcards page 3. Learner studies flashcards by flipping them and going to the next flashcards. 4. Learner completes studying and exits studying mode 5. System redirects learner to the main page
Alternate Flows:	Alternate Flows: Alternate Flow #1: After Step 2 in success scenario System will display the option to Cancel flashcards studying mode. The following steps would occur: 1. Learner selects option to cancel flashcards studying mode 2. System requests confirmation to cancel 3. Learner confirms intent 4. System returns to main page Alternate Flow #2: After Step 3 in the success scenario, the learner may want to shuffle the order of the flashcards. The following steps would occur: 1. Learner selects option to shuffle the flashcards 2. Order of flashcards changes randomly 3. Proceed to the Step 3
Post Condition:	Learner learns terms using flashcards mode

Use Case 6

Use Case Number:	UC-06
Use Case Name:	Taking quiz
Actor(s):	Learner
Description:	Learners take quizzes that are automatically generated based on the module's terms. The questions are of different types. Results are tracked.
Trigger:	External - user takes the quiz after studying flashcards
Pre-condition(s):	- System has been setup and configured

	- System is running - Learner has accessed the website via URL - Learner has an account - Learner created module and flashcards - Learner studied the flashcards
Scenario Flow:	**Main (success) Flow:** 1. Learner selects option to take a quiz 2. System automatically generates quiz based on flashcards 3. Learner completes tasks one by one 4. System tracks correct and incorrect answers 5. Learner completes all tasks and submits 6. Learning progress data updated based on the correct and incorrect answers
Alternate Flows:	**Alternate Flows:** **Alternate Flow: After Step 2 in the success scenario System will display the option to Exit from the quiz. The following steps would occur:** 1. Learner selects option to exit the quiz 2. System requests confirmation to exit 3. Learner confirms intent 4. System ends the quiz and redirects learner to the main page
Post Condition:	**Quiz result is saved and feedback is provided**

Use Case 7

Use Case Number:	UC-07
Use Case Name:	Taking test
Actor(s):	Learner
Description:	Learners take tests that are automatically generated based on the module. The questions are of MCQ, true/false and fill-in type. The result is shown with the feedback
Trigger:	External - user takes the test after studying flashcards
Pre-condition(s):	- System has been setup and configured - System is running - Learner has accessed the website via URL - Learner has an account - Learner created module and flashcards - Learner studied the flashcards
Scenario Flow:	Main (success) Flow: 1. Learner selects option to take a test 2. System automatically generates test based on terms 3. Learner completes questions one by one 4. System tracks correct and incorrect answers 5. Learner completes all questions and submits 6. System shows the result, feedback including mistakes and correct answers
Alternate Flows:	Alternate Flows: Alternate Flow #1: After Step 2 in the success scenario System will display the option to Exit from the test. The following steps would occur: 1. Learner selects option to exit the test 2. System requests confirmation to exit 3. Learner confirms intent 4. System ends the test and redirects learner to the main page
Post Condition:	Test result and feedback is provided

5. References

Quizlet. *Quizlet – Learning Tools & Flashcards.* https://quizlet.com/gb

Vercel: https://vercel.com/