Monen

Software Architecture Document

Version 1.0

Revision History

| **Date** | **Version** | **Description** | **Author** |
| --- | --- | --- | --- |
| 29/11/2023 | 1.0 | Initial edit | Quang |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[**1. Introduction 4**](#_heading=h.gjdgxs)

[**2. Architectural Goals and Constraints 4**](#_heading=h.30j0zll)

[**3. Use-Case Model 5**](#_heading=h.1fob9te)

[**4. Logical View 6**](#_heading=h.3znysh7)

[4.1 Component: User Interface 6](#_heading=h.2et92p0)

[4.2 Component: Learning Engine 7](#_heading=)

[4.3 Component: User Management and Authentication 8](#_heading=h.ms498ru1bag4)

[4.4 Component: Course 9](#_heading=h.u14f1kwz6p38)

[**5. Deployment 10**](#_heading=h.tyjcwt)

[**6. Implementation View 10**](#_heading=h.3dy6vkm)

Software Architecture Document

# Introduction

# Architectural Goals and Constraints

**Platform Compatibility:**

- The app must be compatible with iOS (version 14 and above) and Android (version 8 and above) platforms.

**Performance:**

- The app should load within 3 seconds on standard mobile devices.

- Each flashcard loading time should be less than 1 second to ensure a seamless user experience.

- Response time for the user interaction should be less than 0.5 seconds.

**Scalability:**

- The app should handle a minimum of 30,000 concurrent users without performance degradation.

- The backend server should be scalable to accommodate an increasing number of users and flashcard sets.

**Security:**

- User data must be encrypted and securely stored.

- The app must comply with industry-standard data protection regulations and user privacy standards.

**Usability:**

- The app interface should be intuitive, ensuring ease of use for all users.

- The app must provide clear instructions and guidance for new users.

**Reliability and Availability:**

- The flashcard deck library server should have a minimum uptime of 99.9% to ensure accessibility at all times.

- Regular maintenance and updates should be performed during non-peak hours to minimize user disruption.

- The app should be available even when there is no internet connection.

**Documentation:**

- Provide an online user manual accessible within the app, detailing features, navigation, and troubleshooting steps.

- Include in-app tooltips and guides to assist users in understanding app functionalities.

**Design Constraints:**

- The app design should be responsive and optimized for various screen sizes, including smartphones and tablets.

- Utilize lightweight graphics to minimize app size and reduce data usage.

**Dependency:**

- Integrate the app with popular messaging and social media platforms for easy sharing and collaboration among users.

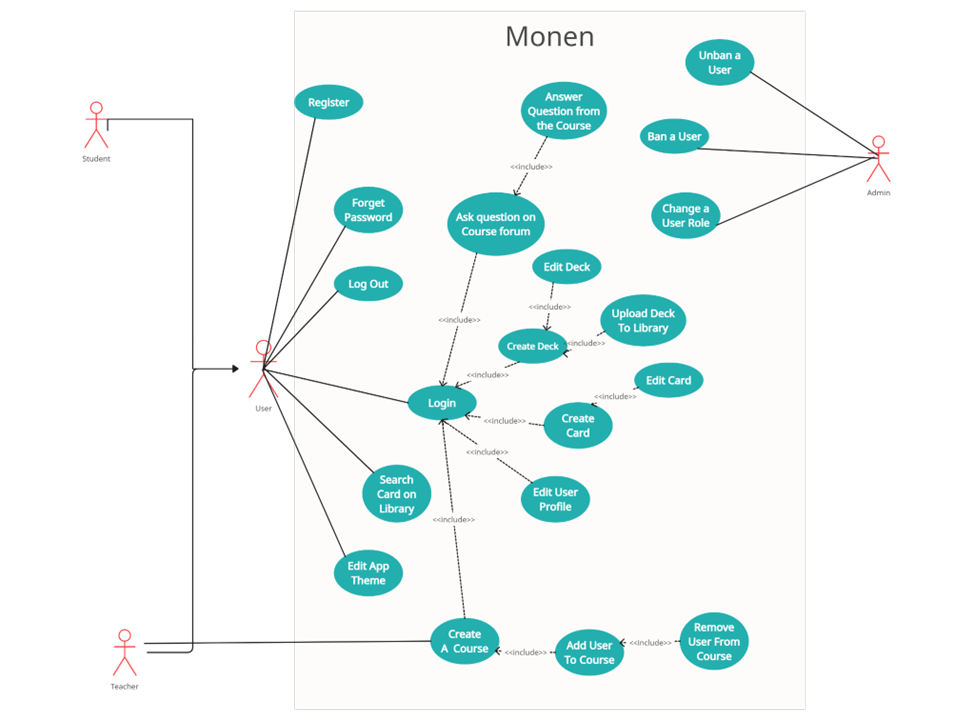
- Collaborate with cloud service providers to ensure reliable storage and backup of user data.

**Documentation and Packaging:**

- Provide clear installation instructions for both iOS and Android users.

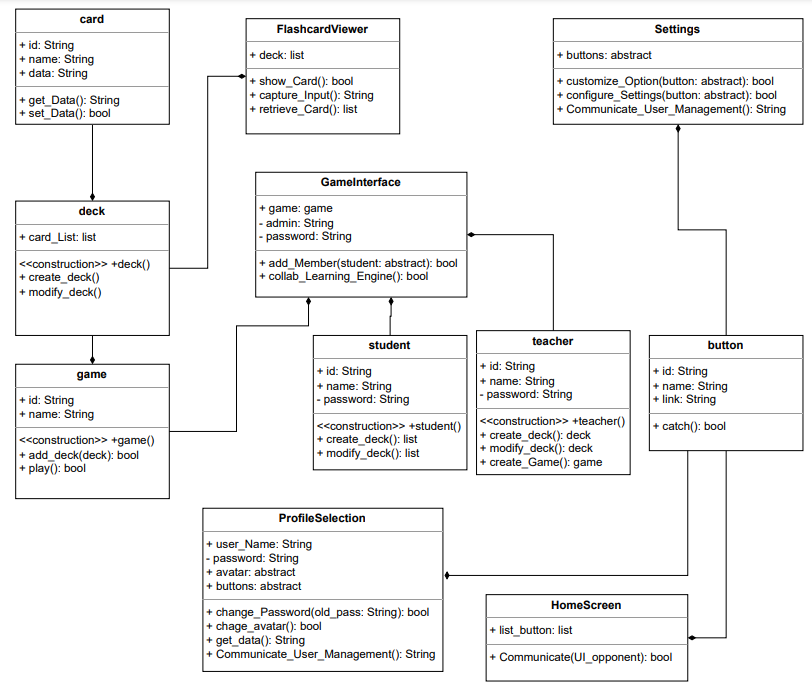
- Include a comprehensive user guide explaining app features and best practices.

# Use-Case Model

**

# Logical View

## Component: User Interface

**

**HomeScreen**:

* Responsibilities:
  + Displaying the main screen with access to different features.
  + Handling navigation to other sections of the app.
* Relationships:
  + Communicates with other UI components for feature access.

**FlashcardViewer:**

* Responsibilities:
  + Presenting flashcards with questions and answers.
  + Capturing user input for flashcard interactions.
* Relationships:
  + Interacts with the Learning Engine to retrieve flashcards.

**GameInterface:**

* Responsibilities:
  + Hosting interactive learning games.
  + Managing game sessions and user progress.
* Relationships:
  + Collaborates with the Learning Engine for game content.

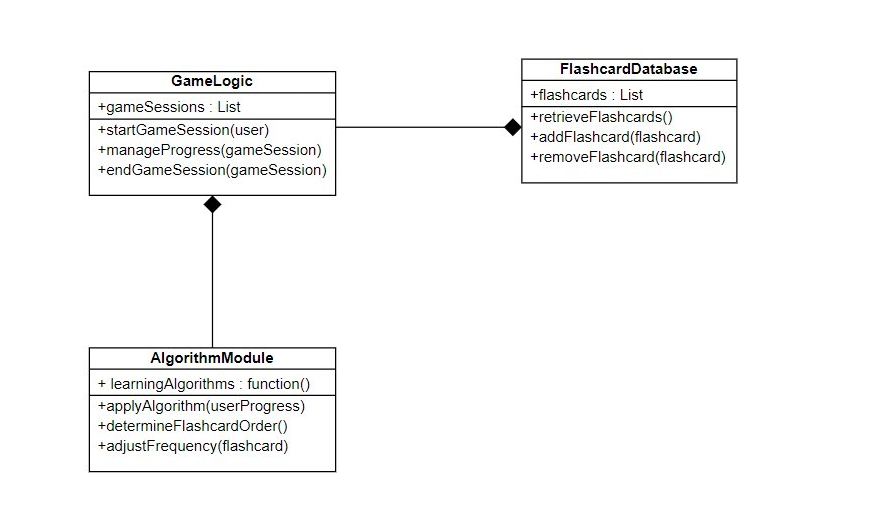
**ProfileSection:**

* Responsibilities:
  + Displaying user profiles and progress.
  + Allowing users to manage account settings.
* Relationships:
  + Communicates with User Management for profile data.

**Settings:**

* Responsibilities:
  + Providing customization options and preferences.
  + Allowing users to configure app settings.
* Relationships:
  + Interacts with User Management for user preferences.

## Component: Learning Engine

**

**FlashcardDatabase:**

* Responsibilities:
  + Storing and managing a diverse set of flashcards.
  + Facilitating retrieval of flashcards for display.
* Relationships:
  + Accessed by FlashcardViewer and AlgorithmModule.

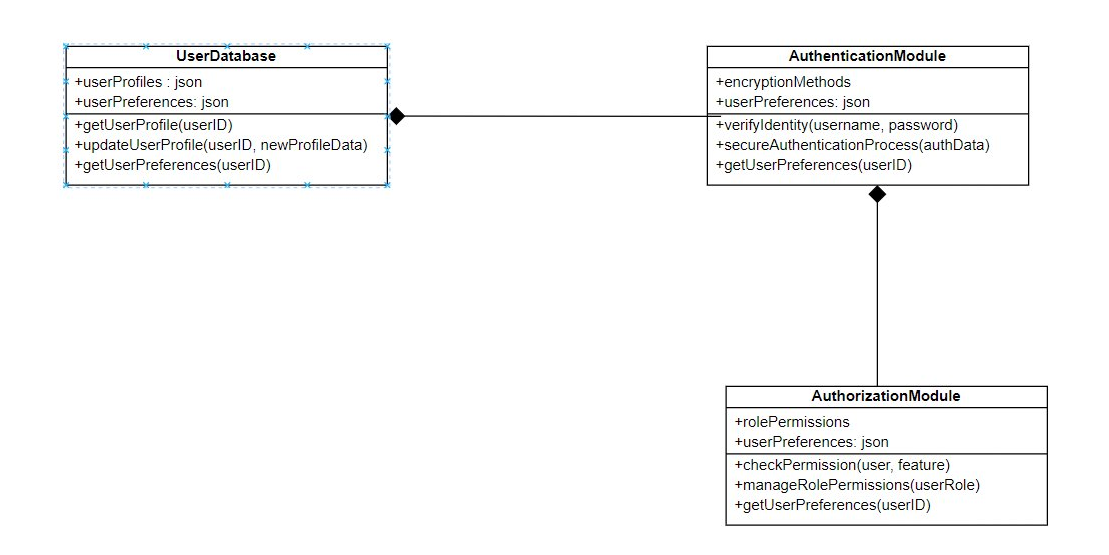
**AlgorithmModule:**

* Responsibilities:
  + Applying learning algorithms to adapt to user progress.
  + Determining the order and frequency of flashcards.
* Relationships:
  + Interacts with FlashcardDatabase and GameLogic.

**GameLogic:**

* Responsibilities:
  + Implementing interactive games aligned with learning objectives.
  + Managing game sessions and user achievements.
* Relationships:
  + Collaborates with FlashcardDatabase for game content.

## Component: User Management and Authentication



**UserDatabase:**

* Responsibilities:
  + Storing user profiles, progress, and preferences.
  + Providing data for user-related functionalities.
* Relationships:
  + Accessed by ProfileSection and Settings.

**AuthenticationModule**:

* Responsibilities:
  + Verifying user identity during login.
  + Ensuring secure authentication processes.
* Relationships:
  + Communicates with UserDatabase for user verification.

**AuthorizationModule:**

* Responsibilities:
  + Controlling access to specific features based on user roles.
  + Managing permissions for various app functionalities.
* Relationships:
  + Interacts with other components to enforce access control.

## 

## Component: Course

## 

**CourseManagementModule:**

* Responsibilities:
  + Facilitating the creation and management of courses.
  + Handling the enrollment of students in a course.
* Relationships:
  + Interacts with UserDatabase for user information.
  + Manages FlashcardDeck instances.

**FlashcardDeck:**

* Responsibilities:
  + Representing a set of flashcards created by a teacher.
  + Storing questions and answers for a specific course.
* Relationships:
  + Belongs to a CourseManagementModule.
  + Accessed by students enrolled in the course.

**ForumModule:**

* Responsibilities:
  + Managing a forum associated with each course.
  + Allowing students to post questions and teachers to provide answers.
* Relationships:
  + Connected to CourseManagementModule for course association.
  + Contains instances of ForumPosts.

**Student:**

* Responsibilities:
  + Representing a student enrolled in a course.
  + Accessing and interacting with flashcard decks.
* Relationships:
  + Enrolled in courses managed by CourseManagementModule.

**Teacher:**

* Responsibilities:
  + Representing a teacher who creates and manages courses.
  + Responding to questions posted in the forum.
* Relationships:
  + Creates and manages courses in CourseManagementModule.
  + Participates in the ForumModule as an authoritative figure.

# Deployment

*[Leave this section blank for PA3 if you are writing this document for PA4.*

*In this section, describe how the system is deployed by mapping the components in Section 4 to machines running them. For example, your mobile app is running on a mobile device (Android, iOS, etc), your server runs all components on the server side including the database]*

# Implementation View

*[Leave this section blank for PA3 if you are writing this document for PA4.*

*In this section, provide folder structures for your code for all components described in Section 4. ]*