



Glossa

Learning Languages Made Easy.

Samuel Jacobs
Matthew Cundiff
Hamza Daruger
Aryan Patel
Ashton LaRoche
Kelly Holl
Brandon Buchanan



Glossa

Learning Languages Made Easy.

Objective

- Create a flashcard-based tool to help learn languages!
- To stand out from other flashcard apps, Glossa offers speaking/listening study sessions in addition to reading/writing study sessions.
- Tailor study sessions to focus on flashcards that the user frequently gets incorrect and flashcards that the user rarely studies.



Glossa

Learning Languages Made Easy.

Cost Estimation

Hardware:

- HostGator hosting plan including a web server, databases, and a domain name: \$3.50 per month
- External hard drive to use as a backup: \$150
- Development laptops: \$1,000 per developer

Software:

- Visual Studio Code text editor: freeware
- WinSCP FTP client: freeware
- GitHub Team: \$44 per user per year

Personnel:

- 4 entry-level developers: \$22,500 per month per developer
- 3-4 months for development
- 1-2 months for training and testing



Glossa

Learning Languages Made Easy.

Timeline

We estimate that our project will take about 6 months to complete. Development will take 3-4 months for a team of 4 developers based on the estimation in 3.2, and the remaining 2-3 months will be used for training, testing, and buffer time. Our team will work a standard schedule of 8 hours per business day (excluding weekends). The start date will be January 9th, 2023, and the end date will be July 7th, 2023.



Functional Requirements

1. The system shall randomize the order of the flashcards
2. A user shall be able to use their flashcards to play matching games
3. The system shall save user progress after the user studies a card
4. A user shall be able to add and remove flashcards from a deck
5. A user shall be able to add and remove decks from a library
6. A user shall be able to view a deck of flashcards
7. A user shall be able to tag decks with languages



Non Functional Requirements

Performance: The system shall randomize card sets in 0.2 seconds.

Space: The application shall take up no more than 1 GB of device storage.

Usability: A user shall be able to navigate to a flashcard deck within 5 seconds of opening the app.

Dependability: The system shall be available 24/7. Downtime within a given day shall not be more than 1 minute.

Security: Users shall authenticate themselves using a username and password.

Environmental: The app shall be able to work in iOS and Android operating systems.

Operational: The screen refresh time shall not exceed 2 seconds.

Development: The system shall remain functional during update deployments.

Regulatory: The app shall fall within FCC requirements and Android/Apple App Store requirements.

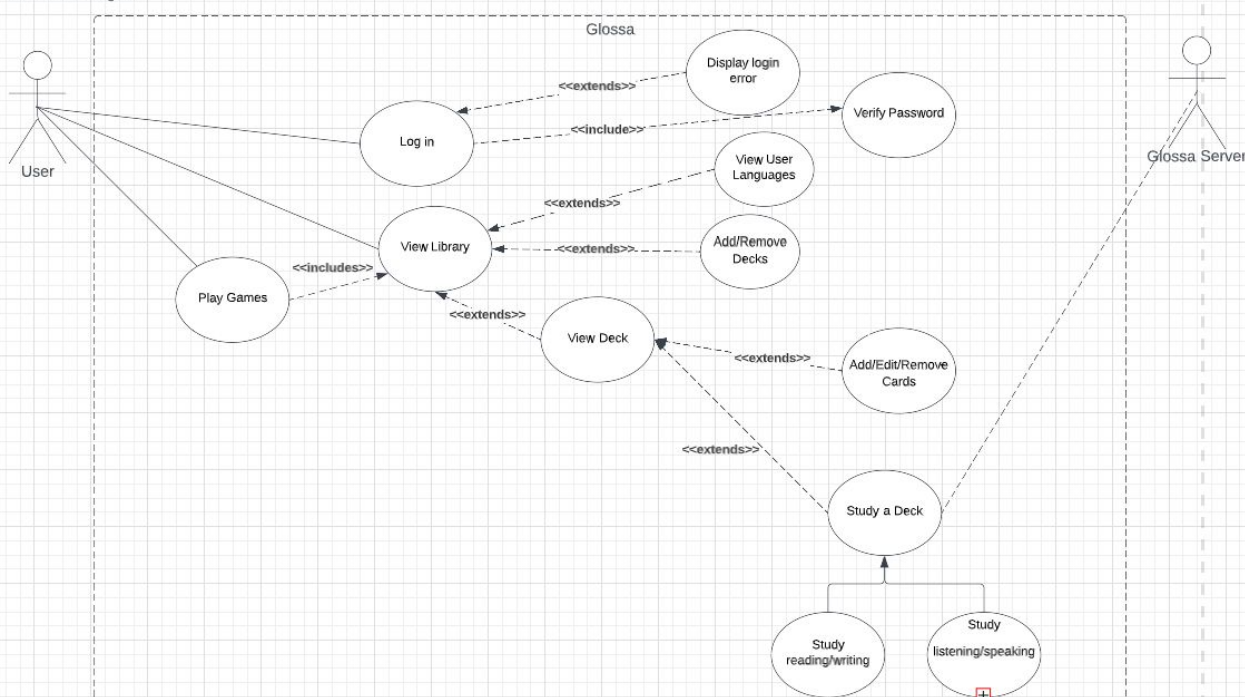
Ethical: User data shall not be sold.

Accounting: Payment plan shall complete in 2 minutes.

Safety/Security: Passwords shall be encrypted so that user accounts are secure.

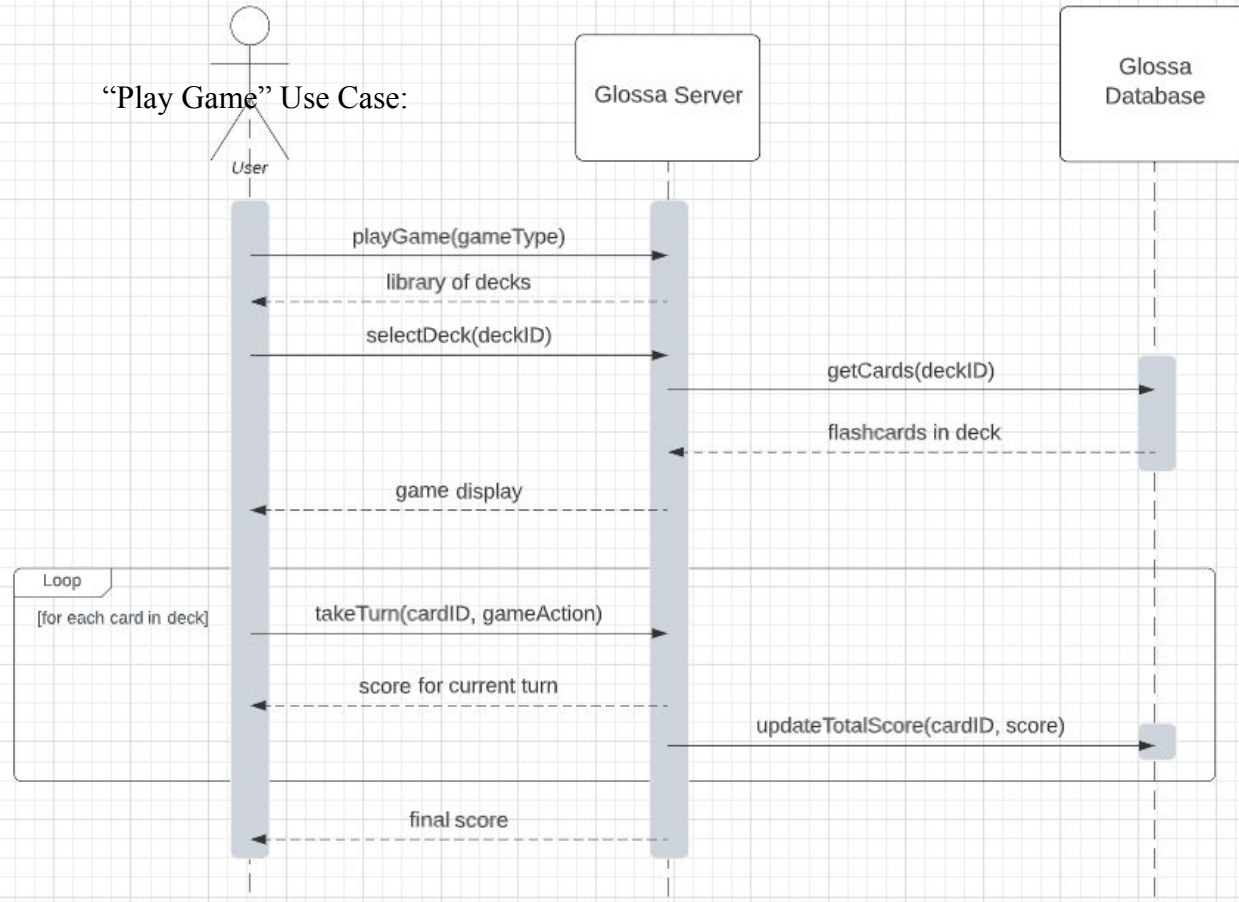
Use Case

Use Case Diagram



After logging in, the user is presented with the option to either play games or view the library. The game category uses decks from the library to generate games for the user. The view library function has powerful functionality that allows you to manipulate your studying decks by adding, removing, or editing elements of the deck. From there you can study decks with options to read/write or listen/speak. The Glossa server is responsible for maintaining the library and the data inside of it.

Sequence Diagram for the “Play Game” Use Case

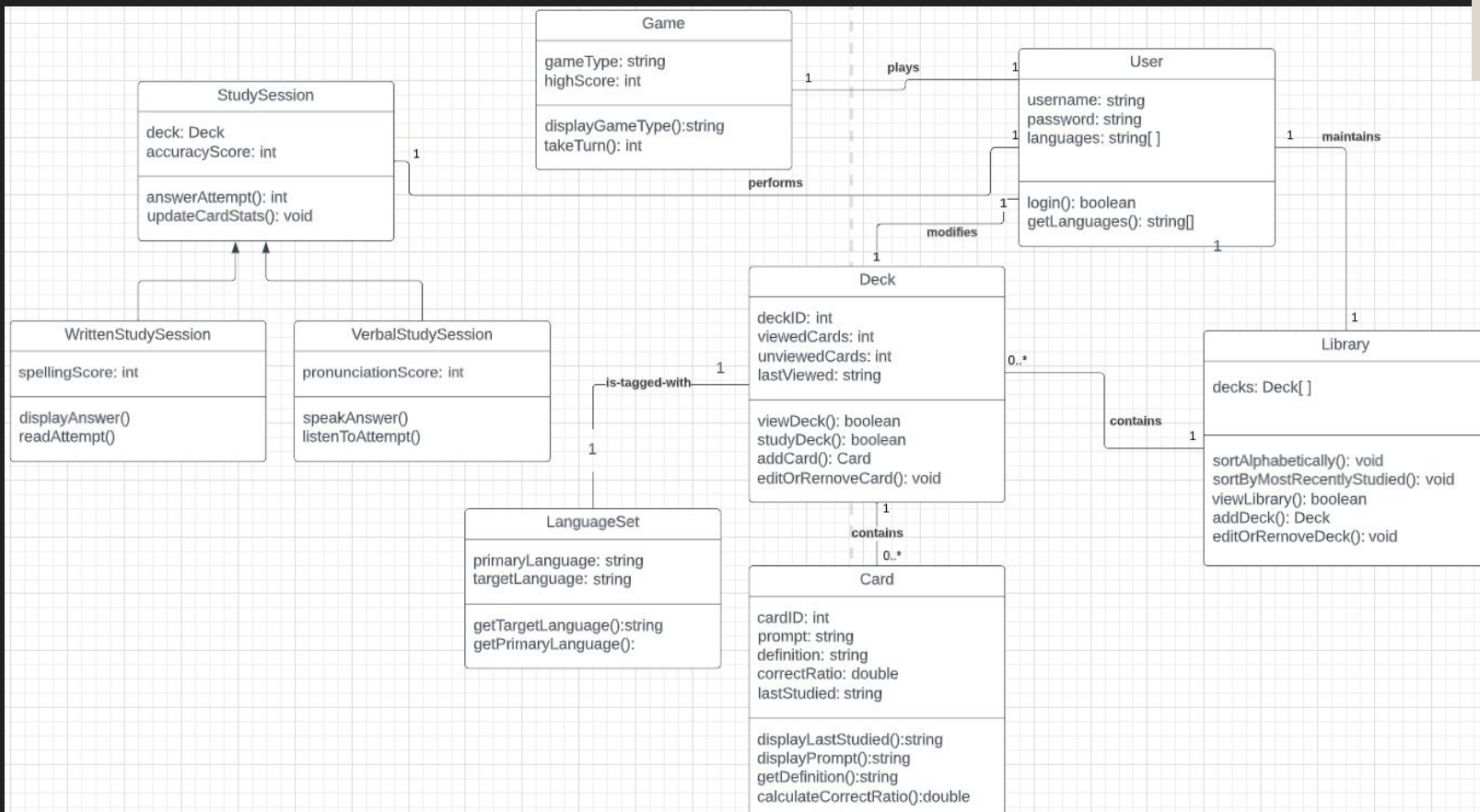


Class Diagram



Glossa

Learning Languages Made Easy.

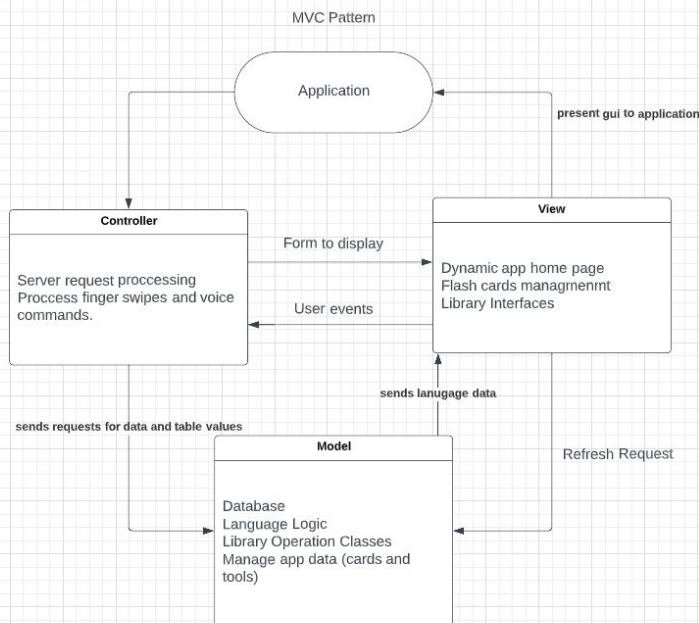


Class Diagram/ Description

- **Game:** Game class shows the types of games available to the user.
- **User:** User Class contains parameters to verify the login information of the user.
- **Deck:** Deck Class allows the cards to be sorted alphabetically or by recently studied. Decks can be added, modified, and deleted.
- **Language:** Language class holds the current language String of the Users chosen deck.
- **Card:** Includes a prompt, the prompt's definition, a correct ratio that tracks how many times you get a card right, and a last studied variable that tracks the last prompt you studied
- **Library:** Class that holds all the cards in an array called Deck[]. This class can sort the cards alphabetically, sort by recently studied. This class can also add decks, remove decks and edit decks to give the user more control of what they study
- **StudySession:** Represents a session of a user studying a flashcard deck
- **WrittenStudySession:** Type of StudySession focused on literacy
- **VerbalStudySession:** Type of StudySession focused on verbal fluency



Architectural design



The Model View Controller model was chosen because of its design. It naturally separates the components so the system's components are forced to interact with each other and are not completely centralized. We wanted users to have multiple ways of viewing the flashcard data.

The Model component: This would manage our users slide deck containing each of their individualized flashcards. It would also contain the languages containing the syntax and grammar of a user's chosen language. It would contain the user's account data and progress in a given language diagram.

The View component: The view component contains the User interface which should be standardized for all users. This includes the presentation of slide decks and library interface, and user account details.

The Controller component: The controller component will regulate the user's interaction with the application. The controller will need to select cards and libraries by tapping the screen as well as swiping through flashcards.

Demo UI

Glossa Language Learning Application

Username : Password : password?"/>



Glossa

Learning Languages Made Easy.

Create new account