Quizliu – Design Document

Project: P02 – Makers Makin' It, Act I
Team (XiaoMack): Evan Khosh, Jake Liu, Thomas Mackey, Haowen Xiao
Target Ship Date: January 16, 2026

---

1. Project Summary

Quizliu is a micro-flashcard web application that allows users to create, share, and study small flashcard decks. Users can review content through three interactive modes:

- **Flashcard review** (flip-style learning)
- **Matching pairs game**
- **Quiz mode** (question/answer assessment)

The goal of the project is to provide a lightweight, user-friendly study tool that encourages repetition, engagement, and friendly competition through points.

---

2. Core Features

- User authentication (register, login, profile)
- Create and manage flashcard decks
- View and study decks created by yourself or others
- Three study/game modes: Flashcards, Match Pairs, Quiz

---

3. Technology Stack

Backend

- **Language:** Python
- **Framework:** Flask
- **Database:** SQLite3

Frontend

- **HTML** for structure
- **CSS (Bootstrap)** for layout and styling
- **JavaScript** for interactivity and game logic

APIs

- None (internal Flask routes only)

---

4. Program Components

Backend Components

- `sqlite3` – persistent data storage
- `user_info` table – stores users and progress
- Flask app (`__init__.py`) – routing, authentication, database access

Frontend Pages (HTML)

- `login.html`
- `register.html`
- `profile.html`
- `home.html`
- `create.html`
- `flashcards.html`
- `match_pair.html`
- `quiz.html`

JavaScript Files

- `flashcards.js` – flashcard flip and navigation logic
- `match_pairs.js` – matching game logic
- `quiz.js` – quiz scoring and question handling

---

5. Component Explanations

Database Access

The **SQLite3 database** is accessed by multiple frontend pages through Flask routes:

- **register.html** – inserts new user records
- **login.html** – validates user credentials
- **profile.html** – displays user stats and decks
- **home.html** – shows dashboard and available flashcards
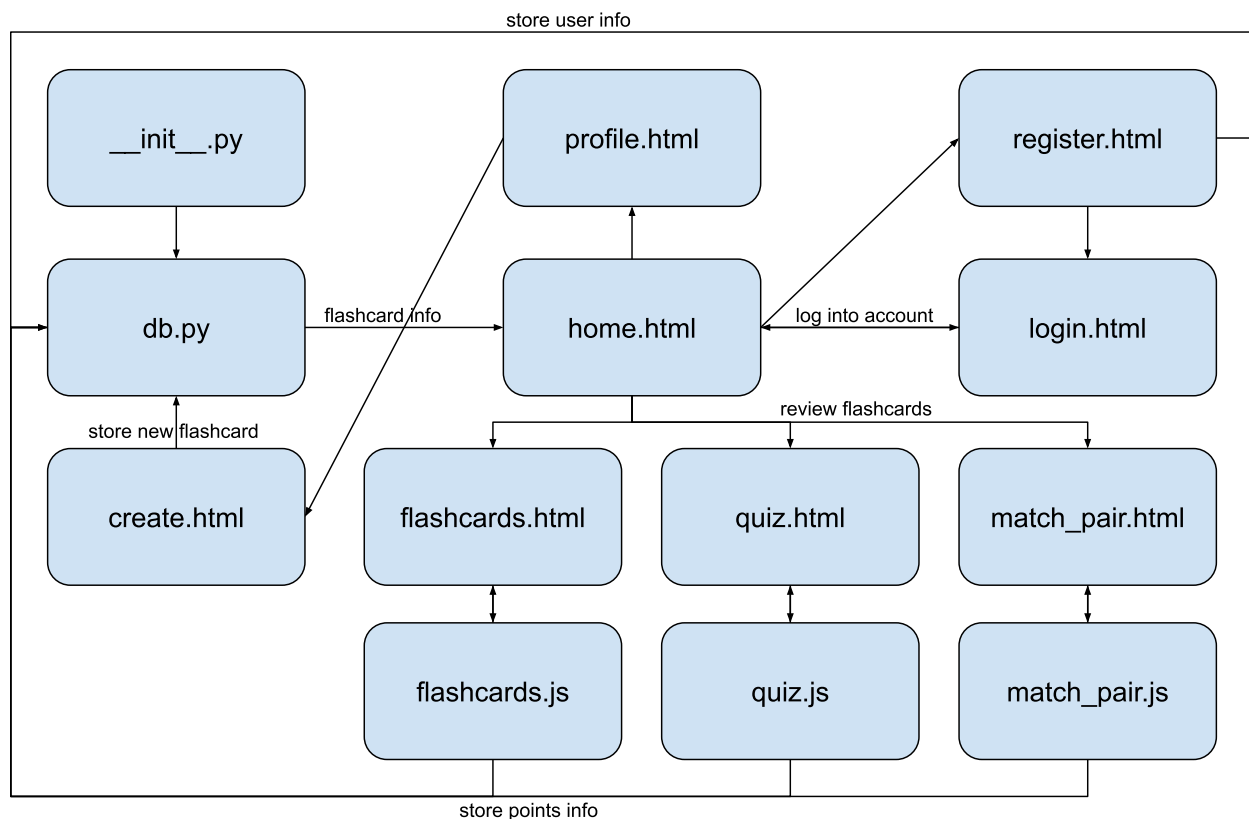- **create.html** – saves new flashcard sets

Game Interaction Flow

Each game page uses a dedicated JavaScript file to control interactivity:

- `flashcards.html` ↔ `flashcards.js`
- `match_pair.html` ↔ `match_pairs.js`
- `quiz.html` ↔ `quiz.js`

Flask supplies the data, JavaScript handles gameplay, and HTML/CSS renders the UI.

---

6. Component Map (UML Diagram)
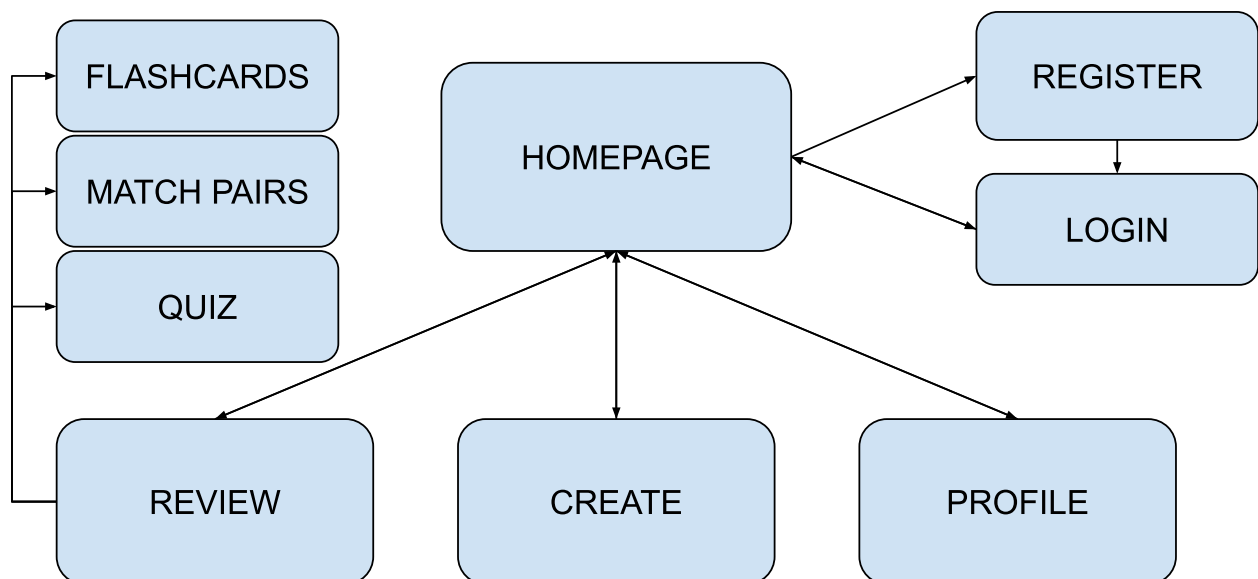
7. Database Organization

USER_INFO Table

Field	Type	Constraints
username	TEXT	PK, NOT NULL
password	TEXT	NOT NULL
points	INTEGER	NOT NULL
flashcards	TEXT	

FLASHCARDS Table

TEXT	title	NOT NULL
TEXT	creator	NOT NULL
INTEGER	card	NOT NULL
TEXT	front	NOT NULL
TEXT	back	NOT NULL

8. Site Map (UML Diagram)

9. Team Roles & Responsibilities

- **Evan Khosh (PM):** Database design, Flask backend, project coordination
- **Jake Liu:** Home page, Create page, Flashcards UI
- **Thomas Mackey:** Registration, Login, Profile pages
- **Haowen Xiao:** CSS styling, Match Pair game, Quiz game

10. Front-End Framework (FEF)

- **Bootstrap** is used for:
  - Responsive layout
  - Consistent styling
  - Faster UI development