

Semester In-Class Project Brief

Objective

During this semester, students will learn about component design, MVC design, ReSTful API services, document storage, tiered architected systems, job systems, caching, testing, and the importance of separating concerns. To illustrate this, in a practical example, we'll be building an interactive, multi-player, game, like Cards Against Humanity (but safe for school).

Architecture

Utilizing the idea of separation of concern, we'll follow a tiered architecture strategy. The following illustrates that plan:

- **A dedicated ReSTful API**
 - Routes for managing content, users and profiles, and games
 - Routes for managing authentication
 - Routes for accessing above systems for users and admins
- **A dedicated admin content management system (CMS)**
 - UI for authentication and authorization
 - UI for content management
 - UI for game management and monitoring
- **A dedicated front-end application**
 - UI for authentication
 - UI for game management
 - UI for profile management
- **A message broker**
 - Image optimization management
- **A cache broker**
 - Content redundancy management
- **Web socket manager**
 - Multiplayer system manager

Release Plan

Feature	Release Schedule
ReSTful API	Week 6
CMS	Week 13
Front-End UI	Week 13
Message Broker	Week 6
Cache Broker	Week 13

Web socket manager	Week 9
--------------------	--------

Risks

- Message Broker: This feature requires a service known as Redis to implement, and that may limit us as it is complex to run
- Cache Broker: This feature also requires Redis, and therefore may have the same risk as the message broker feature
- Web socket manager: This feature is quite complex, and developer understanding could prohibit this feature