

Design Document for Perfect Ten

Group **3_rk_3**

John Barmen: Design description, API documentation for backend

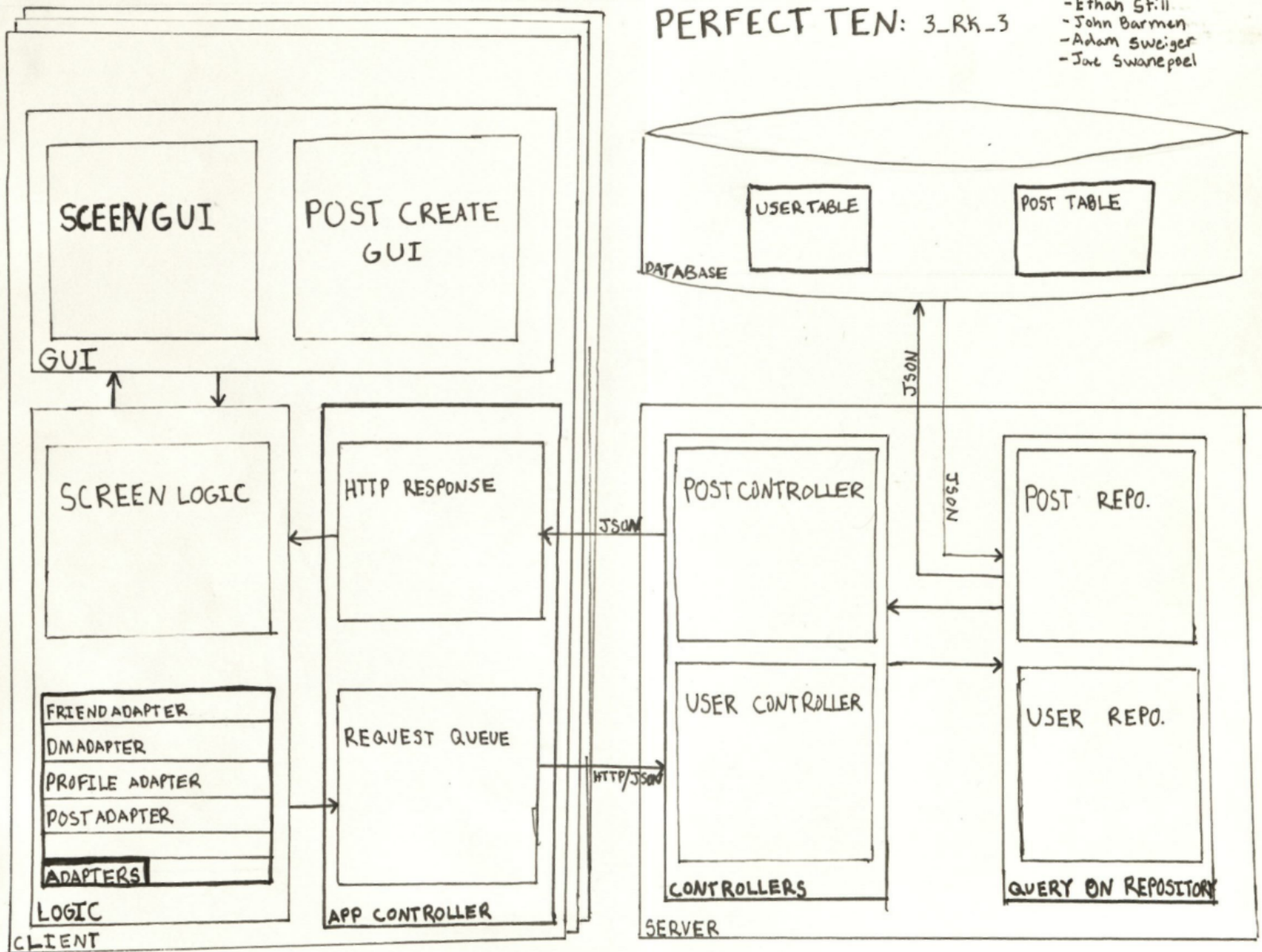
Ethan Still: Block Diagram, API documentation for frontend

Jae Swanepoel: Design description, API documentation for frontend

Adam Sweiger: Database relationship diagram, API documentation for backend

PERFECT TEN: 3_RK_3

-Ethan Still
-John Barmen
-Adam Sweiger
-Jae Swanepeel



The majority of the functionality behind Perfect Ten is maintained through intricate database management. For an end-user, most of the Perfect Ten experience comes down to finding different ways to view and engage with content. This is achieved in essentially two ways: **server calls that request data to view**, and **server calls that create new, change existing, or remove database entries**. The following lists many of the ways Perfect Ten users can interact with the database.

Requesting Data

- Post View
 - On the home screen and within user profiles a client is shown posts requested from the server. These HTTP requests are used to get lists of json objects, each of which represents a post. These json objects are then parsed and translated into post views for the user to read.

User Information

- On the profile view screen a user's information is displayed. This information is requested from the server using an HTTP request which requests the json object that represents the user from the user repository that is managed by the back end server. These json objects contain all of the information related to the user or in the case of requesting a list of friends, just the friends of a given user.

Making Database Entries

- Creating an Account
 - A user can create an account on the Signup Screen. The user must enter the fields for username, password, and email. These fields are sent to the server for verification in the form of a json object. If the entered credentials are valid, the client receives a success message and is redirected to the Login Screen.
- Changing Friendship Status
 - A user can add a friend through the Add Friend Screen or the Profile View Screen. The client's username is added to the parameters of the HTTP request, while the target username (the user to be friended) is included in the JSON Request Body. Upon a success response, the user will be given visual confirmation that they have added the target user as a friend. The Profile View Screen allows the client user to remove a friend using the same process.
- Creating and Modifying Posts
 - A user can from the home screen or any other location that a post is visible modify their own posts by sending a json request to the server specifying the id of a post to be modified. Post creation is a similar task but instead of passing an id one is assigned. Next the post is modified or created to match the json specifications and saved to the post repository.

Database Relationship Diagram

