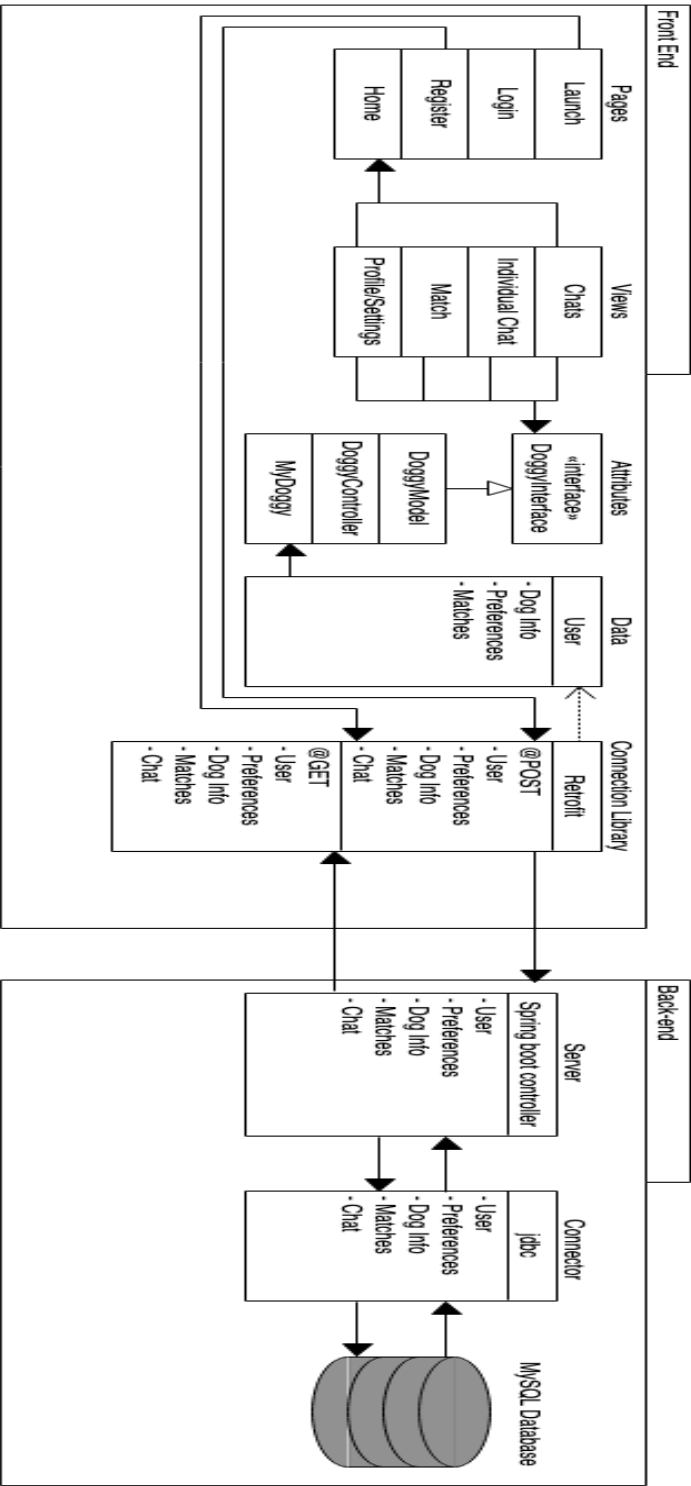


Block Diagram

Peyton Arbuckle
Diego Saavedra
Mark Gores
Corbin Graham

Doggy Match
3_rk_1

Block Diagram



Design Description

The application starts at the Launch page and that is where it checks for user data being active in the Bundle. If the Bundle is empty, the user will be sent to the Login page, or if the bundle contains user data, it will be sent to the Home page after a new request has been sent to the server to retrieve matches and chat data. This information, gathered from a Login, or Launch, will be stored in the MyDoggy class with User objects: to be applied throughout the UI using DoggyModel. Storing the data in MyDoggy allows for it to be retrieved one time and used quickly throughout the user experience. This prevents it from having to be refreshed every time the user switches between pages. These pages are all generated with XML. There are 3 layouts, each with multiple views. The view depends on the page and user.

The Match page displays information stored in MyDoggy's Match Collection (A list of User objects). These are ranked based upon interest matching (those with the best matches for criteria will be displayed first). When the client decides they do not want to match with a User, they can press no and it will be discarded from their list and never displayed again. If they decide yes, then it is added to a list of matches; if their ID is on both users' lists, then a Chatable function is generated. This will add them to a list of Chats where they can continue the matching process.

The Profile page allows for the client to modify or simply see their profile in the way that other users would be able to see it. They can update preferences they selected at Registration or add new ones. They may also change contact information or general user information.

Registration is a simple, straightforward process. It allows users to generate a profile based upon their Dog's information and criteria for finding a suitable dog for it to play with. If the user is not a dog owner, they can create a Viewer profile that allows them to see Dog's profiles without being able to generate matches.

Individual Features

Chat

- We will use webSockets.
- Manages the chat between players. Send and receive messages.

Model

- User objects will be received and sent as a JSON object to the database.
- We will use byCryptEncription to convert the user's password into a long string using the 'Salting' method.

Tables

- The tables on the database will contain all the users information, and login status including its last login time.
- The chats between users will not be used for any commercial purposes, and will be safely stored on the database.

Tables and Fields

