Bomshika and TestAccount like this

Rails Project

Pictogram

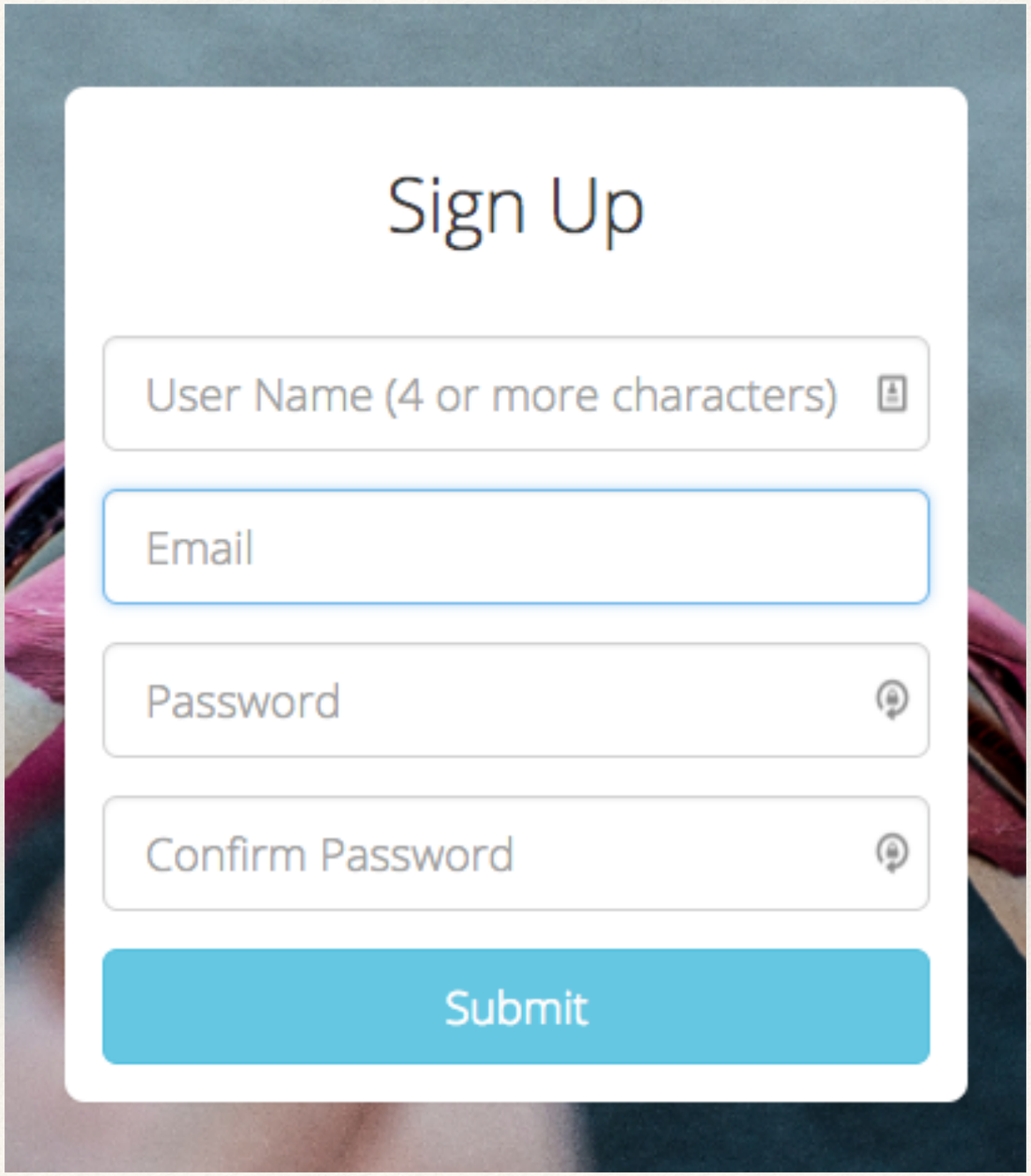
Michael Taylor
20064376
BSc (Hons) Multimedia
Application Development
Web Server Programming

Pictogram

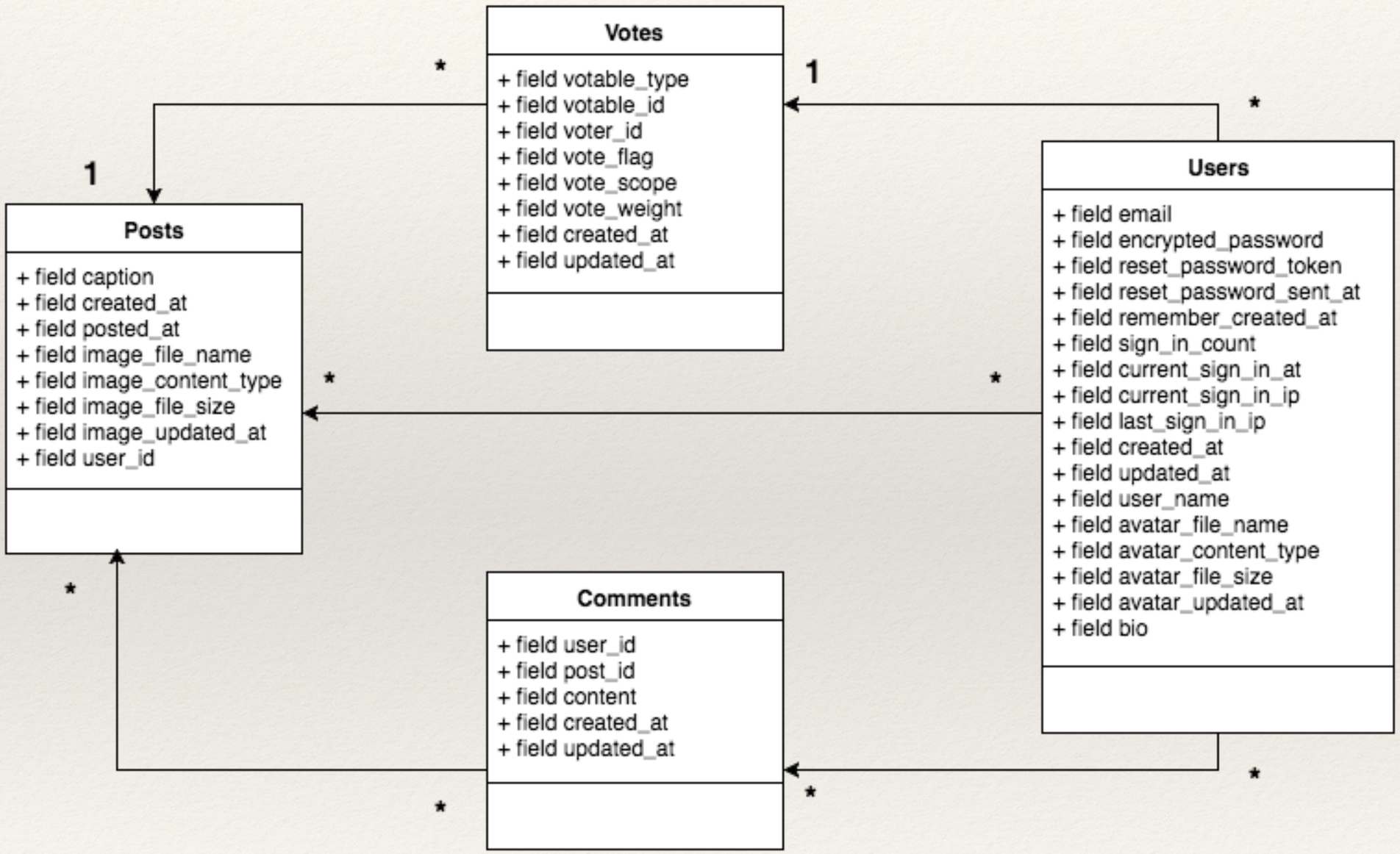
- Pictogram is an Instagram style application
- A user can create an account, upload pictures, comment and like their own posts as well as other pictogram users
- Before a user can view content on pictogram, they have to either login to their account or create an account
- A user can update posts, captions or if unsatisfied, delete the post

Sign up

- A user can choice a unique username, must be between 4 and 20 characters
- Email must be valid and contain @
- Their password must be at least 6 characters
- If a user fails to enter in the correct information, upon clicking submit, they will be prompted with error checks for each wrong field

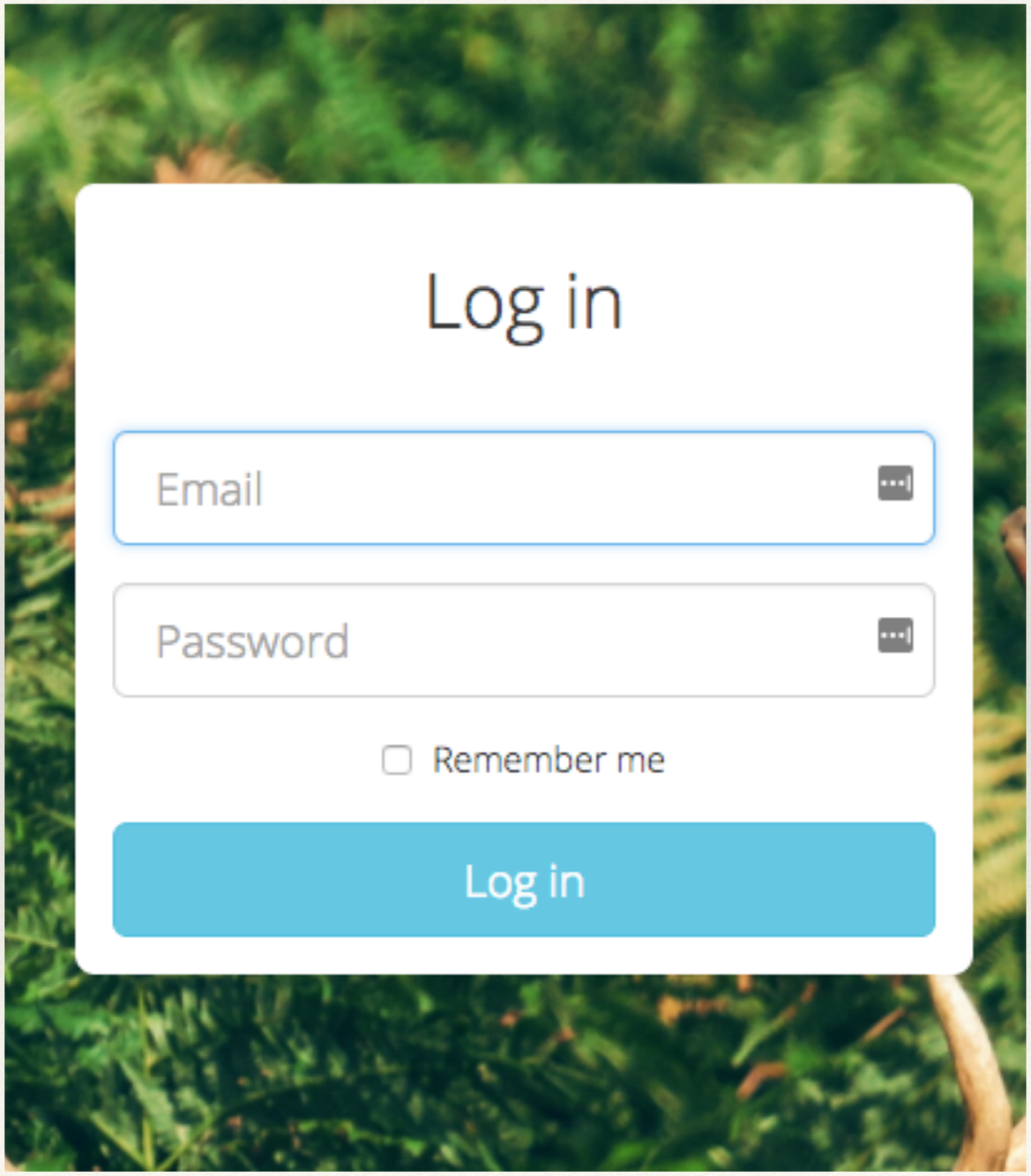


Data Model

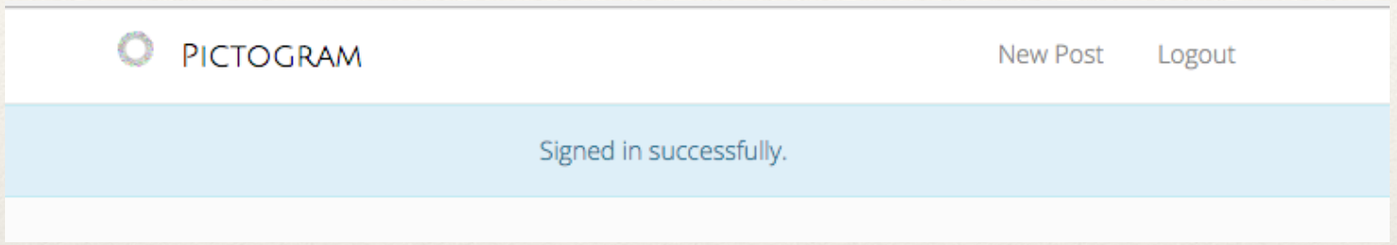

Votes
+ field votable_type
+ field votable_id
+ field voter_id
+ field vote_flag
+ field vote_scope
+ field vote_weight
+ field created_at
+ field updated_at
Posts
+ field caption
+ field created_at
+ field posted_at
+ field image_file_name
+ field image_content_type
+ field image_file_size
+ field image_updated_at
+ field user_id
Users
+ field email
+ field encrypted_password
+ field reset_password_token
+ field reset_password_sent_at
+ field remember_created_at
+ field sign_in_count
+ field current_sign_in_at
+ field current_sign_in_ip
+ field last_sign_in_ip
+ field created_at
+ field updated_at
+ field user_name
+ field avatar_file_name
+ field avatar_content_type
+ field avatar_file_size
+ field avatar_updated_at
+ field bio
Comments
+ field user_id
+ field post_id
+ field content
+ field created_at
+ field updated_at
1
*

Login

- If a user has logged out, they can sign into their account
- If a user enters in the wrong information, a flash message will be displayed information them of the wrong email / password combination



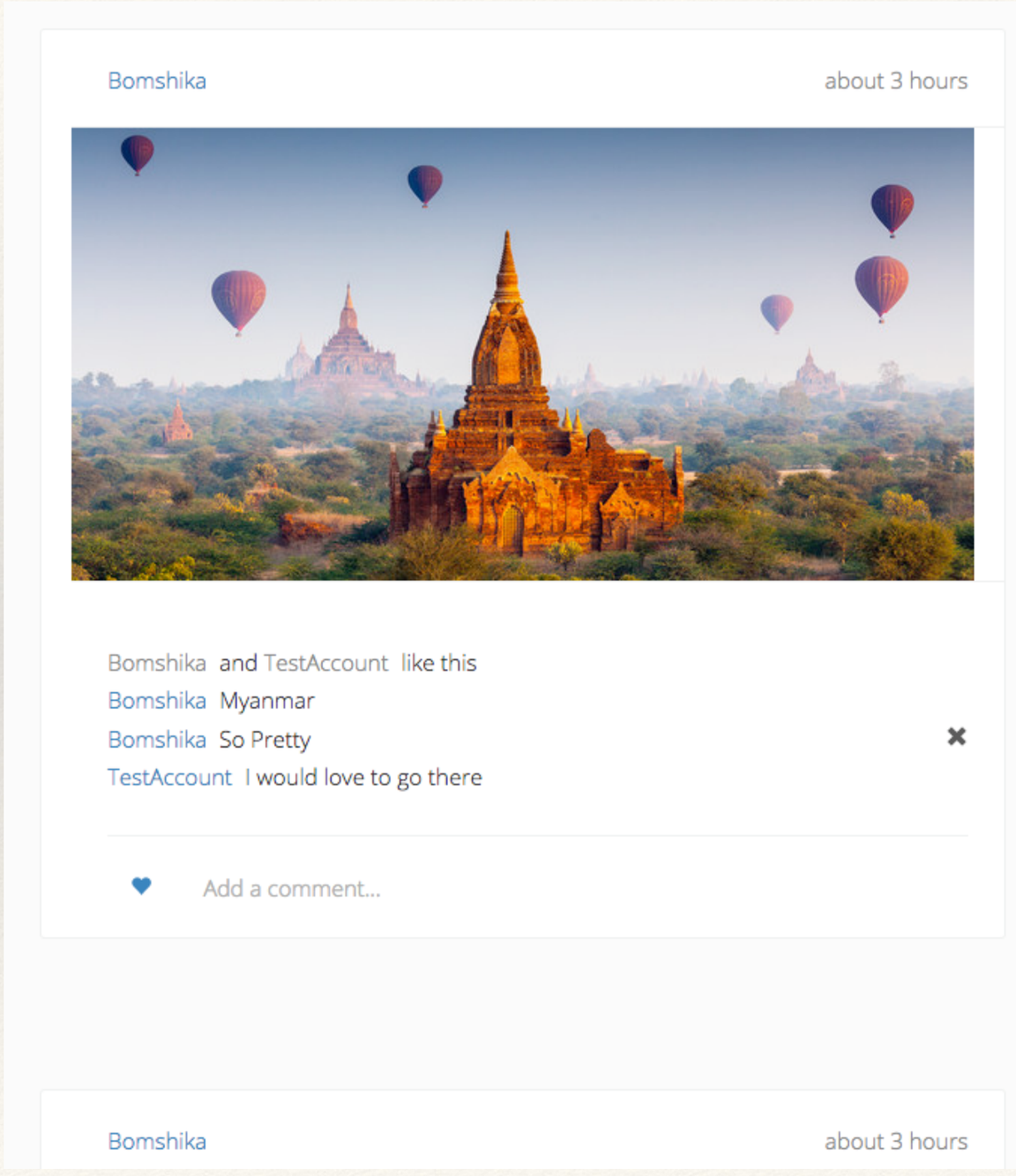
Navigation



- Pictogram has a clean and simple navigation
- The navigation is also dynamic, the first picture illustrates the navigation when not logged in
- When a user is logged in, the navigation will change to suit their needs better

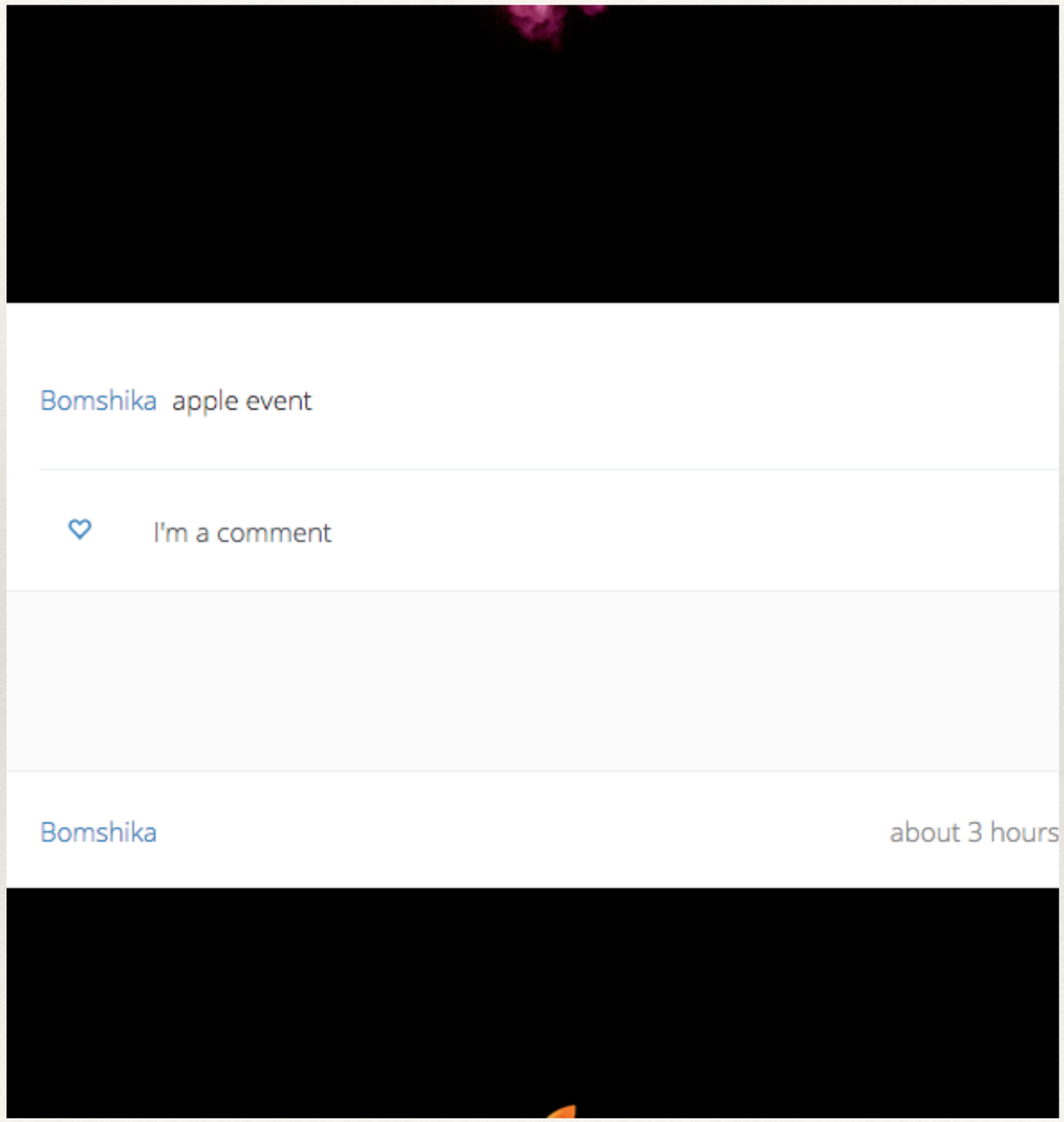
Feed

- When a user is signed in, they are directed to the main page called their feed
- The feed is made up of other pictogram users pictures
- the feed will show who created a post, roughly when they did, the image, and caption associated with the post



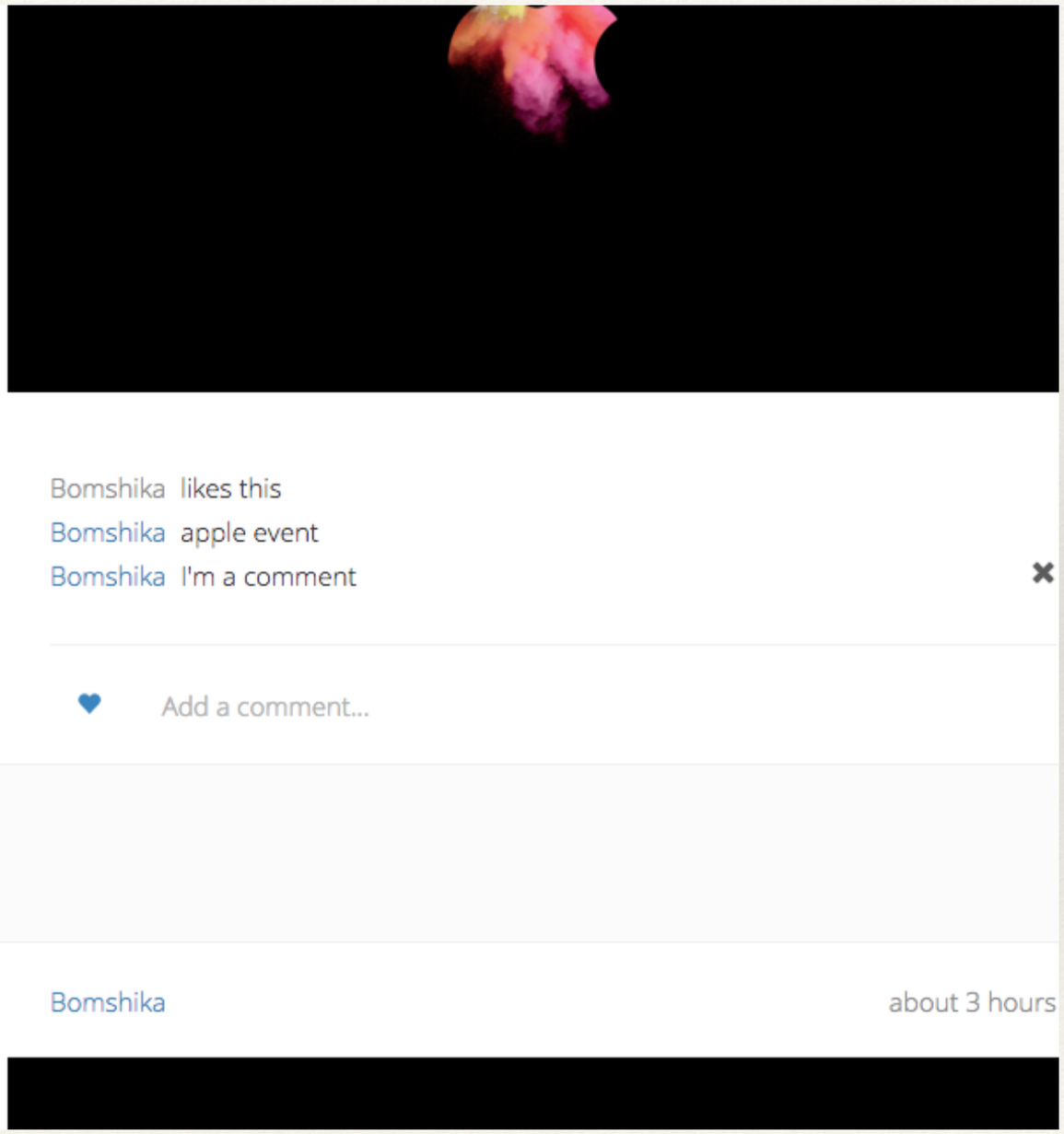
Comments

- A user can comment on pictures in their feed as demonstrated
- If a user spots a spelling error or decided their comment is not witty enough, they can delete their comment
- Using pagination, only the top four comments are shown, clicking on “more comments” will display all the comments on a post



Likes

- A user can also like their own post or another users
- The heart is animated, when a post is not liked, the heart is empty and when a post is liked, the heart is filled

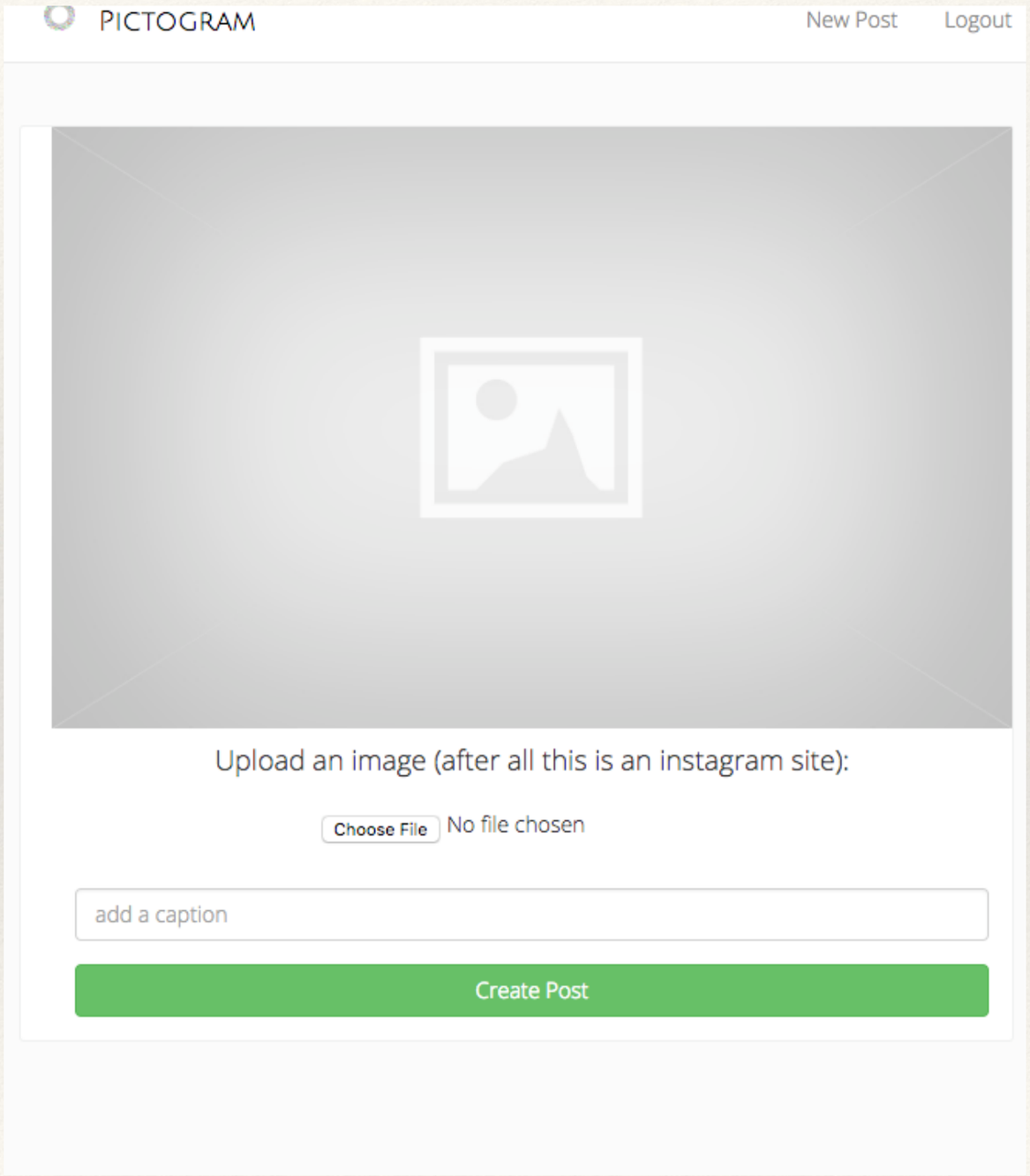


Pagination

- Using pagination, the number of posts per page is limited
- To see more posts the user has to click the “load more’ button

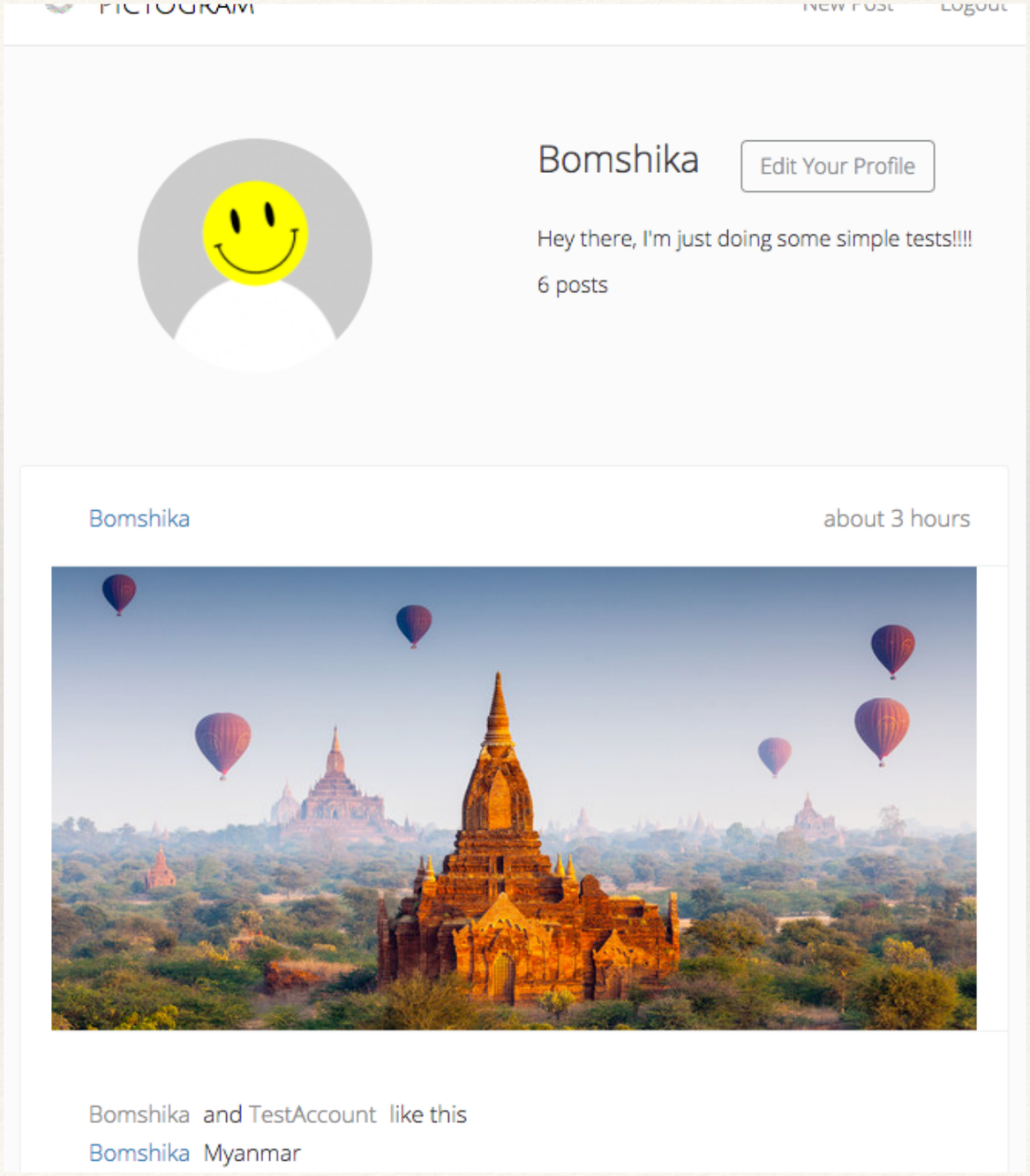
Upload image

- As pictogram is an Instagram style application, one of the main focuses is images
- A user can upload an image from their computer, and add a caption
- On the upload page, the user is shown the image illustrated. The page has a placeholder image, and shows them where to add a caption



Profile

- Each user has their own profile page
- This page will contain all of their uploaded posts
- A user can edit their profile with a bio section and change their user image



Independent learning

- For image uploading, the student used *gem paperclip*. This was an easy method to attach files to the ActiveRecord. The files aren't saved to their final location on disk but rather saved to the filesystem and can then be referenced by the browser.
- Instead of creating forms by hand, the student used *gem simple_form*, this allowed the student more flexibility with their forms.
- The student used *gem devise* for the authentication, devise allowed them to hash the users password before storing it in the database. Devise allowed for authentication to occur using POST requests.
- The student used *gem acts_as_votable*, to create the "this user has liked this post". This gem allows for any model to be voted on, or liked in this case. Using this gem, they student could show the first few likes, and list the remainder as "8 more people liked this".
- For pagination, the student used *gem kaminari*. This gem both for the posts and comments sections. The student changed the default configuration of this gem from displaying 25 items to 8 items.
- The student is hosting their application on DigitalOcean, to convert the application from local hosting and sql database. To do this the student used *gem unicorn* as the HTTP server and and *gem pg* for the postgresSQL interface for the database.