

ORIGINAL APP Design Project - README Template

NATURE WALK

Table of Contents

Overview

Product Spec

Wireframes

Schema

Overview

Description

Encyclopedia of animals/plants where people can take pictures of animals and plants and drop a location of where the photo was taken. Tentative: By clicking the location link a map shows up to the location.

App Evaluation

[Evaluation of your app across the following attributes]

Category:

Mobile:

Story:

Market:

Habit:

Scope:

Product Spec

1. User Stories (Required and Optional):

Required Must-have Stories

User can sign up

User can login / log out

User can populate timeline with posts about an animal

User can upload pictures and a description with post

User can tag geo location of where they took a picture is at

Optional Nice-to-have Stories:

User is displayed a map when clicking on the location

...

2. Screen Archetypes

Login / Register

User can sign up

User can login / log out

Stream

User can see a timeline with posts about animals

Creation

User post a picture with a description

User can tag geo location of where they took a picture is at

3. Navigation

Tab Navigation (Tab to Screen)

Profile Tab

Home Tab

Compose Tab

Flow Navigation (Screen to Screen)

[list first screen here]

[list screen navigation here]

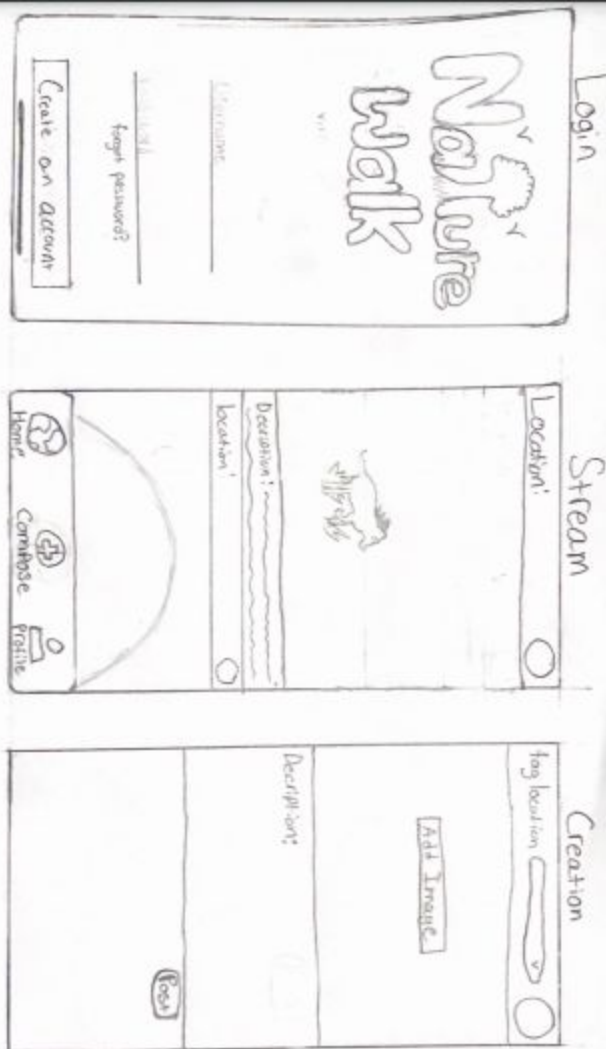
...

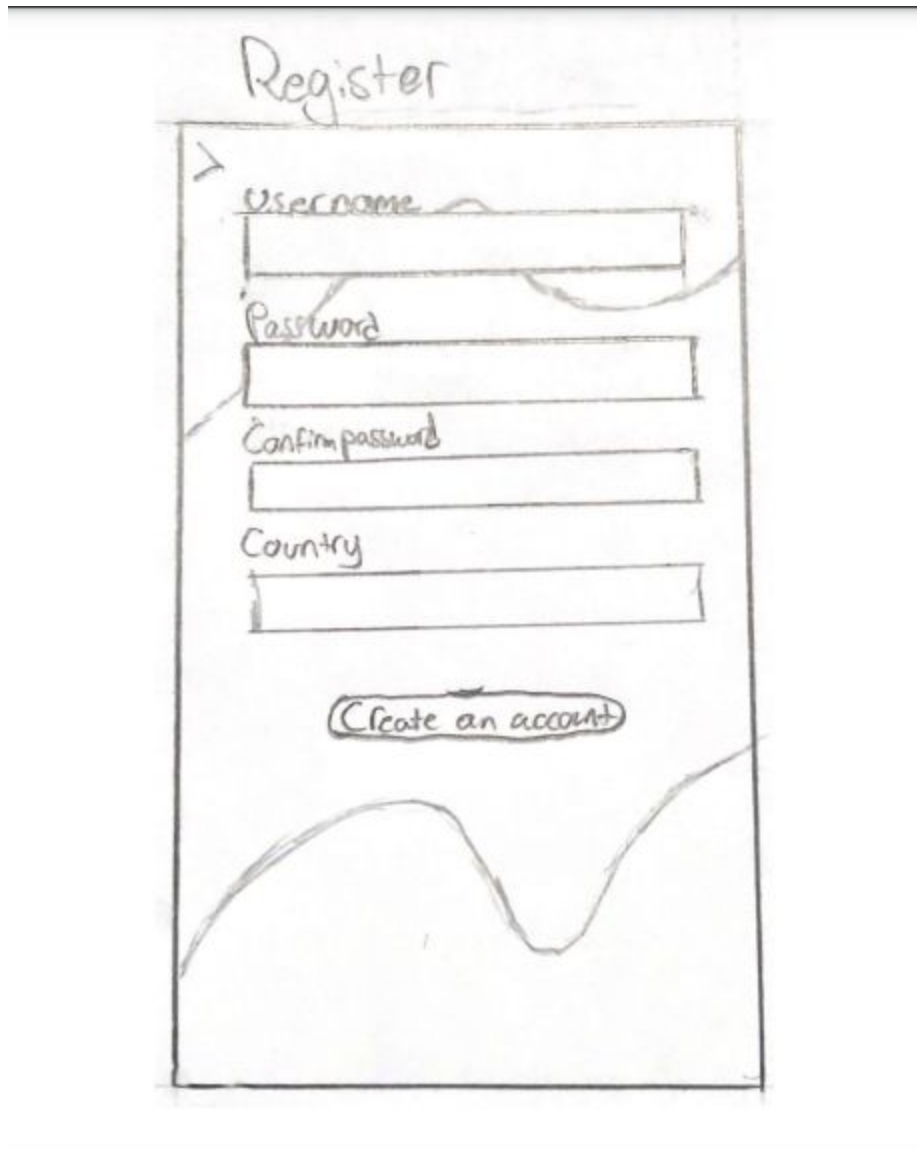
[list second screen here]

[list screen navigation here]

...

Wireframes





[BONUS] Digital Wireframes & Mockups

[BONUS] Interactive Prototype

Schema

Models

Post

Property	Type	Description
-----	-----	-----
userId	String	unique id for the user post (default field)
author	Pointer to User	image author

image	File	image that user posts
caption	String	image caption by author
createdAt	DateTime	date when post is created (default field)
location	String	name of nearby location of where the image was taken

Networking

List of network requests by screen

- Home Feed Screen

- (Read/GET)

```

    ``ParseQuery<Post> query = ParseQuery.getQuery(Post.class);
    query.include(Post.KEY_USER);
    query.findInBackground(new FindCallback<Post>() {
        @Override
        public void done(List<Post> posts, ParseException e) {
            if (e != null) {
                Log.e(TAG, "Issue with getting posts", e);
                return;
            }
            for (Post post : posts) {
                Log.i(TAG, "Post: " + post.getDescription() + ", username: " +
post.getUser().getUsername());
            }
        }
    });
    ``

```

- (Create/POST) Create a new like on a post

- (Create/POST) Create a new comment on a post

- (Delete) Delete existing comment

- Create Post Screen

- (Create/POST) Create a new post object

```

    ``
    Post post = new Post();
    post.setDescription(description);
    post.setImage(new ParseFile(photoFile));
    post.setUser(currentUser);
    post.saveInBackground(new SaveCallback() {
        @Override
        public void done(ParseException e) {
            if (e != null) {

```

```

        Log.e(TAG, "Error while saving", e);
        Toast.makeText(MainActivity.this, "Error while saving!",
Toast.LENGTH_SHORT).show();
    }
    Log.i(TAG, "Post save was successful!!");
    etDescription.setText("");
    ivPostImage.setImageResource(0);
}
})
...

```

- Profile Screen

- (Read/GET) Query all posts where user is author

```

```swift
let query = PFQuery(className:"Post")
query.whereKey("author", equalTo: currentUser)
query.order(byDescending: "createdAt")
query.findObjectsInBackground { (posts: [PFObject]?, error: Error?) in
 if let error = error {
 print(error.localizedDescription)
 } else if let posts = posts {
 print("Successfully retrieved \(posts.count) posts.")
 // TODO: Do something with posts...
 }
}
}
...

```

- (Read/GET) Query logged in user object
- (Update/PUT) Update user profile image