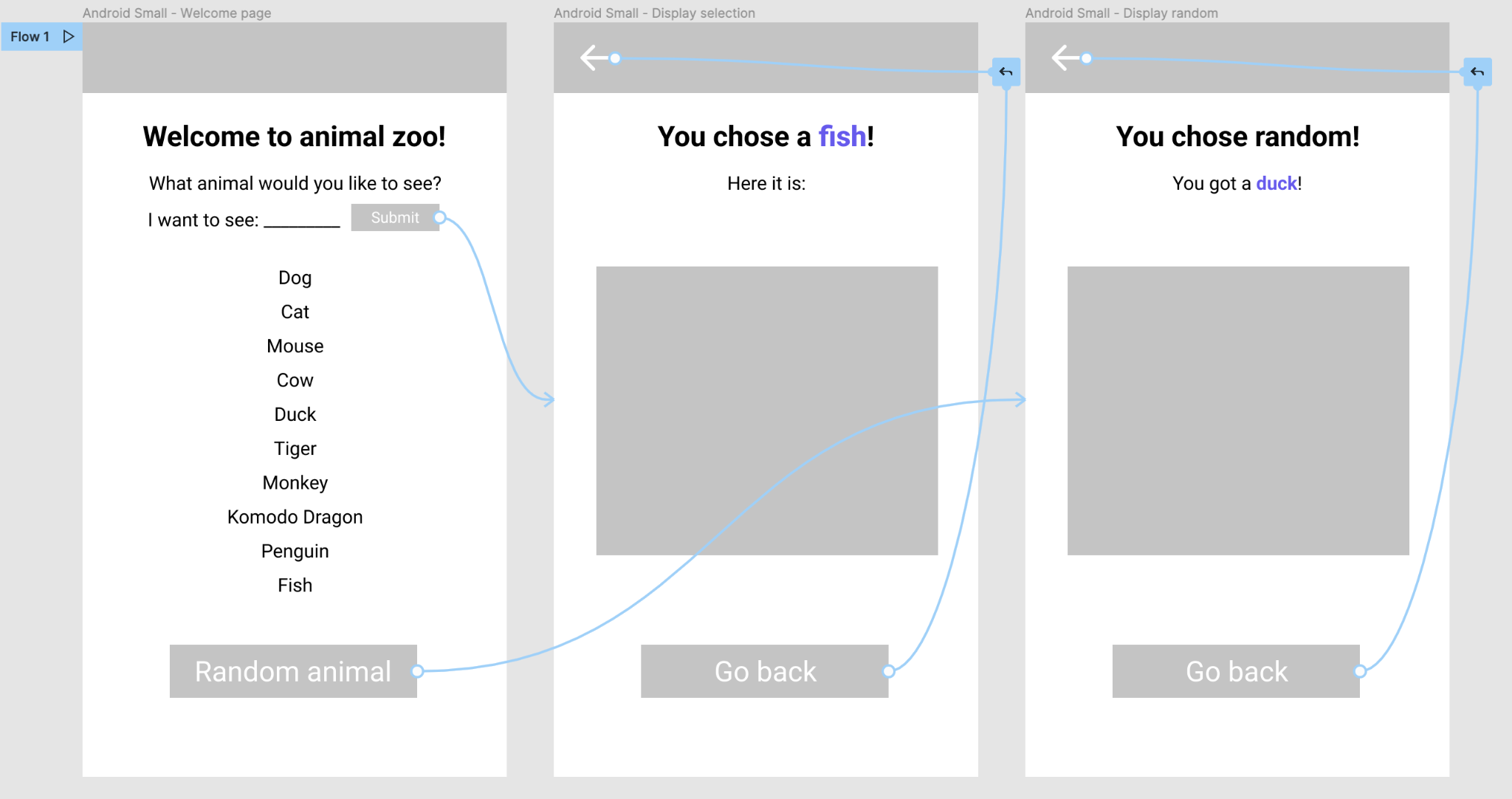
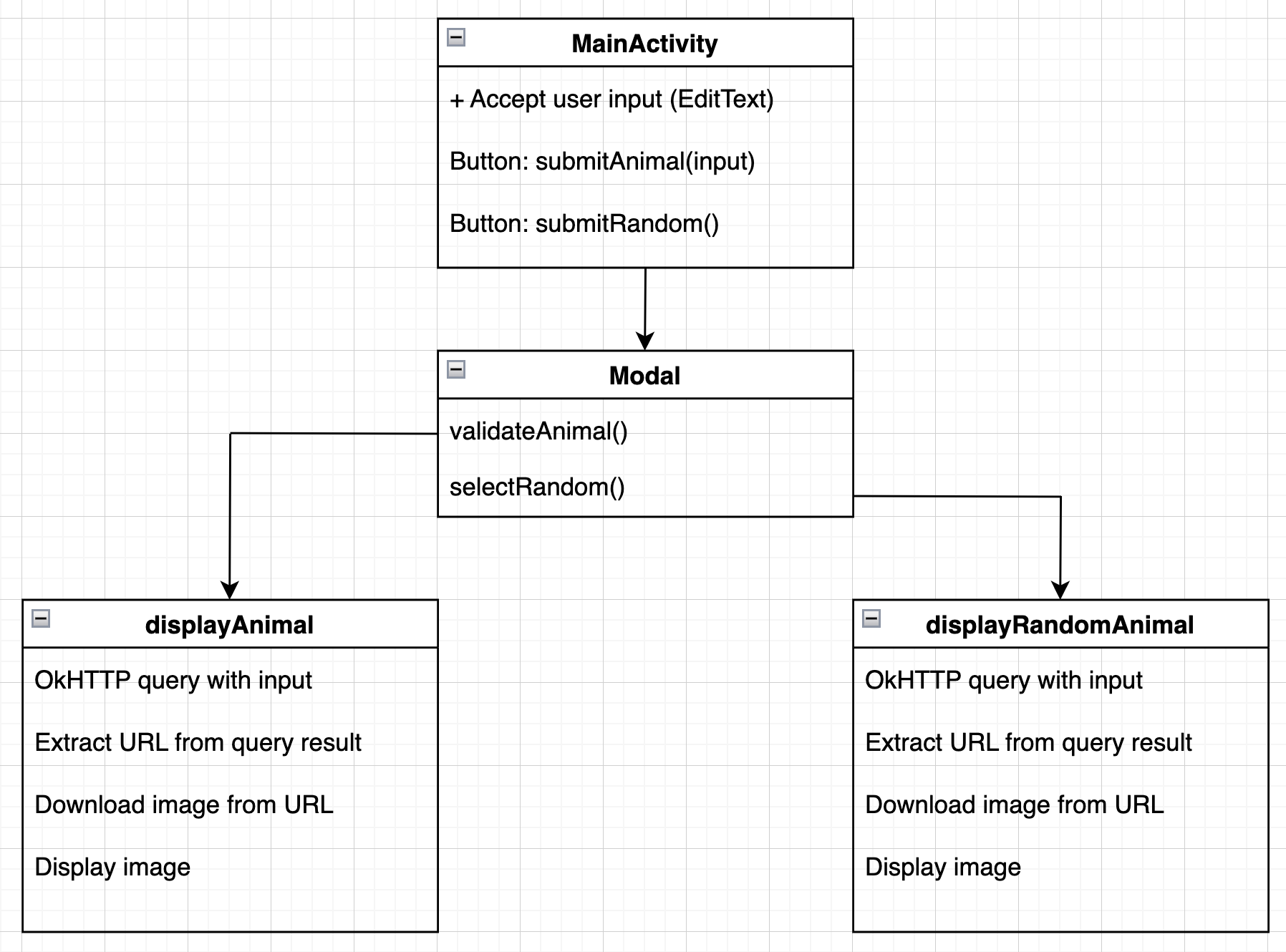
| **Name of Team:** ATV | | | | |
| --- | --- | --- | --- | --- |
| **First name:** Tianna | **Last name:** Fischer | **York Email:** tfischer@my.yorku.ca | **Lecture Section:** N | **Lab Section:** 1 |
| **First name:** Vidhi | **Last name:** Pandya | **York Email:** vidhi20@my.yorku.ca | **Lecture Section:** N | **Lab Section:** 1 |
| **First name:** Alex | **Last name:** Rigney | **York Email:** arigney@my.yorku.ca | **Lecture Section:** N | **Lab Section:** 2 |
| **Project Title:** Virtual Zoo | | | | |
| **Project Description:**  Our project will be a mini database of animal pictures. The user will be given a list of animals to select from. The user will input the name of an animal from the list and correspondingly, an image will be displayed in the ‘display area’. If the input is not valid or does not match an item in the list, the user will be prompted to ensure their input is correct. The images will be displayed via a url stored in the database. The image database will contain a list of ten animals. | | | | |
| **Functional Requirements:**   1. User can view a list of the 10 animals stored in the animal database 2. The database will store the names of the animals in an array for retrieval 3. User types the name of an animal from the list into a text input box 4. Capitalization of user input will not matter, input will be converted to lowercase string 5. User clicks a submit button to submit their choice of animal 6. If the input is valid, an image of the corresponding animal will be returned using an API 7. User will be prompted for a new input if the entered user input does not match any item in the database 8. There will also be a “random” button to return an image of a random animal from list 9. The image will be displayed in the display area on a new page following user input 10. The page that displays the images will have a back button that takes the user back to the input page | | | | |

**Phase 2: Design**

****

**High level design of classes:**

****

**Technologies and Tools:**

1. **APIs**

Our app uses okHttp and RapidAPI.

OkHttp is a request/response API that downloads a URL and prints its contents as a string. RapidAPI accesses Bing Image Search to run a query and return an array of JSON objects with the output. We use OkHttp to query Bing Image Search via RapidAPI as a URL. RapidAPI then returns an array of JSON objects for the images. We extract the first JSON object (first image) and get the thumbnail URL as a string. We then download the image from that thumbnail URL and display it in an ImageView.

1. **Languages used**

Java

1. **Tools**

Android Studio