Final Report

Dave DaCosta, Jake Adamson, Rob Cucchiara, Nathan Ko

Project goal: describes the purpose of this project

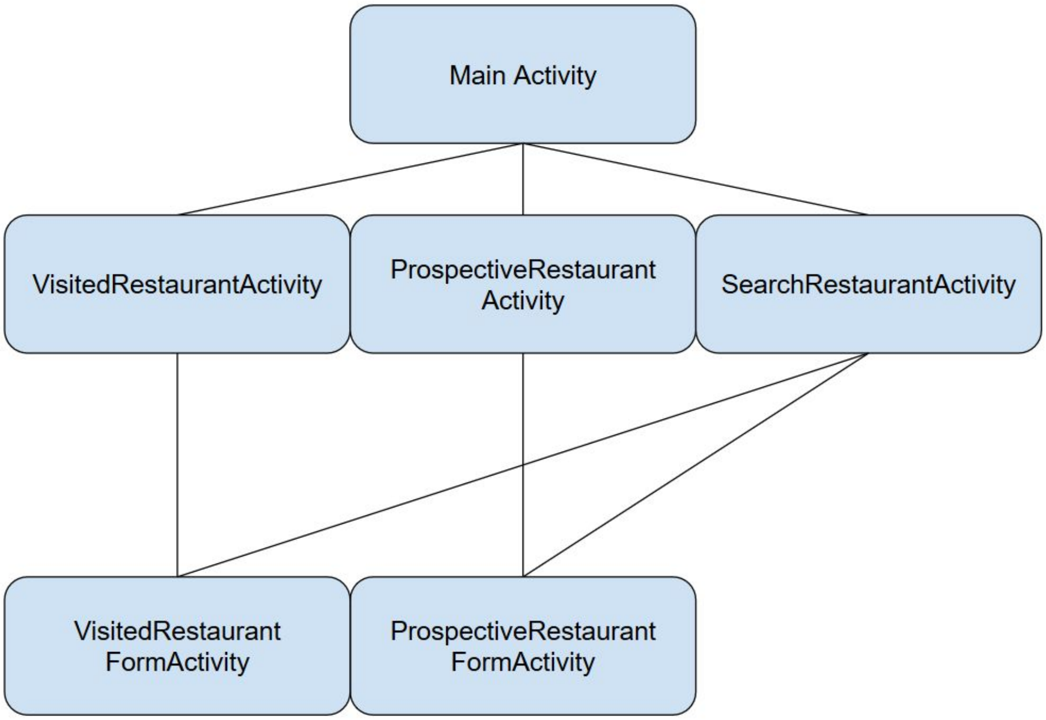
- The purpose of Restaurant Dairy is to be an app that lets you save the restaurants that you have visited or want to visit. It allows the user to take notes on their experience and uses IMB's Watson Ai/ML to give the restaurant a rating based on the notes. Also it is able to search for new restaurants or restaurants the user has been to, with the Yelp Api. This is convenient for looking up the restaurants by location.

Project features: abstract from the user stories

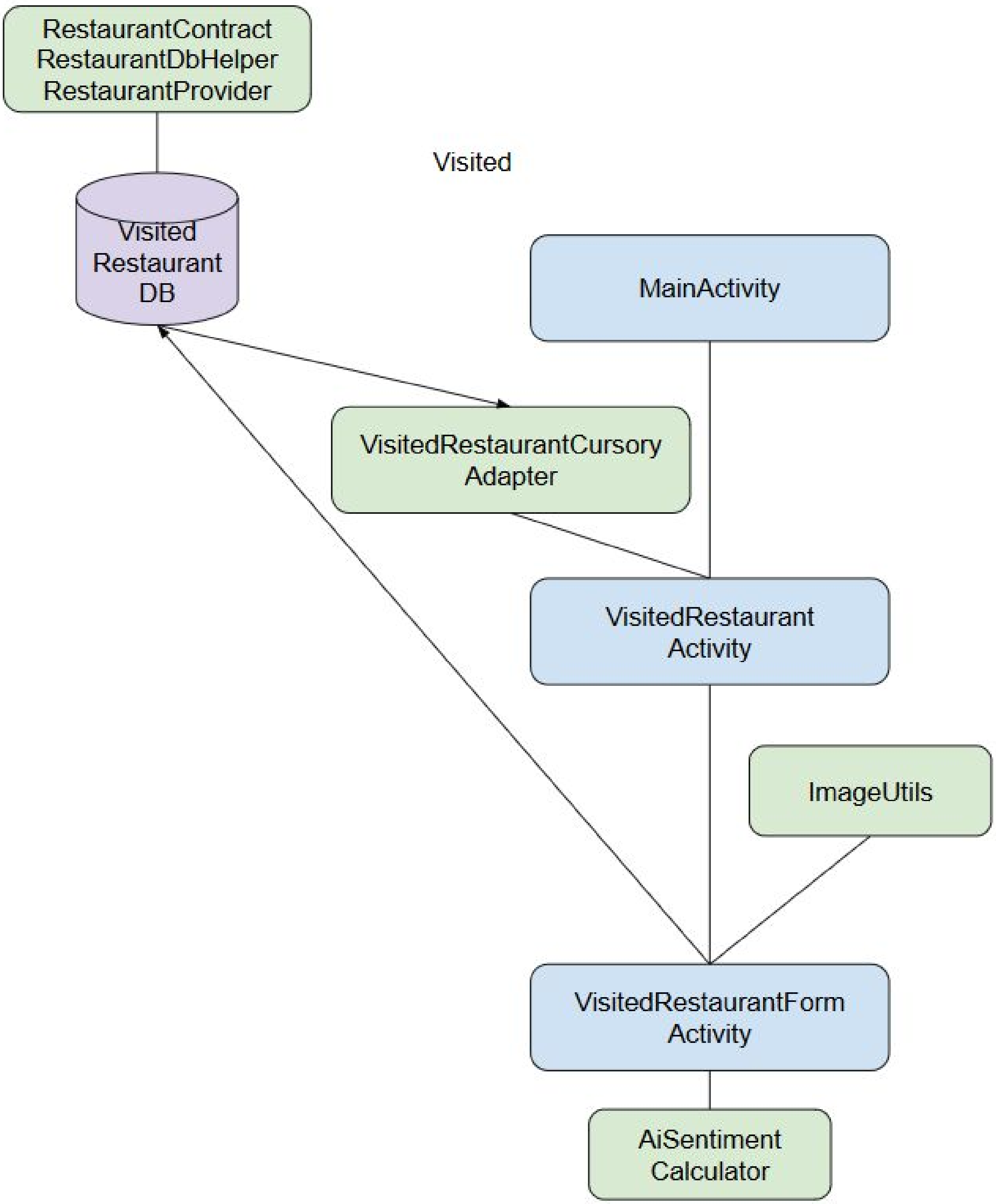
- The user is able to search on Yelp through the search function. This lets you view the restaurants for a particular town or location.
- The user is able to mark the restaurants they would like to visit from the listing on yelp.
- If the user wants to visit a restaurant seen on the yelp listing, pressing a button to add to the notes will take the user to the note taking form with the basic information prefilled, such as name, address, and phone number of the restaurant.
- The user is able to make a list of restaurants they would like to visit manually as well, without relying on yelp for those that aren't listed on yelp.
- The user can take notes on the restaurants they have visited and the restaurants they would like to go to. The notes consist of the name, address, phone number, and a summary of the experience or general comments.
- When the user edits a restaurant in the notes, the form checks to see if changes have been made before exiting the form, so that new information is not discarded accidently.
- The Watson machine learning API uses natural language processing to see how positive the user's review of the restaurant was, and displays either a smiley face, neutral, or frowning face in the notes listing.

Project design: visual diagram of project components/flow

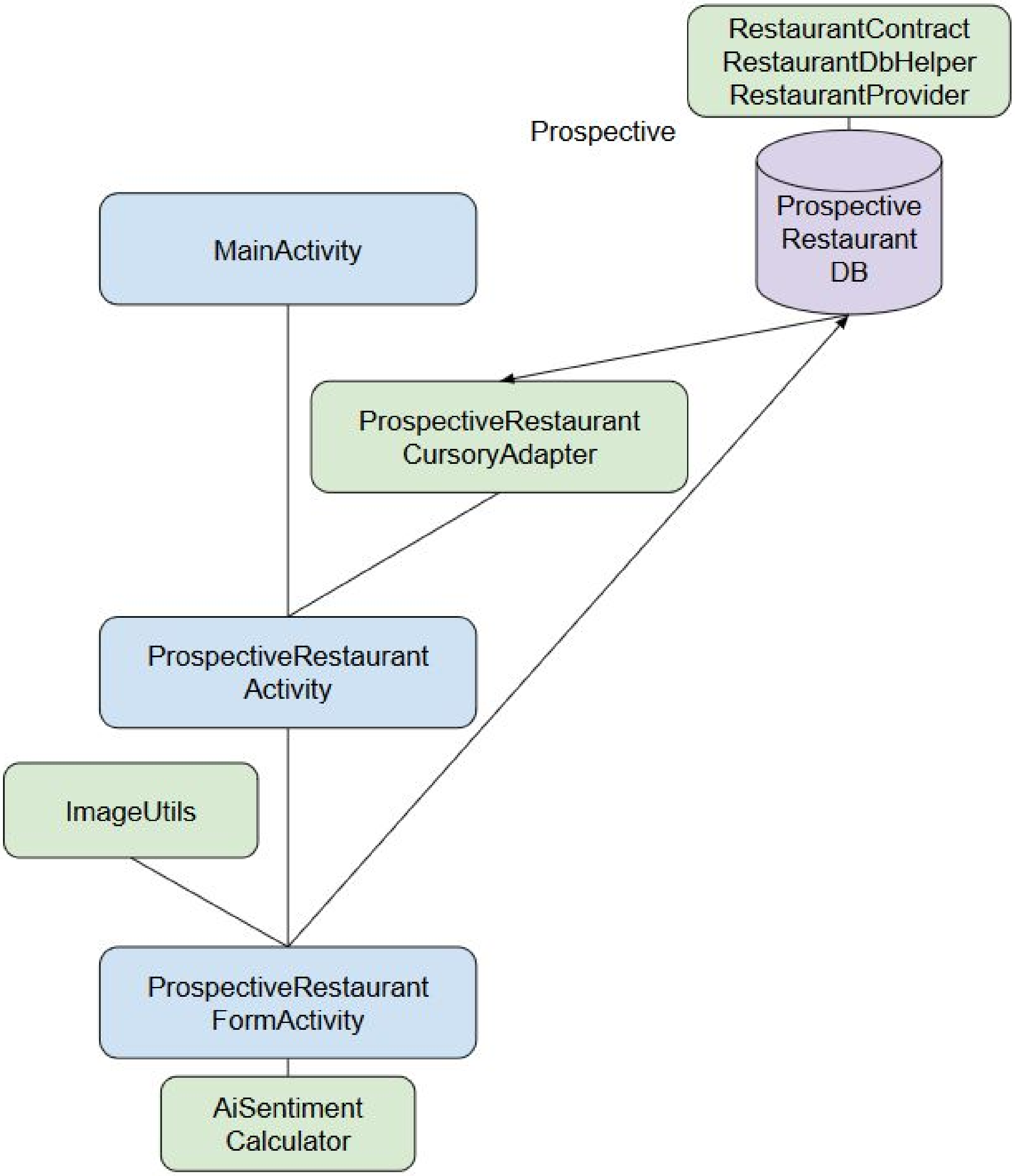
- Activity Flow:


Main Activity
VisitedRestaurantActivity
ProspectiveRestaurant Activity
SearchRestaurantActivity
VisitedRestaurant FormActivity
ProspectiveRestaurant FormActivity

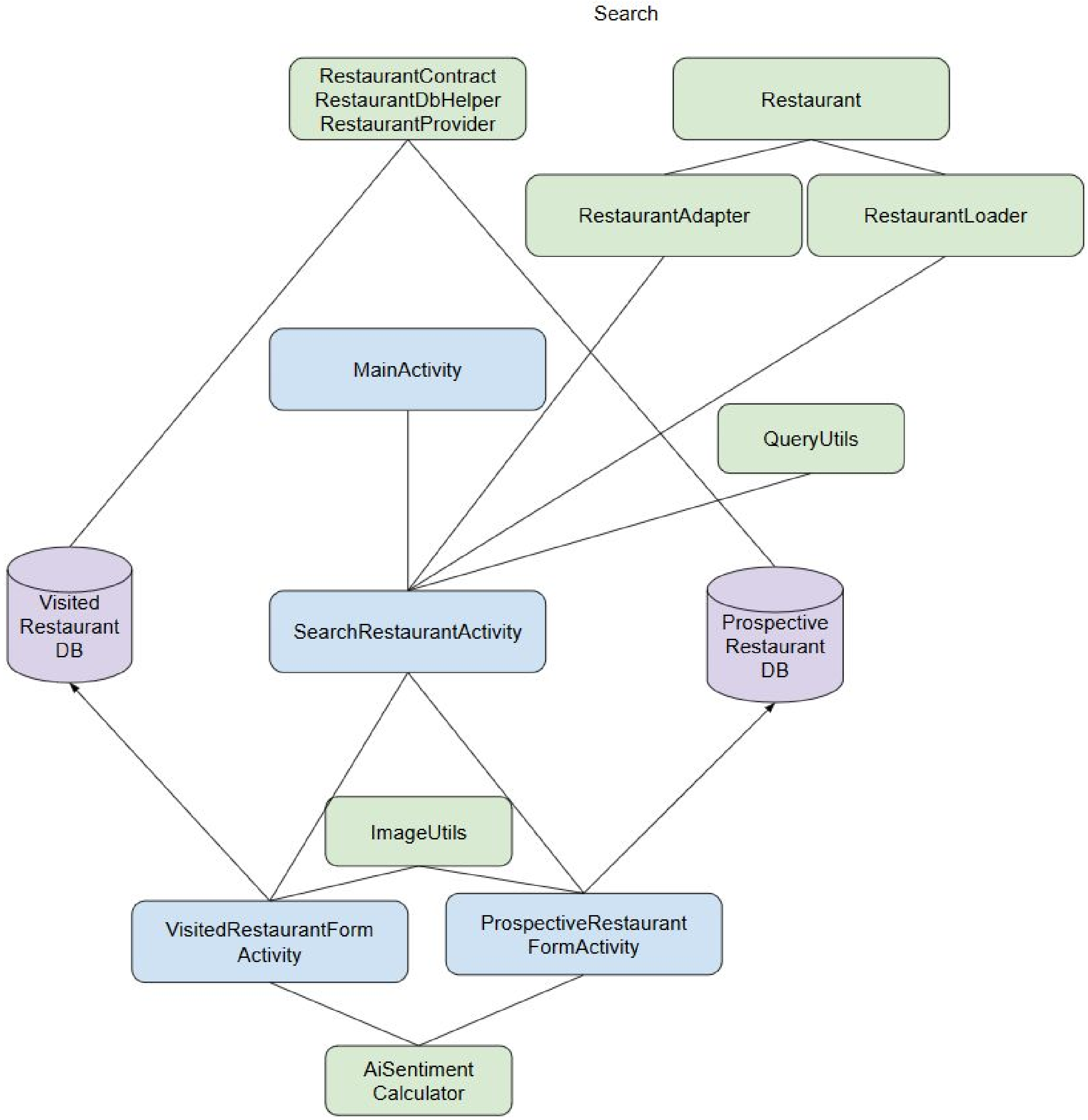
- Visited Restaurant Logic Flow:



- Prospective Restaurant Logic Flow:


RestaurantContract
RestaurantDbHelper
RestaurantProvider
Prospective
Prospective Restaurant DB
MainActivity
ProspectiveRestaurant CursoryAdapter
ProspectiveRestaurant Activity
ImageUtils
ProspectiveRestaurant FormActivity
AiSentiment Calculator

- Search Restaurant Logic Flow:



File structure: what does each file do and who wrote that file

- restaurantdiary/
  - Data/
    - RestaurantContract: Defines the contract for the restaurant in the database
      - Nate: Wrote the file
    - RestaurantDbHelper : Database helper for the restaurantdiary app. Manages database creation and version management.
      - Nate: Wrote the file
    - RestaurantProvider : The implementation of CRUD for the database
      - Nate: Wrote the file
  - Utils/
    - ImageUtils: Helps to store and retrieve images from the database
      - Nate: Wrote the file
    - QueryUtils: Helps the app interact with the Yelp api
      - Dave: Wrote the file
  - AiSentimentCalculator: This class uses the Watson AI Api to take and analyze a restaurant note and return how positive/negative the rating was.
    - Jake: Wrote the file
  - MainActivity: Where you navigate to the rest of the app
    - Rob: Wrote the file
  - ProspectiveRestaurantActivity: Displays restaurants that you want to go to and lets you select a restaurant to edit and add new ones
    - Jake: Hooked up the cursorAdapter, made the save async so Ai did not crash app by stalling the thread/or have race condition
    - Rob: Wrote stubs for all the buttons and menus
    - Nate: Added a way to insert a dummy restaurant item for testing purposes.
  - ProspectiveRestaurantCursoryAdapter: Logic that help display the data in the database in the ProspectiveRestaurantActivity
    - Jake: Wrote the file
  - ProspectiveRestaurantFormActivity: This is where user add and edit restaurants to the Prospective Restaurant database
    - Jake: Hooked up the cursorAdapter, made the save async so Ai did not crash app by stalling the thread/or have race condition
    - Rob: Wrote stubs for all the buttons and menus
    - Nate: Added a way for the form to interact with the database, as well as adding validation so user does not exit out without saving accidentally.
    - Dave: Added a way for the form to interact with the Yelp Data.
  - Restaurant: Wrote this in store restaurant temporarily for displaying them in the search activity
    - Jake: Wrote the file

    - RestaurantAdapter: This was made to help display resturants in the search activity
        - Jake: Wrote the file
    - RestaurantLoader: This was made to help display resturants in the search activity
        - Jake: Wrote the file
    - SearchRestaurantActivity: This is where users will search for new or restaurants they have been to using the Yelp Api. They can add them to the appropriate database
        - Dave: Wrote all the search logic
        - Jake: wrote some dummy data to allow for data formatting earlier
    - VisitedRestaurantActivity: Displays restaurants that you have gone to and let you select a restaurant to edit and add new ones
        - Jake: Hooked up the cursorAdapter, made the save async so Ai did not crash app by stalling the thread
        - Rob: Wrote stubs for all the buttons and menus
        - Nate: Added a way to insert a dummy restaurant item for testing purposes.
    - VisitedRestaurantCursoryAdapter: Logic that help display the data in the database in the ProspectiveRestaurantActivity
        - Jake: Wrote the file
    - VisitedRestaurantFormActivity This is where user add and edit restaurants to the Prospective Restaurant database
        - Jake: Hooked up the cursorAdapter, made the save async so Ai did not crash app by stalling the thread
        - Rob: Wrote stubs for all the buttons and menus
        - Nate: Added a way for the form to interact with the database, as well as adding validation so user does not exit out without saving accidentally.
        - Dave: Added a way for the form to interact with the Yelp Data.
- Layout: xml files for activity and list items
    - Dave: Wrote some layouts for SearchRestaurantActivity
    - Jake: wrote some temp layout to display dummy data
    - Rob: Did most of the layouts
    - Nate: Edit some layout to get images in right places
- Styles: Style of the app
    - Rob: Wrote the file
- Values: This is where you store constants
    - Dave, Jake, Rob, Nate:

Extra Credit: IBM’s Watson Natural Language Understanding

- We use the API to analyze the note that the user makes by giving it a rating based on the score the Watson gives it. It takes this score to see how positive the user’s review of the restaurant was, and displays either a smiley face, neutral, or frowning face in the restaurant listing.