Team Rocket Design Document

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CSC301 Winter 2019 |  |  |  | University of Toronto Missisauga |

Contents

1. [CRC Cards 1](#_Toc804560)
2. [System Interaction with Environment 4](#_Toc804561)
3. [System Architecture 5](#_Toc804562)
4. [System Decomposition 5](#_Toc804563)

# CRC Cards

Class name: MainActivity

Parent class: AppCompactActivity

Subclasses: None

Responsibilities:

* Direct user to login or signup pages

Collaborators:

* LoginActivity
* SignupActivity

Class name: SignupActivity

Parent class: AppCompactActivity

Subclasses: None

Responsibilities:

* Create the user’s account after they have filled in the registration form correctly

Collaborators:

* MainActivity
* UserProfileActivity

Class name: LoginActivity

Parent class: AppCompactActivity

Subclasses: None

Responsibilities:

* Authenticate, and log in the user upon filling in correct sign in information

Collaborators:

* MainActivity

Class name: EventActivity

Parent class: AppCompactActivity

Subclasses: None

Responsibilities:

* Displays information regarding an event.

Collaborators:

* EventIndexActivity

Class name: EventIndexActivity

Parent class: AppCompactActivity

Subclasses: None

Responsibilities:

* Displays an index of all the events in the local area

Collaborators:

* EventCreateActivity

Class name: EventCreateActivity

Parent class: AppCompactActivity

Subclasses: None

Responsibilities:

* Displays the view for event creation.

Collaborators: None

Class name: UserProperties

Parent class: AppCompactActivity

Subclasses: None

Responsibilities:

* Helper class used to save user information into the database

Collaborators: None

Class name: UserProfileActivity

Parent class: None

Subclasses: None

Responsibilities:

* Pulls data from the database to show a user’s profile.

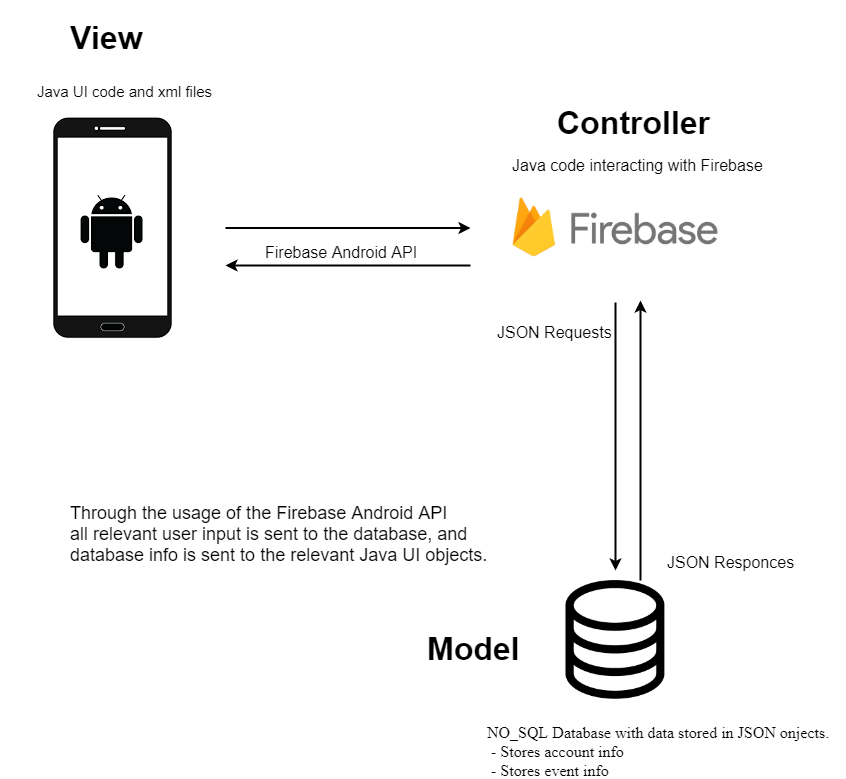
Collaborators:

* UserProperties
* LoginActivity

# System Interaction with Environment

For our project we will mainly be using Java for the front end and UI. For the rest we will be using Firebase which will handle our server and database (in no-sql) and API. It will also allow data synchronization across devices. Our project is designed to run on Android devices running the Android OS which is well integrated with Java and Firebase, allowing for UI scaling for different android devices.

# System Architecture



# System Decomposition

The main classes that interact with the firebase database are SignupActivity, LoginActivity and UserProfileActivity which add information to the database and query information from the database through use of the firebase API, in the form of adding and validating user information. Both of these classes are set up display errors upon invalid or empty user input, should the user try to login or sign up with invalid information. The class UserProperties is also used in the database to easily store the user’s information in the database as a single object.

The view components of the app are xml files that are connected to their relevant controller classes, for example the activity\_signup.xml file is connected to the SignupActivity class which acts as a controller, that will react to user input from that view.

Our general strategy for dealing with invalid user input will be to display an error message to the user, we will also implement layers of validation to prevent any unexpected input from entering the backend.

Since our app will require network connectivity to fully function only some portions of the app will be able to function properly should connection issues arrive, we will also display an error message to the user informing them of their connection issues.