# Design Document for <PickLyfe>

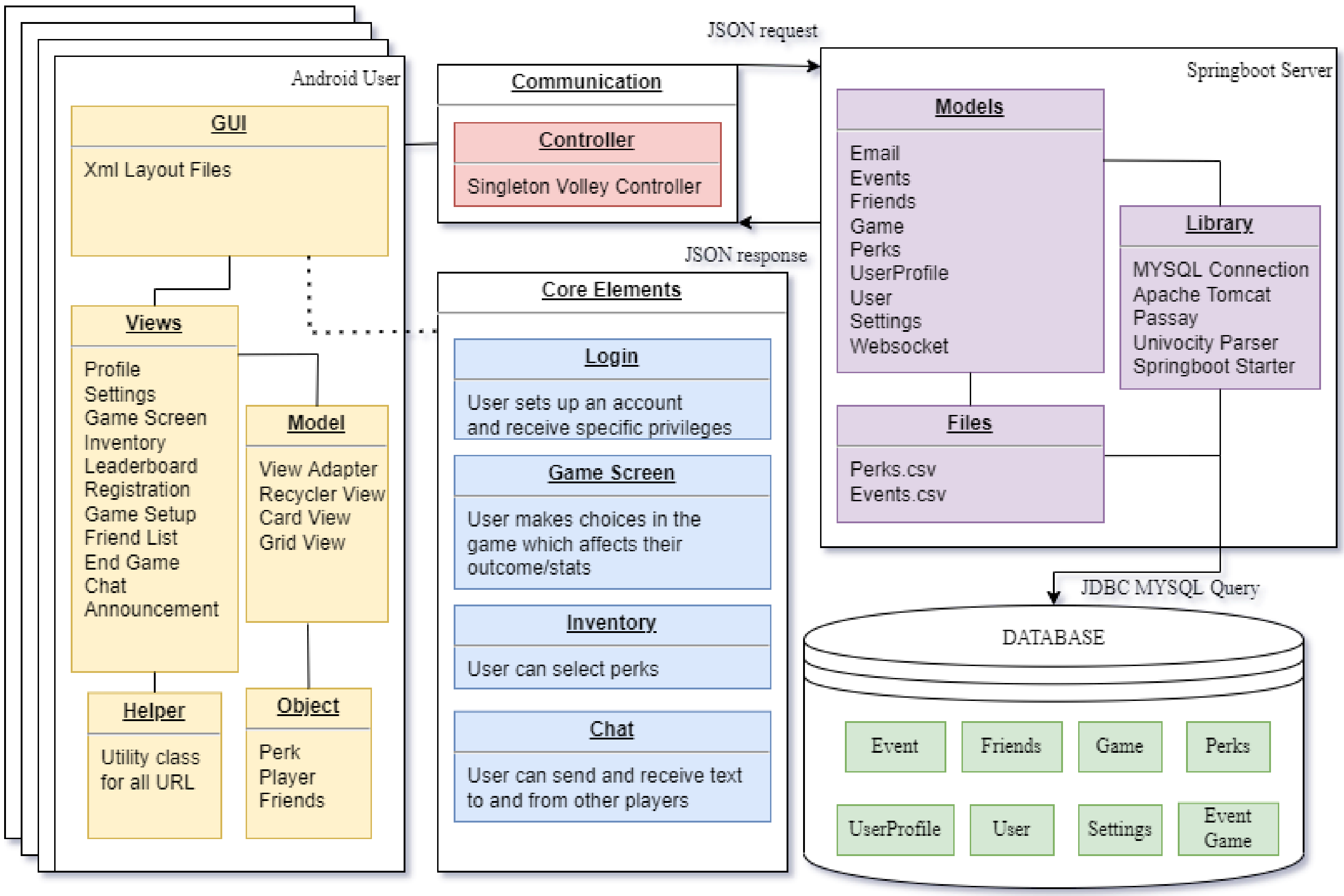
**Team:** 1_CW_1

**Team Members:**

Andy Foo
Chen Yu Goh
Roy Khoo
Yin Hao Choong

**Project Name:** PickLyfe

# Block Diagram



# Design Descriptions

## Android User GUI

In the project, we will have view thread for all classes that would initialize all elements needed in the activity. Our android application has 9 screens as of now and we plan to have 2-3 additional screens (chats, announcements, etc). Each activity will have their own design and layouts reflected by their corresponding .xml files. More complicated layouts and UI elements would have their own additional xml drawable files integrated with their own .xml files (for buttons, borders, card adapters and recyclerviews). Activity classes handle most of the user interactions and CRUD operations to and from the backend. The intent module is used extensively throughout the application for switching between activities/screens.

## Android Utility Classes

For the purpose of simplifying urls and clarifying specific urls to fetch/post data from the backend, we made a utility class to hold url addresses. The addresses are all initialized to their respective static variables, allowing other classes / activities in the program to access these variables with ease. Changes and addition of specific urls within this module reduces the need of reupdating the paths of urls via a repetitive, error-prone procedure.

## Communication

In the essence of modular programming, a good system design is to separate elements of a program's process into different 'stages', such that any part is replaceable, scalable and cohesive with the rest of the application. Thus in pursuit of this, the communication model act as the middleman in transmitting and receiving data from the Backend. Subsequently, server request from the client are handled and procedural queued and funneled through the singleton volley controller. All interface applicable and interactable to the client would have its corresponding request commands handled by this class. This includes and is not limited to the game screen, inventory, userprofile and leaderboard. Scope of the communication class includes JSON, JSONArray, string and image request and operates on a single-threaded manner.

## Core Elements

This covers the main core features of the application ranging from the login page to the game screen to inventory and chat screen.

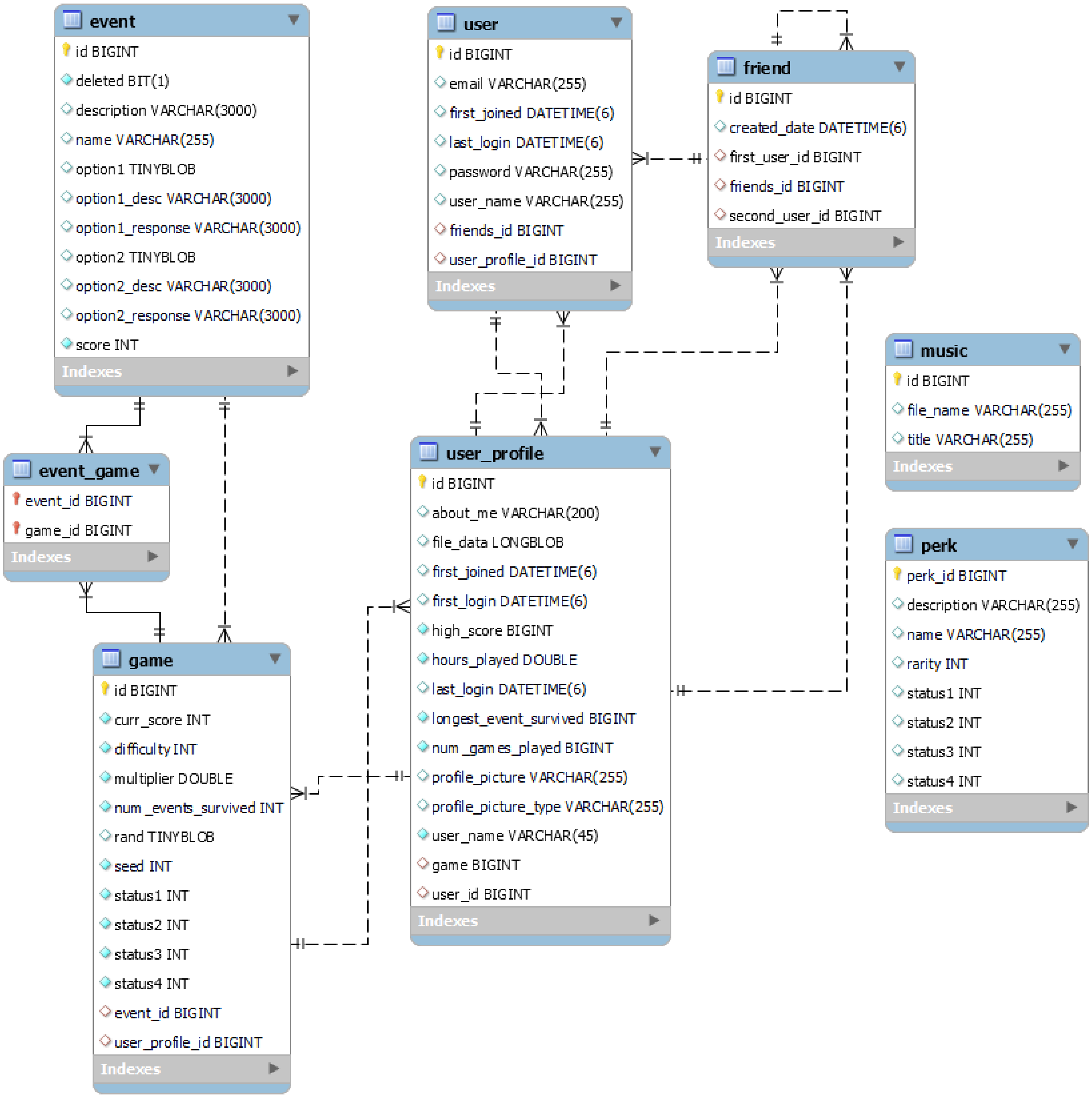
## Server

The main server receives the JSON request from the android client side to process the information. The JSON request is sent for processing in the respective models. The server uses multiple library to manipulate the data stored. Files implemented in the server are parsed and will change the processed JSON. The JSON is then stored in the MYSQL server base. The server sends back the response to the android client side after.

## Database

All data produced or edited by the server is stored in the database. The database contains all current user information and can be accessed by volley requests from the frontend to view it. Events and Perks data are stored automatically on start-up and no JSON request will change it. The database will be updated upon every JSON request sent to the server.

# DATABASE SCHEMA


event
id BIGINT
deleted BIT(1)
description VARCHAR(3000)
name VARCHAR(255)
option1 TINYBLOB
option1_desc VARCHAR(3000)
option1_response VARCHAR(3000)
option2 TINYBLOB
option2_desc VARCHAR(3000)
option2_response VARCHAR(3000)
score INT
Indexes
user
id BIGINT
email VARCHAR(255)
first_joined DATETIME(6)
last_login DATETIME(6)
password VARCHAR(255)
user_name VARCHAR(255)
friends_id BIGINT
user_profile_id BIGINT
Indexes
friend
id BIGINT
created_date DATETIME(6)
first_user_id BIGINT
friends_id BIGINT
second_user_id BIGINT
Indexes
music
id BIGINT
file_name VARCHAR(255)
title VARCHAR(255)
Indexes
event_game
event_id BIGINT
game_id BIGINT
Indexes
user_profile
id BIGINT
about_me VARCHAR(200)
file_data LONGBLOB
first_joined DATETIME(6)
first_login DATETIME(6)
high_score BIGINT
hours_played DOUBLE
last_login DATETIME(6)
longest_event_survived BIGINT
num_games_played BIGINT
profile_picture VARCHAR(255)
profile_picture_type VARCHAR(255)
user_name VARCHAR(45)
game BIGINT
user_id BIGINT
Indexes
perk
perk_id BIGINT
description VARCHAR(255)
name VARCHAR(255)
rarity INT
status1 INT
status2 INT
status3 INT
status4 INT
Indexes
game
id BIGINT
curr_score INT
difficulty INT
multiplier DOUBLE
num_events_survived INT
rand TINYBLOB
seed INT
status1 INT
status2 INT
status3 INT
status4 INT
event_id BIGINT
user_profile_id BIGINT
Indexes