# StudyBuddy
# Block Diagram

Team: VB_5

Team Members
Khanh Tran
Wei Lim
Evelyn Khew
Varuna Rao

## Design Description

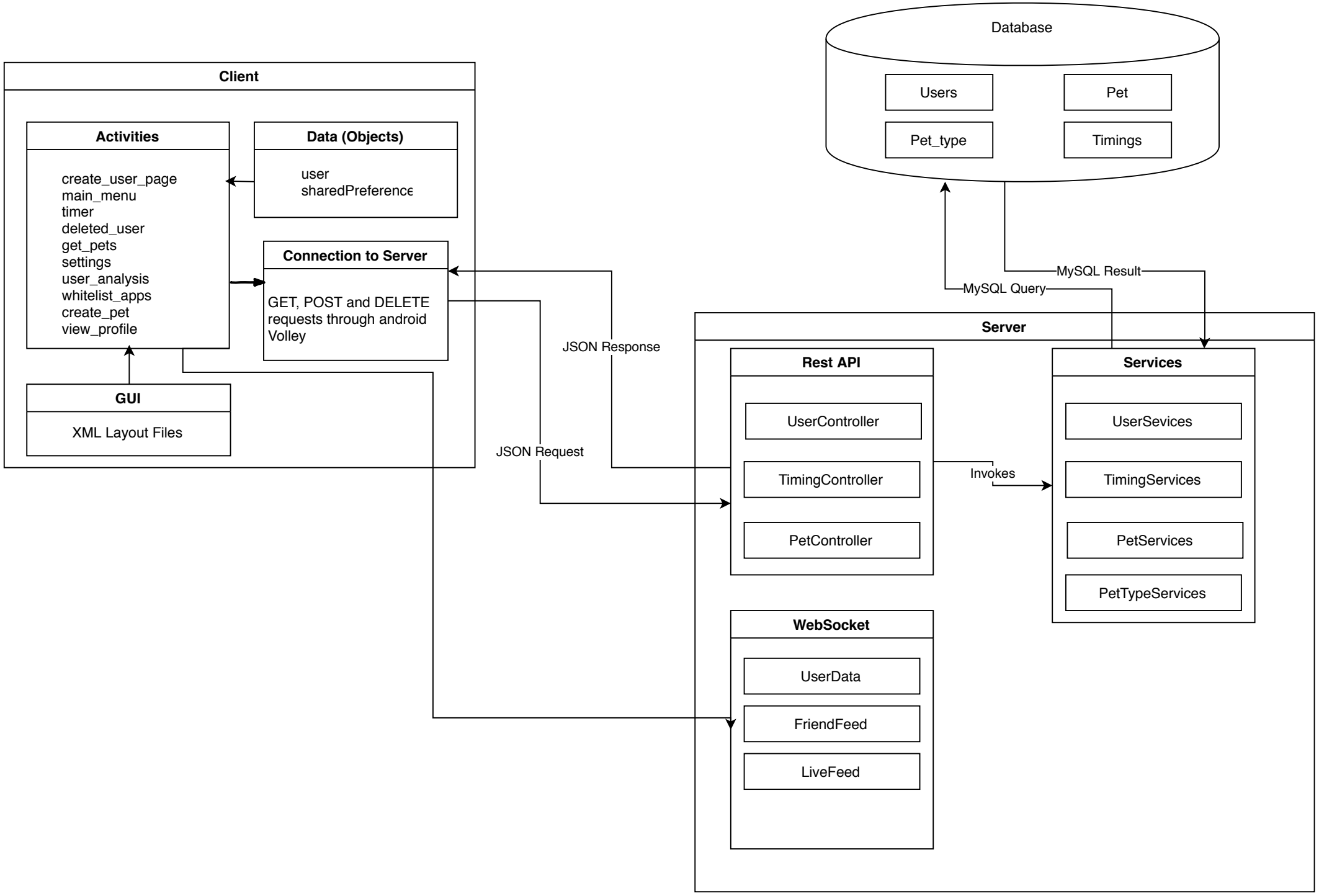
The client side for our project, StudyBuddy, uses android studio as a base. The main GUI for the client user uses XML code. Within our app, there are different activity classes for each individual app page. There are different types of data objects, user and sharedPreference; we use sharedPreference for fetching data from clients using the app, such as amounts of tickets and pet amounts. We also included an app page which will include the statistics of the user study data, which will show how long they have been studying and the user's highest study record and so on. Lastly, we use HTTP requests and JSON objects to send information from our client to our server; information like stopwatch times for a user, tickets, different types of pets, and pet names.

The server allows connection from the RestAPI to the services on the client side. Our database is running with MySQL database and we access it through JDBC MySQL Query. Server is working independently from the client side and the server will receive or send some data whenever a certain button is clicked on the client side.

The table on the database contains all the user's study data which includes the user's study time, available pet, and what type of pet they have.

<table>
<tr><th>Table</th><th>Field</th></tr>
<tr><td>Pet</td><td>
<table>
<tr><td>ID INT(11)</td><td>Pet ID</td></tr>
<tr><td>NAME VARCHAR(255)</td><td>Name of the Pet</td></tr>
<tr><td>TYPE VARCHAR(255)</td><td>Type of the pet</td></tr>
<tr><td>OWNER_ID VARCHAR(255)</td><td>Owner’s ID</td></tr>
<tr><td>OWNER_USERNAME VARCHAR(255)</td><td>Owner’s username</td></tr>
</table>
</td></tr>
<tr><td>PetType</td><td>
<table>
<tr><td>ID INT(11)</td><td>ID of the pet type</td></tr>
<tr><td>RARITY VARCHAR(255)</td><td>Rarity of the pet</td></tr>
<tr><td>SUB_TYPE VARCHAR(255)</td><td>Subtype of the pet</td></tr>
<tr><td>TYPE VARCHAR(255)</td><td>Type of the pet</td></tr>
</table>
</td></tr>
<tr><td>Users</td><td>
<table>
<tr><td>USERNAME VARCHAR(255)</td><td>Username of user</td></tr>
<tr><td>PASSWORD INT(11)</td><td>Password of user</td></tr>
<tr><td>SALT VARCHAR(255)</td><td>Random generated string</td></tr>
<tr><td>TICKETS INT(11)</td><td>Amount of ticket user has</td></tr>
<tr><td>USED_TICKETS INT(11)</td><td>Amount of ticket user used</td></tr>
</table>
</td></tr>
<tr><td>Timings</td><td>
<table>
<tr><td>ID INT(11)</td><td>ID of each timing</td></tr>
<tr><td>END_TIME DATETIME</td><td>End time of each timing</td></tr>
<tr><td>START_TIME DATETIME</td><td>Start time of each timing</td></tr>
<tr><td>OWNER_USERNAME</td><td>Username of the timing owner</td></tr>
</table>
</td></tr>
</table>

Users to pet is a many to many relationship.
Users to timings is one to many relationship
Pet to pet_type is many to one relationship


Database
Users
Pet
Pet_type
Timings
Client
Activities
create_user_page
main_menu
timer
deleted_user
get_pets
settings
user_analysis
whitelist_apps
create_pet
view_profile
Data (Objects)
user
sharedPreference
Connection to Server
GET, POST and DELETE requests through android Volley
GUI
XML Layout Files
JSON Response
JSON Request
MySQL Result
MySQL Query
Server
Rest API
UserController
TimingController
PetController
Invokes
Services
UserSevices
TimingServices
PetServices
PetTypeServices
WebSocket
UserData
FriendFeed
LiveFeed