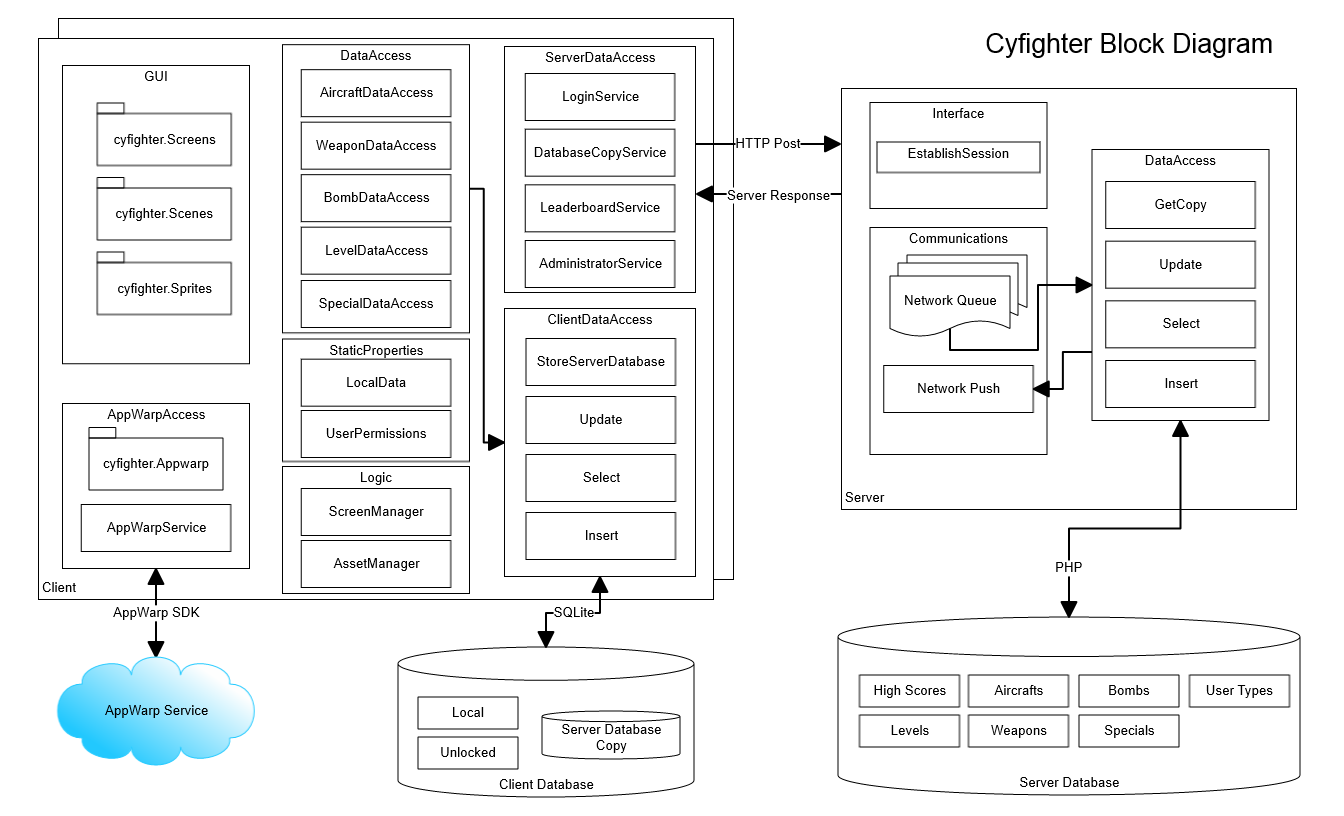
Block Diagram

Team 18

Brendan McCluer  
Zachariah Brase  
Edwin Benggawan  
Yangxiao Wang

**Cyfighter**



The Cyfighter project is broken into five components: Client, Server, Server Database, Client Database, and AppWarp Service. Communications are described in the arrows connecting them. The client communicates with the server via HTTP posts. The server communicates with the server database using PHP. The client also communicates with the client database via SQlite and communicates with the AppWarp service via the AppWarp Java SDK. The components are further broken down into tiers.

**Server**The server contains three tiers: Interface, DataAccess, and Communications. The Interface tier establishes the user session requested by the client. The client sends a username and password in its HTTP post. The server then uses this information to log into the server database. The Communications tier holds all pending requests made from multiple clients while the DataAccess tier executes the requests. Notice that the GetCopy class returns a copy of the database to be stored in the client database. The server is written in PHP code.

**Client**

The client contains seven tiers and is written in Java code. There are multiple clients. The GUI, StaticProperties, and Logic tiers implement the LibGDX framework classes and are used for main game play. The AppWarp access tier uses the AppWarp SDK to communicate with the AppWarp service. This service is used to host multiplayer sessions. The DataAccess tier contains logic for interacting with the ClientDataAccess tier. It uses the ClientDataAccess Insert, Update, and Select classes to communicate with the client database. Notice that this tier also has a “StoreServerDatabase” class. This class calls the “DatabaseCopyService” in the ServerDataAccess to store a copy of the server database. The ServerDataAccess class then uses the GetCopy class from the server. At the start of the client system, this is called. This increases throughput as the majority of data accesses by the client will be internal and will not require posting to the server.

The ServerDataAccess tier also contains classes for viewing and posting to the leaderboard, logging into the server, and adjusting prices of weapons, aircrafts, and bombs (via the AdministratorService class). LeaderboardService uses the server to read and store high scores in the server database. Notice that high scores are stored on the server and should not be accessed locally. AdministratorService updates prices on the server database and the client database (as changes should be consistent on all client systems).

**AppWarp Service**

This service is a “black box” and handles all messages sent by two client threads communicating with each other.

**Server Database**

This database contains seven tables. The High Scores table is the only table not copied to the client database.

**Client Database**

This database contains 2 tables along with the tables copied from the server database. Local contains information on current settings and currency. Unlocked contains IDs of unlocked weapons, levels, aircrafts, and bombs.