## Architectural Descriptions

The monster program consist of three components:

- Database
- Client
- Servlets

### Database

We will be using Java Database which is a database management system (DBMS) that we use to stores and retrieves information in a database. It is used to organize the data according to the subject, and it is easy to track and verify the data, and it can store information about how different subjects are related, so that it makes it easy to bring related data together. We will create table such as the user's profile and their monster's information, login database and the friend's database. The database will implement to meet the requirement of (FR1),(FR2),(FR3),(FR6) and (FR11) of the requirement specification documents.

### Client

The client is where the user can request webpage from the server. Our program webpages will be consist of:

- Login page
  - This is where the users can login existing account or register a new account.
- Home page
  - The users will be able to choose various links to navigate.
- Marketplace page
  - This page allow users to view the monster that are to be bought or rent.
- My monster page
  - This page allow users to breed the monster own.
- Monster Fights page
  - The user will be able to challenge other monster through this page.
- Friends page
  - This page allow users to interact with other users.

The user will be able to communicate the server through POST and GET method. POST method are used to catch the user input field which is in the forms. GET request are used to tell a PHP file which webpage to display. The client will be implement to meet the requirement of (FR5),(FR6),(FR7),(FR8),(FR10),(FR11),(PR1) and (PR2) of the requirement specification documents.

Servlets

On the servlet side, we are using Glassfish servlets as development tool which provide storage and communication storage. The servlet is used to produce a response to the request made by user in the html page and send it back to the requesting browser through GET and POST request. The servlet first looks for incoming request data: if it finds none, it presents a blank form. If the servlet finds partial request data, it extracts the partial data, puts it back into the form, and marks the other fields as missing. If the servlet finds the full complement of required data, it process the request and displays the results. The servlets will be implement to meet the requirement of (FR1),(FR2),(FR3),(FR4),(FR5),(PR1),(PR2),(DC1) and (DC2) of the requirement specification documents.