Subscription based Game Database

Mitchell Gerber and Jason Carpenter

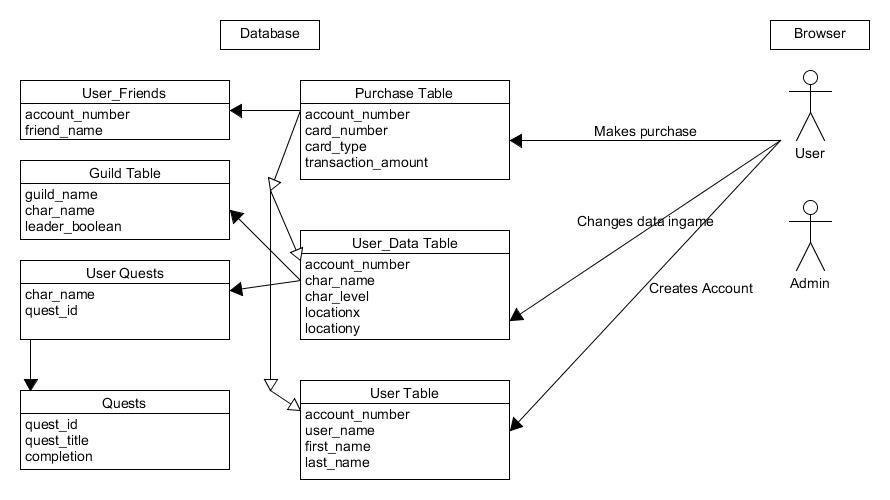
We will be creating a subscription service that is based off of an MMO style game. Systems like this retain and manipulate large amounts of user and developer data. To replicate this we will create a multi-table, multi-ui, and multi-data manipulating system. We plan to create a back end system that will utilize a MySQL database to store user and game information. We will also create a front end user interface to simulate actions within an actual game along will displaying user and game information. The user interface will also offer admin support to add additional content to the game system. It will store and hold user data, purchase data, and subscription statuses. The source of our data will be generated examples done by hand as most of the data is user specific.

Five data tables will be utilized first of which is a User Data table, which will contain Name, Username, and last-name and person specific account information. There will be a purchase database with credit card info, subscription length remaining, and purchase success. There will be a user service data, which will hold in avatar information, location, and game data. We will also utilize a guilds table which will contain player association lists and a friend’s list that will also work in conjunction with the user. We will use a one to many mapping from the User table to the friends list table because the user is going to be able to have multiple friends. There will be a quest data table which will have quest information, location in world, and various other details. This will be a one to many mapping to the various user’s quest tables which contain the quest reference. The user interface will have a quest log that will allow completion of quests from the page. This will manifest as a Boolean indicator in the quests table. We may add or remove tables when necessary during the development of this project.

The application will consist of several user interface pages. The welcome page will be a login page if the user is not logged in. We will make use of sessions and possibly cookies to keep track if a user is logged in or not. If they are already logged in the welcome page will display user information. Upon login the user will be presented with a user interface that displays the combined data from the user table, purchase table, and guild table. A basic logout and register page will also be included. A user will login and be able to access their purchases, user game data, and generic user data. If they are not registered they will be prompted with a register page. Admin’s will have authority to add game content, remove users, and manage payment data.

Below is a rough diagram of tables we plan on using in our project. We have two types of accounts which are admin and user. The admin account is used for adding content to the game such as quests. They will also be able to adjust user information to offer player support. Users will be able to create an account, attach a credit card to start a subscription, add friends, join a guild, and complete quests. This is a rough estimate of the use cases we plan on implementing in our project but they may be adjusted. We are going to utilize a one to many mapping for our user friends table. Each user will be able to have a certain amount of friends, which is unlimited at this time. Users will also be able to have multiple quests so we will use one to many mapping in that case as well.

The objective of our project is to create a database to organize information within a subscription style game. Most games today have massive amounts of player information that has to be stored somewhere. The majority of these games make use of a relational database because it is reliable, easy to access, and can handle a massive amount of requests simultaneously. The relational database is also useful for admin’s to mass update or change data for the game or the users. We are going to develop a web app to simulate an actual game while focusing the majority of our time on back end database design.



Tiered design:

1. The front end of our application is going to be web based. We are going to be designing our project with HTML and CSS.

2. The logic of our application is all handled in Java and JSP. Java servlet is going to act as the controller which will serve JSP pages for each request.

3. Our data access layer is going to use Java and will utilize JDBC to connect to a MySQL database.