MMO FRONT-END AND BACK-END DATABASES FOR SMALL GAME DEVELOPERS

John Matthews Mudasser F. Wyne

School of Engineering, Technology and Media
Technology and Health Sciences Center
National University, San Diego, USA

Abstract

Among the most difficult issues in dealing with the online gaming and virtual world markets are the costs involved in capturing subscriber's payment information and the methods used to circumvent user unauthorized modification of player. Most of the game space or virtual world developers understandably focus on their content development, and have little or no interest or experience in the creation of the business required structures to host and run their creations.While there are several companies which do provide excellent business front end (account management, user financial records and transactions, etc.), they do so at a high cost. If a game or virtual world is popular, the hosting requirements for both the servers and the account management increase radically. In this paper we presentMMO (Massive Multiplayer Online) Front-End that is designed to be a one time,low cost application to solve the account management and billing issues while still offer the security and adaptability of a high priced solution. We also designed a low cost series of the three most significant backend service databases to accommodate the smaller developer budget constraints. The prototype of the system has been implemented and technically reviewed by a group of users.

KEYWORDS: Microsoft Structured Query Language, Active Server Pages, Multi-User Dungeons

1. INTRODUCTION

The Massively Multiplayer Online (MMO) market has been growing at a phenomenal rate since the first versions began as MUDs (Multi-User Dungeons) in the late 1980s [1, 2, 3]. It is estimated that 67% of the American household play video games, and almost half of these participate in some form of online game or virtual environment [4]therefore the potential for a profit market is almost unlimited, especially for supplying the business backend services to game and virtual world content developers.

Currently, the available services for these types of online environments fall into two categories: Those geared toward the needs of the large publishers and developers whose online project budgets are into the range of 20 million dollars or more (known as "AAA" projects as best defined by [5]) having license solution that may cost thousands of dollars per year; and, those whose projects fall within 10+ million dollars (known as "AA" projects) which are generally marketed in a lump sum license. Low budget, small developer or independent "Indy" projects. These developers, generally, do not have the money to license third party backend services. The majority of these games or virtual worlds are released for free on hosting services which attempt to create income through banner advertising, sale of virtual goods, or donation payments.

Most developers of games and virtual worlds fall within the small to medium business category; as such they do not often have the staffing and production budget, or the desire, to develop the backend services for their IPs (Intellectual Properties) on their

own. Most often they do not even develop the middleware to articulate their content, or that used to create the content itself. They rely on third parties to develop middleware and creation toolsets for content, and game world physics and data "engines" to render and display the content interactively for the user. Thesedevelopers additionally rely on a third party provider to license backend services. Four publisher/developers have been able to successful accomplish "AAA" level in-house backend development: EA (Electronic Arts) with Ultima Online, Blizzard/Activision with World of Warcraft, THQ with Warhammer Online, and Square-Enix with Lineage and Lineage 2. All others would have had to license outside third party solutions.In contrast, for the smaller developer products "MMORPG Toplist" [6] hosting site links to 33,205 game space servers.This shows that there exists a large potential market of developers who may be interested in the availability of a low costsolutionsthat can be easily tailored to a series of service modules to accommodateany of the required "AAA" functionality.

2. ARCHITECTURE

In the earlier stages it was thought that running a lightweight interpreter program written in the Erlang language on a common Linux kernel would be the best solution to support the SQL database applications themselves. However, even with the appropriate hardware configuration, there is little performance gained as compared to using a native Microsoft operating environment (Server 2008) to run the MS SQL application environment. It is therefore suggested to use the server configuration, such as: Multi-core processor (Quad core or better) with at least two Gigabytes of dedicated RAM for each processing core; 500 Megabytes of Hard Drive space (or greater); Black Fibre optics connection if cluster configuration is going to be implemented. If server is configured as stand-alone (low volume of transactions expected) then a minimum of gigabit Ethernet should be used for connectivity to the communications device (server or switch/gateway). The choice to develop in MS 2008 is based on the fact that the Express version is available at no cost, and can run the required application with high performance expectation, and also allows multiple "front-end" applications to be developed by the licensee in ASP.NET (C# or Visual Basic), C++, HTTP v.5, or even Adobe FLASH Action Script. An additional consideration is that MS SQL Server can be supported on Linux OS servers as well.

Data security is addressed through three techniques: All user character (and game) data will be stored as Hard Binary; All accounts are assigned unique identifiers using SQLidentity procedural process; and all personal and financial account data is encrypted as well as stored Hard Binary. Required data is retrieved and verified by data string matching.

There are three significant database modules to the customer logon and data capture processes. The system is primarily created in SQL,The MMO Front-end is being developed as a pilot project with the potential of being released by NNG (No Name Games), LLC.

3. BACK-END DATABASE DESIGN

Since this Massive Multiplayer Online (MMO) front-end is being developed in MS SQL (Microsoft Structured Query Language) 2008 R2 with the Microsoft SQL Server Management Studio as well as Visual Studio 2008, there is a high level of confidence that the entire database design,

including all tables, is normalized, with no repeating groups of data and data attributes are fully and directly dependent upon the primary key.

In an effort to create the required backend database structure and services for use in MMO environments, whether it be an online game (such as "World of Warcraft™"), or aalternative reality world environment (such as "Second Life"™), we have designed three lightweight database modules. These modules are developed in MS SQL. The modules will interact with each other in order to create and verify user accounts, as well as facilitate user login and payment criteria. A series of demonstration browser enabled interfaces (discussed in the following section) developed in the ASP.NET (Active Server Pages) framework are included with the database application to either be used as default interface pages, or as templates for the game developer to customize them for their own set of requirements.

We have designed the database consisting of various tables. Table User includes all the fields needed for any financial transaction.Amongother fields there is a field needed to create a bankcard debt or may be franking a check (negotiating electronically). A "game card" is a specific debt card for that particular game or stable of games if issued by a game publisher (such as Microsoft LIVE ™). Fields like EndDate and AutoPayare used to allow for auto payments from user's active checking account. The User table also includes important Security Identification (SID) components of the user's selected username and password combination. We have also included email, security question along with user entered answer that allows a reasonable security protectionforretrieval of forgotten password retrieval service.

Figure 1shows the database design associated with the MMO Character database. Note that table NextIdentity does not have a direct relationship as it is a preprocessor function running ahead of the database to vet character comparison requests and control SID error.We have also designed a character master table for MMO charactersthat includes most of the fields that would be found in a modern (abet, generic) avatar for an online game or virtual world environment.

Table Shield includes the information for defensive shield types, including the graphic file information (both thumbnail and full sized picture) and item value degradation. As this is a generic character table, it should be noted that shield and weapon can refer to any era.Table CharacterShieldcreate instances for the shield from player inventory.Table InventoryPermanentallows access to player inventory.Table Beard allows for character generation of facial hair.Table HairColorallows player to choose hair color on avatar generation. Table Torso allows for clothing or armor to be equipped and displayed in this body area.Similarly there are tables to choose eyebrow style, color and type as well as ear and its type. HairStyleallows player to choose hair style on avatar generation. In short there is a table for every part of the body and its various aspects.

Weapon table allows for the character to be equipped with a weapon. The item value degradation is essentially the same as for shields. In order to allow players to choose a secondary color on avatar generation we have SkinColorSecondary table. Depending upon the game or virtual world this may be an important characteristic.

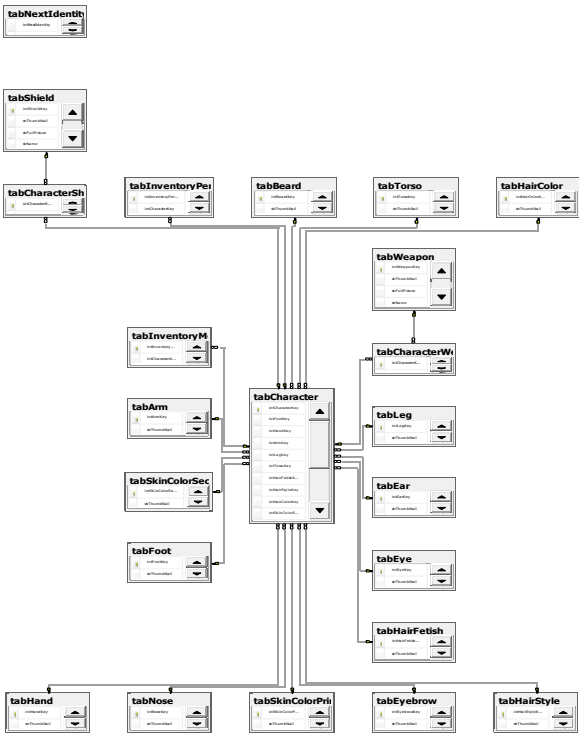

Figure 1: E-R Diagram MMOCharacter database

4. HIGH LEVEL FUNCTIONALITY

This section describesa high level functionality ofsome components of the system, with specific details. Figure 2 shows the player user logon process. The administration logon process is virtually identical, barring the difference of the second and subsequent pages of content. Administrators would go to the administration pages, while the players would be directed either to the player's account page or directly to a server lobby. The game developer using this system can themselves chose the content of pages. Figure 3 shows the logical process of the authentication cycle. This also is virtually identical for both the end-users (game or virtual world clients) and the administrators.

Figure 4 shows the entire logon sequence in a dialogue diagram. All administrative functions are accessed from the administration page, which loads immediately after an authenticated logon. The Administrative page is a link (menu page) loading immediately after logon, and allowing access to the user functions from that point.

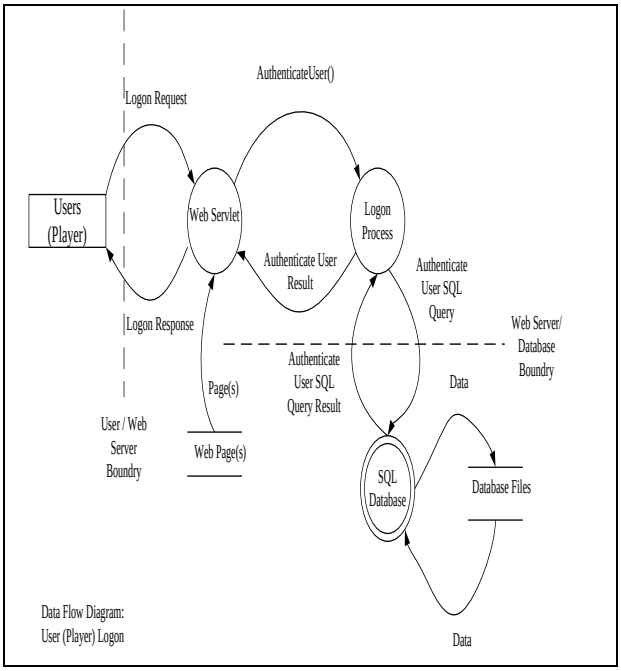

Figure 2: User Logon

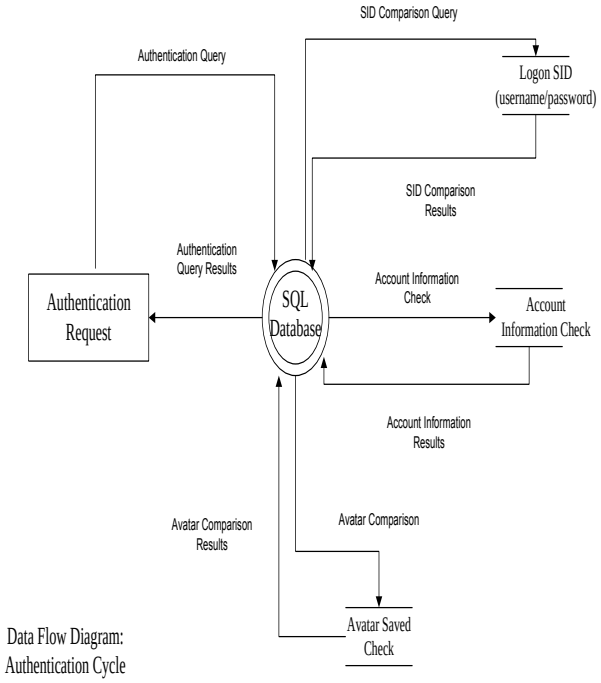

Figure 3: Authentication Cycle

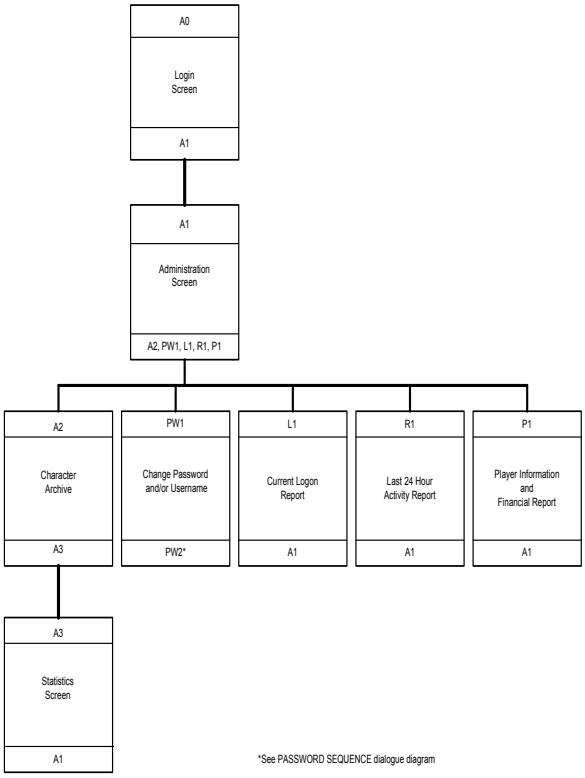

Figure 4: Logon dialogue diagram

Figure 5 shows a sequence for changing password as a dialogue diagram for the administrator access that allows functionalities such as, account password or username change. This default may also be used for the end-users as well, depending upon the individual developer requirements. The password and username change fields, the email contact information as well as the security question and answer for password retrieval service. The required password criteria are listed underneath the "Save" and "Abort" button triggers. The reason for the required acknowledgement is protection against password reset utilities by forcing a user action to commit the change.

5. THE PROTOTYPE SYSTEM

In order to create a testing environment a series of webpages are created as a best way to interface with the database that has been developed. In order to facilitate this we used rapid prototype development model, three web pages (figures 6, 7 and 8) were created in Visual Studio 2008 using ASP.NET. These pages are expected to communicate successfully with the respective database (tables). The first page, a logon page as shown in figure6, includes holding ART, which is in public domain. The logon page uses a simple username and password field configuration with a logon button trigger which can be articulated by either pointer device or by merely hitting the "Enter" keystroke. The functionality of the program is terminated by the return to logon, and thus user must logon again to achieve any degree of function.

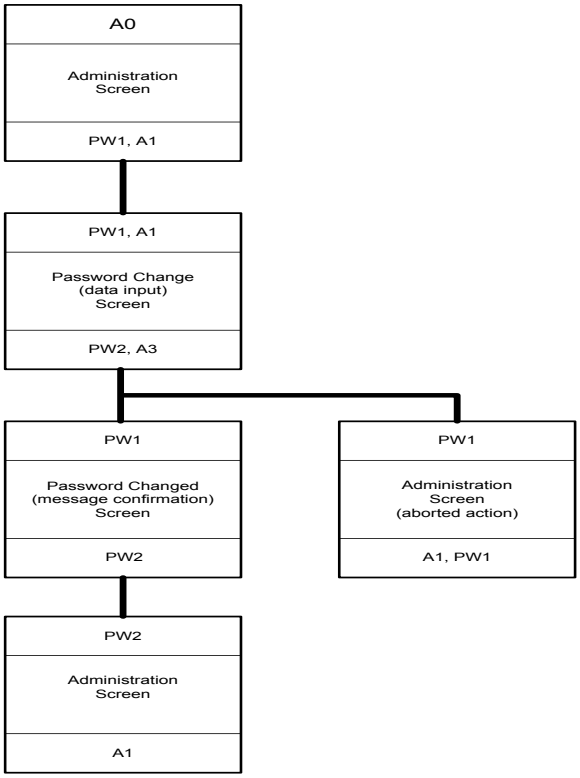

Figure 5: Administration dialogue diagram.

Figure 7 shows the administrator account management page. The "Exit" button returns the user to the logon page where the browser can simply be closed. The functionality of the program has been terminated by the return to logon, and the user must logon

again to achieve any degree of function. We will now discuss each of the options available on this page in detail

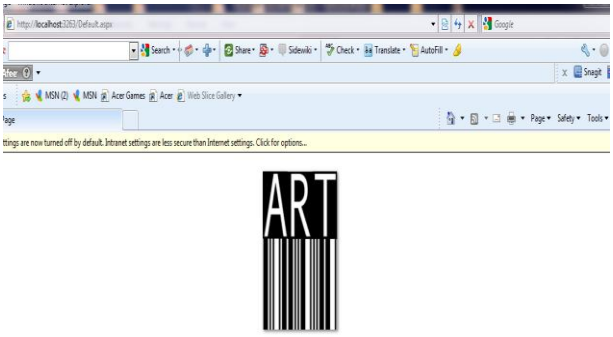

Figure 6: Logon interface page

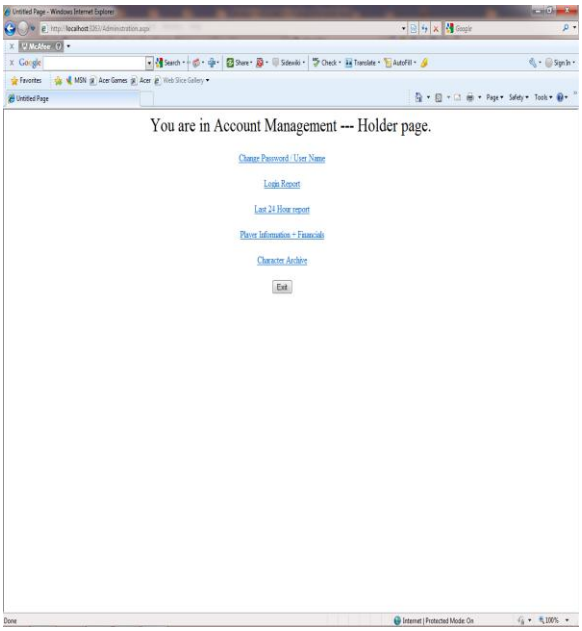

Figure 7: Administrator Account Management page

Figure 8 shows the password and username change fields, the email contact information as well as the security question and answer for password retrieval service. The required password criteria are listed underneath the "Save" and "Abort" button triggers. Once the change is saved the success message page for a completed password/username change will popup. This message must be acknowledged by triggering the "Done" button, which will then return the user to the administrator account management page. The reason for this acknowledgement is to provide protection against password reset utilities by forcing a user action to commit the change.

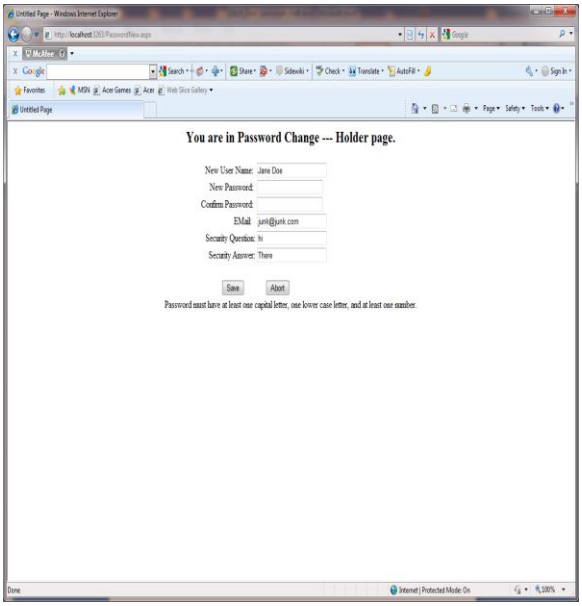

Figure 8: Password/Username change page

The Character Archive (figure 9)is loaded by default under the MMO Front-End original design. This character page is shown to demonstrate potential client customization. The application can load any flagged account data (in sequence by user number if multiple accounts).This page is also available (in ascending user account number ordination) and can be accessed directly from the administration account management page.

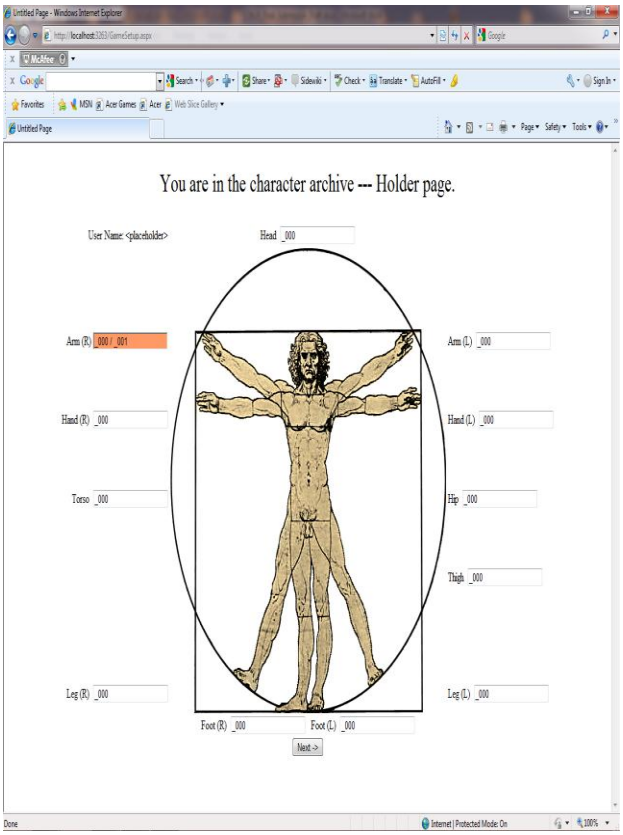

Figure 9: Character Archive page

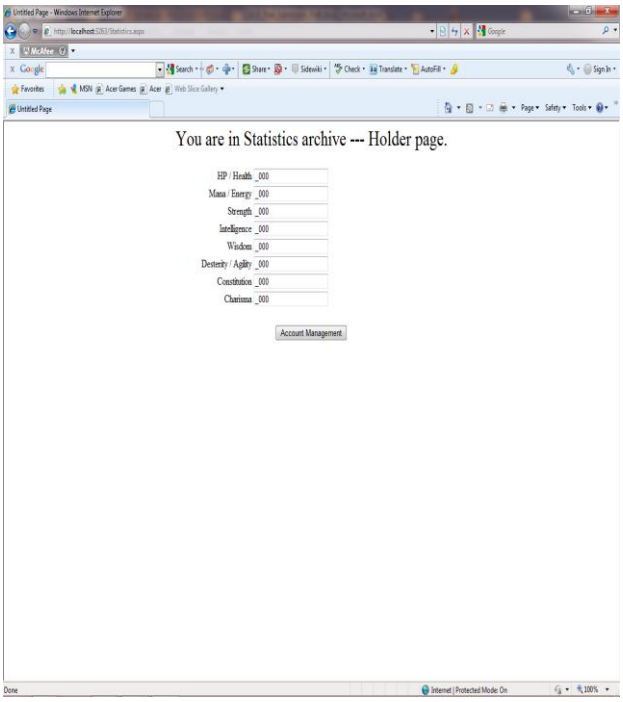

Figure 10:Player Statistics page

Figure 9shows the Player statistics page which is accessed from the character archive when a particular user account is either flagged for review or accessed directly by an administratorpage(figure 7). From this page the "Account Management"is accessed which will return the user to the administration page. This page is highly adaptable with 100 additional fields built into the default database design so the end-users can "collect" many additional game space or virtual items, or the content designer can have a great many skills for the avatar. This is entirely at the discretion of the content developer and their design vision.

The statistics (figure 10) page draws its data from the same character database as the character archive page (figure 9) shot. We have shown here as separate page for ease of data review by an administrator or game referee. Additional pages showing a generic player avatar generation (end user input for the character database tables) are included in the final release. It is the expectation that the game developer will create the product specific player interface pages, instead of using these default administration pages.

Figure 11 shows the System Logon report that includes administrator and end-user who logged onto the system being supported by the MMO Front-End application (the game space or virtual environment). However, it does not show logoff time or session duration length. The information about accounts (administrator and end-user) on the system (game space or virtual world) for the last twenty-four hour period, for reporting purposes is also available. This information can be used for traffic monitoring and load balancing on multiple servers or game shards, as well as for security documentation. However, it also

does not show logoff time or session duration length.

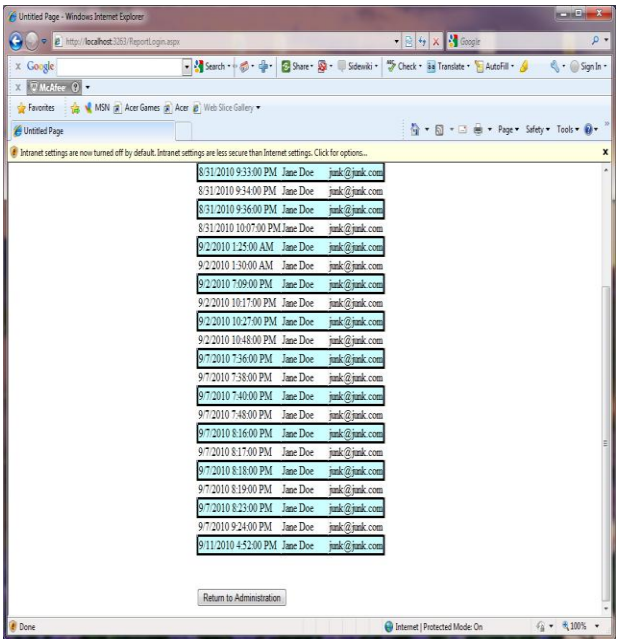

Figure 11: System Logon report

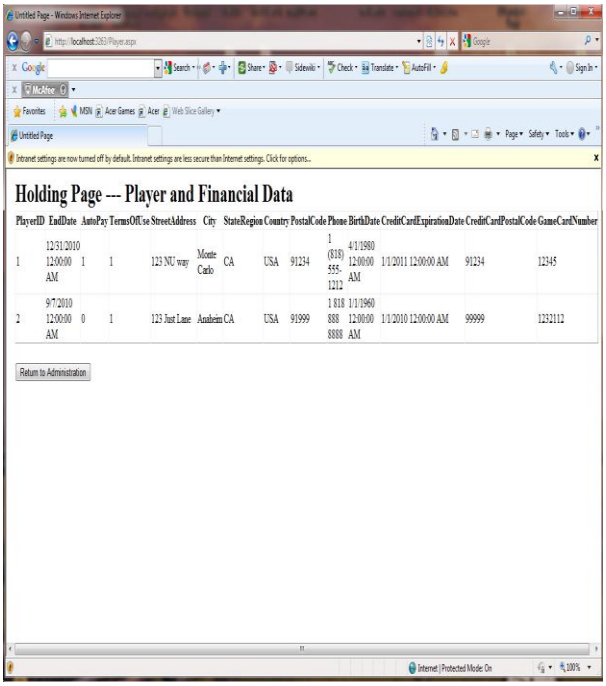

Figure 12: Player information and financial data page

Figure 12 shows the Player information and financial information for credit card billing or electronic check franking. Data security,which is of high concern for liability reasons,is addressed through three methods: All user character (and game) data is stored as Hard Binary; all accounts will be assigned unique identifiers using SQLidentity procedural process; all personal and financial account data will be encrypted as well as stored Hard Binary. Required data is retrieved and verified by data string matching.

Usability testing was conducted with the aid of the Westwood College IGDA (International Game Developers Association) Game Development club (Los Angeles campus) who supplied a group of five testers for this project, all of whom were in their senior year of instruction and had either testing experience, a formal class on testing techniques, or both as qualifications. These students were all selected as their educational program was geared toward becoming the type of technical developer that the project product should interest as a platform hosting solution component. The test platforms were all configured to the recommended hardware and software requirements.

The method selected for testing was to allow full access to the current builds of the database as well as the demonstration interface pages, which allowed the testers to change data fields in the database tables, import data from a trial game [8], create user accounts and alter existing user account data.A short series of questions were then given to the testers to rate the product using a Likert scale ranging from 0 (lowest, unacceptable) through 5 (highest, very acceptable). Additionally comments were encouraged, including a request for “Suggestions for improvement”. The three areas of interest were ease of use, design logic, and meeting functional requirements.

The results were surprising, as a basic interface sequence design flaw was immediately identified by the testers. The survey result were as; ease of use with average 4.4, design (of application) logic with average 3.0 and functional requirements with the average 4.6. Additionally, it was discovered that the browser history block was not functioning correctly and a user could enter interior interface pages without logon authority. This issue,as well as the design concerns,were addressed in the final build.

6. IMPLEMENTATION

MMO Front-end is designed to support Massively Multiplayer Online (MMO) environments by supplying low maintenance user (player) account management functions.MMO Front-end is designed to support Massively Multiplayer Online (MMO) environments by supplying low maintenance user (player) account management functions. The code of the program consists of three Structure Query Language (SQL) databases which can be modified to support any type of user (player) related avatar, "saved game", and personal as well as financial account data.Administration of the MMO Front-end in default mode requires little if any technical know-how, other than in the initial installation. Advanced use for licensed modifications will require the administrator to be familiar with MS SQL environments, in particular SQL Server 2008 and SQL Server Management Studio. Additionally the administrator should be very familiar with the developer's choice of interface implementation language, with ASP.NET being the supplied default format. Backup utility service is included and runs by default to either an administrator assigned directory or the default directory of C:\Backup\TheFiles\ if none other is assigned.

Administration of the MMO Front-end in default mode requires little, if any, technical know-how, other than in the initial installation. Advanced use for licensed modifications will require the administrator to be familiar with MS SQL environment, in particular SQL Server 2008 and SQL Server Management Studio.Additionally the administrator should be very familiar with the game developer's choice of interface implementation language, with ASP.NET being the default language.Backup utility service is included and runs by default to either an administrator assigned directory or the default directory of C:\Backup\TheFiles\ if none other is assigned.

The SQL runtime routine for extracting data is embedded within the MMO Front-end code and will extract and write data to the generic character repository under the appropriate numbered fields, of which there are over 100 to allow for a large depth of character data, character artifact data or game item data to be economically stored. The generic character table allows the system to compare player avatar statistics from the player's hard drive (where most of a modern MMO environment data actually resides) to what the runtime world on the server side thinks the statistics should be. If the fields match, the user may "play". If the fields don not match, the game referee (administrator) must resolve the discrepancy and determine if "cheating" (illegal user alteration of character or item data) has occurred. The last procedure returns a "1" if there is a match and a "0" if not (and color flags the value area in question). This will return a true or false in any basic, C++, Javaor derived language that the developer may chose to use to develop their player content interface.

7. CONCLUSION

We have presented a prototype front and back end for a system that can be used by the game developers for managing players account functions and other functions. The system is designed to be user friendly and simple. In its current implementation the system includes all the mandatory features in order for it to be useable.With this we present an alternate approach to the expensivethird part solution. As future work our approach will be to study and reduce the storage overhead.

References

1. Alexander, T. (2003). Massively Multiplayer Game Development. MASS: Charles RiverMedia.
2. Boyd, G. S. (2006.) Business and Legal Primer for Game Development. MASS: Charles RiverMedia.
3. Crandall, R. W. and Sidak, J. G. (2009). VIDEO GAMES: SERIOUS BUSINESS FOR AMERICA'S ECONOMY. Retrieved June 11, 2010 from: http://www.theesa.com/newsroom/seriousbusiness.pdf
4. Entertainment Software Association (2010). 2010 Essential Facts About the Video GameIndustry. Retrieved July 15, 2010 from: http://www.theesa.com/facts/pdfs/ESA_Essential_Facts_2010.PDF
5. Gallagher, M. (2008). 7th International Conference on Entertainment Computing Speech,September 27, 2008. Retrieved July 15, 2010 from: http://www.theesa.com/search/proxy.pl?terms=AAA&url=http%3A%2F%2Fwww.theesa.com%2Fnewsroom%2Fcmu_092708.asp
6. mmorpgtoplist.com (2010). MMORPG Toplist: Private Servers, free servers, Game Servers,Powerful Servers. Retrieved September 24, 2010 from: http://www.mmorpgtoplist.com