Design and implementation of a web-based battle application for the New Horizon RPG

Bachelor’s Thesis in Computer Engineering

Tinglu Xing
Acknowledgement

I want to thank you the following people for helping me during the thesis project:

— Wagner Ouique de Morais—my supervisor. Thank you for providing helpful suggestion during the process, the discussion inspired me new ideas to implement the project.

— Michal Lysek and Tobias Persson—CEO and technical consultant of the company MedicWave. Thank you for helping me to understand the whole battle rules and offering some technical support.
ABSTRACT

With the popularity of online role-playing games, a web-based battle application is designed to help the players to improve their characters for the combat. The developed prototype concentrates on one-player-multi-characters mode. The project consists of three major objectives. The first one focuses on the development of web-based interfaces for user registration and login. This functionality benefits the user with an individual and personalized member account. The second goal is to improve the existing web-application for character creation so the user data input is stored in a database for posterior retrieval. The third objective refers to the development of a web-based battle application that allows players to test created characters in a battle, one character against another one. This project also contributed with a logical database model that overcomes some design issues, such as normalization problems, in the current database model.
CONTENTS

1. INTRODUCTION ........................................................................................................... 1
  1.1 BACKGROUND ........................................................................................................... 1
  1.2 PROJECT OBJECTIVE ............................................................................................... 1
  1.3 OUTLINE .................................................................................................................. 2
2. METHOD AND TOOLS ................................................................................................. 3
  2.1 WEB-BASED APPLICATION ...................................................................................... 3
  2.2 ONLINE GAMES ........................................................................................................ 3
  2.3 TOOLS ...................................................................................................................... 4
3. USER REGISTRATION AND LOGIN INTERFACE ....................................................... 5
  3.1 PROBLEM DESCRIPTION ......................................................................................... 5
  3.2 USER REGISTRATION ............................................................................................... 5
  3.2.1 DESIGN AND IMPLEMENTATION ........................................................................ 5
  3.2.2 RESULT ................................................................................................................ 6
  3.3 LOGIN ...................................................................................................................... 8
  3.3.1 DESIGN AND IMPLEMENTATION ........................................................................ 8
  3.3.2 RESULT ................................................................................................................ 8
  3.3.3 DISCUSSION ......................................................................................................... 10
4. IMPROVED CHARACTER CREATION APPLICATION .............................................. 11
  4.1 PROBLEM DESCRIPTION ......................................................................................... 11
  4.2 SPECIFICATION DISPLAY ....................................................................................... 11
  4.2.1 DESIGN AND IMPLEMENTATION ........................................................................ 11
  4.2.2 RESULT ................................................................................................................ 11
  4.2.3 DISCUSSION ......................................................................................................... 13
  4.3 NAVIGATION MENU ............................................................................................... 13
  4.4 SAVE DATA IN THE DATABASE .............................................................................. 14
  4.4.1 DATABASE MODEL ............................................................................................... 14
  4.4.2 DESIGN AND IMPLEMENTATION ........................................................................ 14
  4.4.3 RESULT ................................................................................................................ 14
  4.4.4 DISCUSSION ......................................................................................................... 15
5. BATTLE APPLICATION ............................................................................................... 17
  5.1 PROBLEM DESCRIPTION ......................................................................................... 17
  5.2. DESIGN AND IMPLEMENTATION ......................................................................... 17
  5.2.1 APPLICATION ARCHITECTURE .......................................................................... 17
  5.2.2 ORDER DECISION .............................................................................................. 19
  5.2.3 DAMAGE CALCULATION .................................................................................... 20
  5.2.4 RESULT ................................................................................................................ 22
  5.2.5 DISCUSSION ......................................................................................................... 24
6. CONCLUSION .............................................................................................................. 25
7. FUTURE PLAN ............................................................................................................. 26
  7.1 OPTIMIZATIONS ...................................................................................................... 26
  7.2 MULTI-PLAYER ......................................................................................................... 26
  7.2 VALIDATION CODE ................................................................................................. 26

IV
REFERENCE .................................................................................................................. 27
APPENDIX I .................................................................................................................. 28
APPENDIX II ............................................................................................................... 29
APPENDIX III .............................................................................................................. 30
LIST OF FIGURES

Figure 1: Web-based application architecture .................................................................3
Figure 2: User table model ..................................................................................................5
Figure 3: Registration interface .........................................................................................6
Figure 4: Lively validation check during registration ........................................................7
Figure 5: Registration success page ..................................................................................7
Figure 6: Home page with a Drop-down menu .................................................................8
Figure 7: Login/Register panel ..........................................................................................9
Figure 8: Invalid information ...........................................................................................9
Figure 9: Member page ....................................................................................................9
Figure 10: Member panel .................................................................................................10
Figure 11: Specification of character after creation step ....................................................12
Figure 12: Character sheets .............................................................................................12
Figure 13: Navigation menu with submenu .................................................................13
Figure 14: Logical Database model for the battle system ................................................14
Figure 15: Diagram of the database model .................................................................15
Figure 16: Diagram of complete database model .....................................................15
Figure 17: Diagram of the initiative turn of the battle application ................................18
Figure 18: Diagram of the damage turn of the battle application ................................19
Figure 19: Diagram of dice rolling ..................................................................................19
Figure 20: Diagram of comparison .................................................................................20
Figure 21: Diagram of damage cases ..............................................................................21
Figure 22: Battle application interface ...........................................................................23
Figure 23: Battle result interface ....................................................................................23
Figure 24: Result saved interface ....................................................................................24
1. INTRODUCTION

The idea for developing a web-based battle application for New Horizon [1] has been proposed by Lypson Intelligent Systems [2]. Lypson is a local consultant company in Halmstad that provides software and hardware solutions. New Horizon is one of the websites owned by the company focusing on online role-playing games [3]. The fancy world is established on a rule system called Volt [4]. It contains two races which is Humans and Wafans. The battle is launched between these two races.

In order to play games in New Horizon, players must first fill out a character sheet (see Appendix I). A web-based application has been developed previously for this purpose, i.e., to assist players to create characters online. The main goal of the project is to extend the existing web-based application with new functionalities, such as user membership and the possibility to test created characters in a battle.

The presented prototype is implemented on text-based Multi-User Dungeon (MUDs) [5]. When the player launches the game, the request will be sent to the web server, and the output will be sent back the web browser and displayed to the player.

1.1 BACKGROUND

During the recent years, a role-playing game is more prevalent in the word. Each kind of game is based on a customizing rules system such as the most famous one Dungeon and Dragon [6]. The publication of it is considered as the origin of the modern role-playing games. It introduces a new idea to assign each player a specific character to take over the traditional concept.

With the development of the role-playing game, the elements of the game are getting richer. In order to satisfy the demands, a new type of role-playing game which is online role-playing game appears. It differs from the traditional pen-and-paper which uses electronic media to launch the game. It includes text-based MUDs and graphics-based successors. In this type, the computer takes the responsibility as a game master (GM) [7] to finish the calculation work. And as a player, he can receive the output of the interactive experience from rich interfaces.

1.2 PROJECT OBJECTIVE

The specific three goals of the project described as follow:

— To create a user registration and login interface to allow users to create an account and log in using a username, or email, and password before officially starting the battle adventure.

— To improve the present character creation application. Thus, the user is able to
view the specification of the character and choose afterwards to keep the information or not. Such character information is stored in a database. Furthermore, a web page is available for the user to check all the characters belonging to his username and to view the details of each one.

To create a web-based battle application. A web page displays all characters which belong to the current player. The user will select two of these characters to be included in the combat. The player also selects a combat mode, the distance for each character to the middle line. The result of battle will be displayed to the player, and the player has an option to save it or not.

The company Lypson provides a database model with default settings and the battle rules. The model and the rules are not allowed to be changed. The battle application is established according to the present rules.

1.3 OUTLINE

The remainder of this report organization is as follows: Chapter 2 presents the methods and tools used during the implementation. Chapters 3, 4 and 5 present the design, implementation and outcomes related to the user registration and login, improved character creation application, and battle application respectively. Chapter 6 presents conclusion. Chapter 7 concerns the future plan.
2. METHOD AND TOOLS

2.1 WEB-BASED APPLICATION

The architecture of a web-based application is composed by a web browser in the client and a web server and database management system in the server, representing presentation, application and storage respectively. Users access the application using a web browser. In order to run the application on the server side, a solution stack is needed, i.e., an operating system, web server and database management system. The web server is an engine capable of handling dynamic web content technology (such as ASP, CGI, PHP, Perl and etc.). Figure 1 presents such architecture.

This project makes use of the existing company’s solution stack, which is based on WAMP. WAMP means that the server runs Windows as operating system, Apache as web server, MySQL as database management system and PHP as scripting language. Furthermore, in order to lower the overload problems of web server in the classic client-sever model, a client-side scripting language and a group of interrelated web development methods, such as JavaScript and Ajax (Asynchronous JavaScript and XML) [8], can be used. Applications designed in this way provide a richer interactive experience without page reloading.

The advantage of web applications over platform dependent software applications are that web application wrap the complex logic in the backend and presents a dynamic and friendly interface for the user. Users can easily interact with the application following presented instructions in the website without getting confused even if they are outsiders in the field. Besides, another benefit concerns usability, since the user does not need to install an application.

2.2 ONLINE GAMES

The idea of computer games occurs to the computer scientists such as Babbage,
Turing and Shannon before the first personal computers came into existence. Now it grows rapidly with the advent of Internet.

Game online provides players an opportunity to experience the interaction with game players around the world. In order to display a multimedia content, a group of standard web technologies (HTML, JavaScript, common gateway interface scripts, etc.) are required to be installed on a local computer. These technologies are used to deliver the player’s input to the web server and present the output of the game on the web browser for the player.

Online games contain two modes such as single-user games and multiplayer games. In the multiplayer games, there are two main issues referring to simultaneous game-state and latency. The former one is generated when the multiple instances of the game client software are running simultaneously. The second one is the response time of the first problem that occurs when the input of the player and the output of the game need to be shared with other players connecting a slow network. The multiplayer type games are usually implemented as turn-based games or event-based games so that latency problem cannot easily be found by the user.

Online computer games rely on network communication. There are two models such as peer to peer and client/server. In peer to peer networking, the player’s computer needs to run both the game logic and input/output of the game. However, in client/server networking, the player’s machine only needs to run a client program.

In an online game, text, voice, sound and video play important roles. The first two elements enable the players to communicate with each other easily, and the latter two enhance the story line. Text communication requires keyboard and voice requires the use of microphone. Moreover, the player needs to install a plug-in on the local computer to for the media type used like sound and video.

2.3 TOOLS

The prototype is developed in Zend studio which is an integrated development environment (IDE) for PHP embedded in the Eclipse platform [9]. It helps the developer to track down the problems and debug step by step to fix bugs.
3. USER REGISTRATION AND LOGIN INTERFACE

3.1 PROBLEM DESCRIPTION

The existing website doesn’t provide a function for the users to register and login which means no data can be stored and no customizing pages can be loaded to the different users. Therefore, new pages should provide an interface for the user to register an account to obtain full functionality which non-registered users don’t have. Moreover, he can sign in by providing the proper credentials for the website to access privileges.

3.2 USER REGISTRATION

3.2.1 DESIGN AND IMPLEMENTATION

The registration form requires user to inform several fields such as ‘Username’, ‘Firstname’, ‘Lastname’, ‘Password’, ‘e-mail’, ‘City’, ‘State’ and ‘Country’. They are mandatory fields except the last three ones. The ‘Username’ and ‘e-mail’ should be unique because both of them are used to login. Moreover, ‘e-mail’ is also used to track password back which means no duplicated email address is allowed. Besides, the username requires more than 3 characters.

Figure 2 presents the table user in the database. In the table, ‘user_id’ is set as a primary key, which generates automatically during record insertion.

![User Table Model](image)

Figure 2: User table model

A validation check is applied under two circumstances. One is when the user fills
in each input box and leaves it, and another is when the register button is clicked. The validation check contains username and email address duplication, email validation, password match, first name and last name missing check. The result of validation check will be displayed to inform the user. Registration can be continued when all the fields are valid. When the user clicks the button ‘Register’, all the information will be stored in the database named ‘userinfo’.

Some websites make this availability check when the user fills in all the details in the form and presses the submit button. However, a live availability check is a preference when the textbox loses focus during the implementation. Therefore, the user can directly know whether the username has been occupied and the email address is valid.

Validation check can be implemented by several solutions such as PHP, JavaScript or Ajax. Comparing with common methods to check (with page refreshing), using Ajax to do a live validation is more efficient. By using this method, data is exchanged asynchronously between browser and sever to avoid full page reloading which decreases the problems of overload of sever and consumption of bandwidth. Thus, Ajax has been chosen with PHP to validate forms in both sides: client side (using JavaScript with Ajax) and sever side (using PHP).

### 3.2.2 RESULT

Figure 3 presents the registration interface. The left column lists the required fields with an asterisk symbol and optional fields. Furthermore, a note below reminds the user that the fields with an asterisk are required which means it has to be filled in. The words on the right side are the instruction for the user to fill in the blanks.

![Registration Form](image)

**Figure 3: Registration interface**

In Figure 4 a, the result of username validation check is displayed. When the
username has not been occupied, a message ‘Available’ will be presented. Otherwise, a message ‘Not available’ will be presented.

In Figure 4 b, when the user doesn’t fill in the firstname or lastname input box, a message ‘Lastname must be filled!’ or ‘Firstname must be filled!’ will be displayed.

In Figure 4 c, the message shows the result of email address validation check. If the email matches the standard format, ‘Valid’ will be displayed. Otherwise, ‘Invalid’ will be presented when it is input in a wrong way or even missing. In addition, if the same email address has been used, a message ‘Email already exists’ will be shown.

In Figure 4 d, validation checks whether the second input of password is equal to the first one when the textbox ‘Password confirm’ loses focus. If they are equal, a message shows ‘Confirm’. Otherwise, ‘Password does not match!’ will be displayed.

Figure 4: Lively validation check during registration

Figure 5 presents a registration succeed page. The page will be loaded when all the information offered by the user is valid, after the user clicks the ‘Register’ button.

Figure 5: Registration success page
3.3 LOGIN

3.3.1 DESIGN AND IMPLEMENTATION

A drop-down menu is applied in the login interface (see Figure 6), although the layout of the interface is not one of the requirements of the company. It keeps a simply and attractive style.

The menu contains login interface and registration entrance. The login interface requires the user to inform username or email and password to login. The user can choose remember me to save the password so that he can access the member page directly next time. Besides, the register button in the menu leads the new user to the registration form.

When the ‘Login’ button is clicked, it will lead to a validation check. If the username or email address or password is incorrect, an error message will pop out. When the information informed by the user is valid, login succeeds and a member page will be loaded.

3.3.2 RESULT

Figure 6 presents a home page with a drop down menu which provides basic information about the website. The menu contains a login interface and registration entrance.

Figure 6: Home page with a Drop-down menu

In Figure 7, when the menu pulls down, the main interface is divided into three areas. On the left side, there’s some useful instruction for the new visitors. The interface of login is in the middle position, and the register entrance is on the right side.
In Figure 8, when the information input by the user has no matching records found in the database, a message ‘Wrong username or email or password’ is displayed.

While the information is confirmed invalid, user succeeds logging in the website. Then a member page is loaded (see Figure 9). There are two links available in the page which is ‘create your character’ and ‘member panel’. It leads the user to the character creation page or member panel respectively.
Figure 10 displays the member panel. It is wrapped in a drop-down menu. In the member panel, it offers two options for the user which is ‘view a member page’ and ‘log off.

![Member panel]

**3.3.3 DISCUSSION**

When implement the login code, SQL injection attack [10] has to be taken as a consideration. User may input a SQL statement instead of a username which can cause unexpected security problem. Therefore, a function `mysql_real_escape_string` is used to escape the special SQL words as follow:

```php
$_POST['username'] = mysql_real_escape_string($_POST['username']);
$_POST['password'] = mysql_real_escape_string($_POST['password']);
```

Moreover, session and cookie are necessary methods in order to keep track of the user’s logged-in state, the ID as well as the password. The code is shown as follow:

```php
$row = mysql_fetch_assoc(mysql_query("SELECT id,username FROM user_info WHERE username='".$_POST['username']."' AND password='".md5($_POST['password'])."'"));

if($row['username']) // If everything is OK login
{
    $_SESSION['username']=$row['username']; // Store some data in the session
    $_SESSION['id']=$row['id'];
    $_SESSION['rememberMe'] = $_POST['rememberMe'];
    setcookie("tzRemember",$_POST['rememberMe']);
}
```
4. IMPROVED CHARACTER CREATION APPLICATION

4.1 PROBLEM DESCRIPTION

The present character creation system allows the player to create the characters. However, the user data input is not stored in a database so that the user cannot review the details afterwards. The improved system should provide a function to store the data in order to enable the player to view the specifications of the character which are stored in the database.

4.2 SPECIFICATION DISPLAY

4.2.1 DESIGN AND IMPLEMENTATION


Each item of ‘Attributes’ should show a value even it’s zero for the reason that it’s the basic property of the character (required by the company). Besides, it will show ‘None’ in the other properties when the player doesn’t select one.

4.2.2 RESULT

Figure 11 presents the character specification page. When the player creates a character and click the button to continue, a specification page will be loaded. The page covers the player’s name, character’s name and race name in the first field, and all the details about the character below.
After finish the creation step, player can view a character sheets. Figure 12 shows the page which is divided into two parts. On the left side of the character sheets, it lists all the characters belonging to the user. There’s a radio button before each name and a show button below.

The user can select one of them, and then the specification will be displayed on the
right side. It includes all the information of the character (see Appendix II).

4.2.3 DISCUSSION

By using SESSION, data can be passed to the showing-specification page from the character-creation page. First store the two-dimensional array in the SESSION in character creation page, and retrieve the data from SESSION in the specification page. The process is shown as follow:

```php
$arr7 = array(  
    array("$equipment1_ID","$number1"),  
    array("$equipment2_ID","$number2"),  
    array("$equipment3_ID","$number3"),  
    array("$equipment4_ID","$number4"),  
    array("$equipment5_ID","$number5"),  
    array("$equipment6_ID","$number6"),  
    array("$equipment7_ID","$number7"),  
    array("$equipment8_ID","$number8"),  
    array("$equipment9_ID","$number9"),
);  
//store in the session
$_SESSION["equipment"] = $arr7;  
//retrieve from the session
$arr7 = $_SESSION["equipment"];  
```

Moreover, some essential data belonging to ‘Stun’, ‘Injury’, ‘Willpower’ and ‘Weapons Modifiers’ property can be obtained by a special calculation method (see Appendix III).

4.3 NAVIGATION MENU

The navigation menu is on the top of the page. It contains four items which is Home, Character, Battle and Contact us. Home links to the start page. Character has two submenus which is Load Saved Characters and Create More Characters. Battle contains one submenu which is Start battle (Figure 13). It leads the player to the battle page. By the navigation menu, it is easy for the player to review the character’s information again, or to create more characters.

![Navigation menu with submenu](image)
4.4 SAVE DATA IN THE DATABASE

4.4.1 DATABASE MODEL

Users can create players, and players can create characters. Data is stored in the database named ‘userinfo’. There are seventeen tables designed in the section of battle application. Figure 14 presents the relation of these tables.

![Logical Database model for the battle system](image)

Figure 14: Logical Database model for the battle system

4.4.2 DESIGN AND IMPLEMENTATION

In the table ‘character’, the primary key ‘character_id’ is an automatic increment number. It generates when the data is stored in the table. Thus, the ‘character_id’ can uniquely identify each record to avoid the problem can be caused by ‘character_name’ duplication.

Similarly, the rest of the tables set different id as a primary key instead of the name. For instance, in the relation of table ‘character_armor’ and ‘armor’, ‘armor_id’ acts as a primary in the table ‘armor’ as well as a foreign key in the table ‘character_armor’.

4.4.3 RESULT

When the storage of data is a success, a page is loaded (see Figure 15). It presents a message that ‘character has been saved’, and the information for the player to start a battle next step.
4.4.4 DISCUSSION

The diagram below presents the relation among the whole tables. There’s one exception, table ‘encumbrance’ has no relation to others because it’s difficulty to display appropriately (see Figure 16 a). It requires a strength value to get connected. However, strength is one of the attributes which is stored as a point. The value of that point has to be obtained first from table ‘points_value’, afterwards the data can be retrieved by the strength value in the table ‘encumbrance’.

There’s an easy method to solve this problem if replace the strength value with the strength points. But table ‘encumbrance’ is the model provided by the company which is not allowed to modify.

During the character creation process, using method ‘Session’ to reserve the data and put it into an array or a two-dimensional array, then the required data can be passed...
to the storage step from the creation step.

Before storing the data into the database, a null check is necessary. For instance, if the player doesn’t choose any close weapons, then the element in two-dimensional array is null.

Array structure:

```
$arr5 = array("$weapon_id", "$strike", "$guard", "$grapple",
              "$Defense", "$Pierce", "$Stun", "$Injury")
```

Data is stored in the $SESSION ['close_weapon']. When comes to the IF-STATEMENT, it checks whether the first element in the array which is "$weapon_id" is null, here it is, then skip the insert query which means nothing is stored into the database. If the player selects any close weapons, then IF-STATEMENT will continue to check the first element in each dimension array. This is accomplished by a for-loop. Once it is not null, directly jump into the body of IF-STATEMENT. Before executing the insert query, there’s another IF-STATEMENT to check whether the rest elements in this array is null. If there’s any Null element, locate the position in the array and execute a particular insert query.

In the case of close weapon, the third weapon Wrestling’s details is as follow:

- **Row information:** (3, 1, 3, Null, Null, 8, 5, Null, 24, 22),
- **Table structure:** (id, user_id, weapon_id, strike, guard, grapple, defense, pierce, stun, injury).

The particular insert query uses the word ‘Null’ to replace the null array element. The SQL query is as follow:

```php
elseif (!isset($arr5[1][1]) && !isset($arr5[1][2]) && !isset($arr5[1][5])) {
    $query8 = "INSERT INTO close_weapon(user_id, weapon_id, strike, guard, grapple, defense, pierce, stun, injury) VALUES($user_id, "$arr5[1][0]", Null, Null, "$arr5[1][3]", ", ", "$arr5[1][4]", Null, "$arr5[1][6]", ", "$arr5[1][7]");"
    $result = mysql_query($query8) or die(mysql_error());
}
```

The reason for the SQL query above is that a null array element cannot use the same insert query as the existing array elements to store into the database. It should use Null in the position of the null element rather than mix them together.
5. BATTLE APPLICATION

5.1 PROBLEM DESCRIPTION

The existing application just allows the player to create characters. They cannot check how powerful created characters really are in a battle. The further step is to accomplish a battle application.

5.2. DESIGN AND IMPLEMENTATION

The battle application is required to be established according to the RPG Rule System named Volt provided by the company. It is a customizing rule system for the New Horizon which means it is not allowed to modify.

In the application, there’s an interactive interface for the player. The player is required to choose which one he wants to battle another one. The selection is among the saved character sheets. He should also decide a battle mode between close combat and ranged combat mode. The distance of each character to the middle line should be also informed by the player.

The result of the battle will be displayed as soon as the battle is over. It presents who is the winner, who is the loser and all the details of damage got in the combat.

5.2.1 APPLICATION ARCHITECTURE

Basing on the battle rules, the application contains two turns which is Initiative (turn 1) and Damage Calculation (turn 2). The battle starts when the player accomplishes the preparation step.

Both characters will run Initiative turn first (see Figure 17). The goal of this turn is to determine the order about who is going to shoot first next turn. No matter whether the aim or attack action is a success, the dice numbers (White dice and Black dice, each one is a 20 sides dice) affect the order. The method of order decision will be discussed next section.
When the second turn starts, the attacker (the first order character) begins to shoot the target (the second order character). Before hit area calculation, it firstly checks the action state and weapon carried. The action state is either aiming or running. When the state is aiming, the character shoots. The hit area will be checked and the damage will be calculated (see Figure 18 a). When the state is running, the weapon carried will be checked next (see Figure 18 b). If the character uses a ranged combat weapon, he aims under the circumstance the weapon within rang. And he jump to point 1 to shoot next turn (see Figure 18 c). If the character uses a close combat weapon and the distance to the target is within two meters, he attacks the target and the hit area and damage will be calculated (see Figure 18 d).

Two characters execute turn 2 alternately. The damage is accumulated separately turns by turns. It includes stun damage and injury damage. Both of them will be updated to a higher level when twice damage level is same. The method of implementation will be described next section. The game is over when one of the damage is up to critical level.
5.2.2 ORDER DECISION

The order is decided by the dice number. Each player has a white dice and a black dice. Each of them is a twenty-side dice. There are four circumstances of dice rolling of each character as follow:

- **Figure 19**: Diagram of dice rolling

  ‘Q’ means the dice number is qualified which is smaller than or equal to the weapon’s aimed value, while ‘Unq’ means the dice number is unqualified which is larger than the weapon’s aimed value. The aimed value is one of the properties of the combat weapon.

  In the condition of one qualified dice and one unqualified dice, the qualified one is selected (see Figure 19 a). While in the circumstance of both qualified dices and both unqualified dices, the larger dice number is selected (see Figure 19 b). The selected number is reserved temporarily.
When the second character also gets a dice number after rolling, a comparison between these two numbers is shown as follow:

![Diagram of comparison](image)

‘1 Q’ means first character’s dice number is qualified, and ‘1 Unq’ means the first character’s dice number is unqualified. Similarly, ‘2Q’ and ‘2Unq’ stands for the same meaning.

In the condition of one qualified dice and one unqualified dice, the one who has the qualified dice number gets the shoot order one. At the same time, another character gets the second shoot order (see Figure 20 a). While in the circumstance of both qualified dices and both unqualified dices, the shoot order one belongs to the one who has the larger dice number. And meanwhile, another character’s shoot order is two (see Figure 20 b).

### 5.2.3 DAMAGE CALCULATION

The damage is caused by two new rolling dice numbers. Each of the two dices has twenty sides which means the number the dice can get is from one to twenty. In the code implementation, a random function is used as follow:

```c
$white_dice=mt_rand(1,20);
$black_dice=mt_rand(1,20);
```

Depending on the battle rules, different hit area causes distinct damage. When the hit area is no armor protection area, damage is calculated without damage level reduced. When the hit area is reinforced area or normal area, damage calculation is divided into four cases as follow:
Pierce is one of the attributes belonging to the attacker’s weapon. The stun and injury damage reduction is implemented as follow:

```javascript
$scanarray=array(
    "$s_negdable1","$s_minor1","$s_moderate1","$s_serious1","$s_serious2","$s_critical1",
    "$s_minor1","$s_moderate1","$s_serious1","$s_serious2","$s_critical1",0)
);

$injuryarray=array(
    "$i_negdable1","$i_minor1","$i_moderate1","$i_serious1","$i_serious2","$i_critical1",
    "$i_minor1","$i_moderate1","$i_serious1","$i_serious2","$i_critical1",0)
);
```

A two-dimension array is used to reserve the data. The first element in the array is an array storing original damage data. The second one is the one-level reduced by forwarding each element one position. Therefore, when it comes to the one-level damage reduction, the second array in the two-dimension array should be used to lock the real damage level.

When the hit area is no protection area, no damage level reduces. Therefore, the first array in the two-dimension array is used to define the damage level.

The whole damage calculation procedure is divided into eleven functions as follow:

- `function hit_area1(...)`
- `function hit_area2(...)`
- `function passive(...)`
- `function aim(...)`
- `function run(...)`
- `function stun_white_die(...)`
- `function damage_black_die(...)`
- `function normal_stun_damage(...)`
- `function normal_injury_damage(...)`
- `function stun_die_check(...)`
- `function injury_die_check(...)`

- Function `hit_area1` is called when the first character is hit.
- Function `hit_area2` is called when the second character is hit.
- Function passive is called to transform a distance value to a passive value.
- Function aim is called when the character needs to aim.
- Function run is called when the character needs to run.
- Function stun_white_die is called when the character gets stun damage caused by the white dice number.
- Function injury_black_die is called when the character gets injury damage caused by the black dice number.
- Function normal_stun_damage is called when the character gets stun damage in the no protection area.
- Function normal_injury_damage is called when character gets injury damage in the no protection area.
- Function stun_die_check is called to check whether the character is dead by the stun damage.
- Function injury_die_check is called to check whether the character is dead by the injury damage.

The function hit_area1 is called to calculate the damage the second character got caused by the first character. It shows as follow:

```
damage_level2-hit_area1($reinforced_armor_coverage2,$white_die,$black_die,$normal_armor_coverage2,  
$place,$reinforced_deflects,$reinforced_stop,$setuarra,$
$reinforced_slow,$injuryarky,$setum,$injury1,$counter,$channel);
```

Similarly, the rest of functions are called to calculate values.

### 5.2.4 RESULT

Figure 22 presents the battle application page. The interface contains two fields. In the left side, it presents a character list for the player to select. The player can chooses a team for each character in the check box. Besides, player can also selects a battle mode between ‘Close combat’ and ‘Ranged combat’ by clicking the radio button, and inputs a distance for each character in the input box.

The right field in the page below displays the battle specification and the result. In Figure 23 a, it shows the original information including character’s name, team and the distance input by the player. In the Figure 23 b, the result of initiative turn is displayed. The dice numbers of both characters are included. The information below which is in Figure 23 c, presents the order of the shoot turn. In the Figure 23 d, the result of battle with the dice number and damages are shown. In the Figure 23 e, the winner, the loser and the illustration of death is displayed.

Moreover, in the area below, player can choose ‘Save’ to store the battle result (see Figure 23).
When the battle result is successfully saved, a page will be loaded to inform the player (see Figure 24).
5.2.5 DISCUSSION

The character Megan and Sucher are chosen to combat to each other in a ranged combat mode. The distance between them is 40 meters (see Figure 23 a).

Both of them are succeed in aim in the initiative turn. Therefore, the shoot order one belongs to the one whose dice number is larger. In the real game, the dice number is a random number in the range from 1 to 20. However, the dice number is set in the test to make sure that the codes work as expectation. Here, Magen’s white dice number is seven, and Sucher’s is six (see Figure 23 b).

Obviously it turns out Megan gets the order one so that he first shoots next turn. And Sucher second shoots when Megan finishes his turn (see Figure 23 c).

It shows the dice number and damage caused by the character each turn. In turn one, Sucher causes Megan both critical stun damage and severe injury damage. Because Megan’s stun damage level is up to the critical level, game is over (see Figure 23 d).

The result display the winner is Sucher and the loser is Megan. Besides, it also illustrates the reason of death (see Figure 23 e).
6. CONCLUSION

The prototype of a web battle application based on the current battle rules is completely implemented. Three objectives are achieved.

Firstly, users can register and login on the website. Secondly when the character creation is finished, all the data can be stored in the database. User can review the specification from the character sheets page. In the end, the battle application can be launched. And then it presents the result and it can be saved in the database.

The major technologies used in the thesis project are PHP, JavaScript, Ajax, CSS, HTML and MySQL. A ‘form’ and ‘POST’ method is used to transport data from page to page. In addition, ‘SESSION’ and ‘COOKIE’ are also effective way to reserve and pass the data.

Functions are used during the implementation. The encapsulation and scalability of the codes is highly improved. Furthermore, it avoids the duplicated changes in the code caused by the change of the battle rules.

In order to develop the battle calculation part, diagrams are useful to analysis the structure of the rules. It demonstrates the demand of each section clearly. The codes are built according to the diagrams to meet each requirement to make sure that the whole application works.

In conclusion, the web application offers an interactive experience for the players of the online role-playing game. The result of the battle application is as correct as the pen-and-paper mode. According to the feedback from the company, it is beyond the expectation. The structure of the application is well designed. The most complicated battle rules are succeeding in translating into a calculation model. Moreover, nice user interfaces are designed although it is without of the requirements.
7. FUTURE PLAN

Although the prototype works, there are also some future plans can be made.

7.1 OPTIMIZATIONS

In order to reduce the response time and bandwidth cost, more optimizations of structure have to be made. Since the HTML codes and PHP codes mix together, it’s better to build a frame to distinguish them.

7.2 MULTI-PLAYER

The prototype concentrates on one-player mode. Although there’s some interactive experience between the web sites and the player, it is not enough. A multi-player mode battle application can be designed. However, a multi-player game is more complicated than a one-player game. The interaction times and damage calculation should be considered.

7.2 VALIDATION CODE

In the process of the registration step, a validation code check needs to be added. It protects the information to be falsified by hacker.
REFERENCE

[1] New Horizon  
http://www.newhorizon1.com/index.php
[2] Lypson Intelligent Systems  
http://www.lypson.se/
[4] Volt system  
http://www.voltsystem.net/doku.php?id=en3.0:start
[9] Zend studio  
http://www.dynamicdrive.com/dynamicindex1/ddsmoothmenu.htm
[12] Drop-down Menu  
[13] Username availability check  
http://youhack.me/2010/05/04/username-availability-check-in-registration-form-using-queryphp/
[14] Validation check  
### APPENDIX I

**Character Creation - Step 2**

**New Horizon**

- **Player Name:** [Name]
- **Character Race:** [Race]
- **Economy:** [Economy]
- **Class:** [Class]
- **Appearance:** [Appearance]
- **Background:** [Background]
- **Traits:** [Traits]
- **Armors:** [Armors]
- **Close Combat Weapons:** [Close Combat Weapons]
- **Ranged Combat Weapons:** [Ranged Combat Weapons]
- **Equipment:** [Equipment]

---

**Attributes**

- **Strength:** [Value]
- **Dexterity:** [Value]
- **Constitution:** [Value]
- **Intelligence:** [Value]
- **Wisdom:** [Value]
- **Charisma:** [Value]

**Backgrounds**

- **(No background)**
- **(No background)**
- **(No background)**
- **(No background)**
- **(No background)**
- **(No background)**

**Traits**

- **(No trait)**
- **(No trait)**
- **(No trait)**
- **(No trait)**
- **(No trait)**
- **(No trait)**

**Armors**

- **(No armor)**
- **(No armor)**
- **(No armor)**
- **(No armor)**

**Close Combat Weapons**

- **(No close combat weapon)**
- **(No close combat weapon)**
- **(No close combat weapon)**

**Ranged Combat Weapons**

- **(No ranged combat weapon)**
- **(No ranged combat weapon)**
- **(No ranged combat weapon)**

**Equipment**

- **(No equipment)**
- **(No equipment)**
- **(No equipment)**

---

*To learn more about attributes, and to read about the different kinds of Backgrounds and Traits which are available for your character visit our RPG page and have a look at the Player's Manual. You can read more about the different traits, weapons, and armor in our sexual tabletop. They are all available in this table.*

*For more information on character creation and how different things are calculated, see the Character Creation Guide.*
APPENDIX II
**APPENDIX III**

Calculation method:
First, calculate using the data offered by the user by a special formula. Then use the result as a key word to search for records from the database to retrieve the required data.
For instance: To get the data of Stun.
Formula:
Stun Table Value=Strength+Constitution+2*(Toughness + Willpower)
Calculate the stun value according to the formula. Assume the result is 37. Search the sum value in the table ‘cc_stun’, retrieve the row when the sum value is equal to the result. The matching record is as follow:

<table>
<thead>
<tr>
<th>ID</th>
<th>Sum</th>
<th>Negligible</th>
<th>Minor</th>
<th>Moderate</th>
<th>Serious</th>
<th>Severe</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>37</td>
<td>5</td>
<td>10</td>
<td>15</td>
<td>21</td>
<td>25</td>
</tr>
</tbody>
</table>

Then, retrieve the value of Negligible, Minor, Moderate, Serious and Critical.
The output is:
Stun:
Negligible: 5
Minor: 10
Moderate: 15
Serious: 20
Critical: 25
Other formulas:
Injury Table Value=Strength+Constitution+2*Toughness
Weapon Modifiers:
Fence=Close Combat-7
Power= (Strength + Close Combat)/2-7
Precision=Ranged Combat-7