Monster Mash – Team Awesome-er

Design Specification

|  |  |
| --- | --- |
| Author: | Joshua Bird, Phil Wilkinson, Tom Hull, Dave Haenze, Chris Morgan, Kamarus Alimin, Szymon Stec, Lewis Waldron |
| Config Ref: | SE\_02\_DS\_01 |
| Date: | 07-12-2012 |
| Version: | 1.1 |
| Status: | Draft |

Department of Computer Science

Aberystwyth University

Aberystwyth

Ceredigion

SY23 3DB

Copyright © Aberystwyth University 2012

CONTENTS

CONTENTS 2

1. Introduction 3

1.1 Purpose of this Document 3

1.2 Scope 3

1.3 Objectives 3

2. Architechtural Description 3

2.1 The DATABASE component 3

2.2 The CLIENT component 3

2.3 The SERVLET component 3

3. Sequence Diagrams 4

4. Class Diagram 8

5. Dependency Description 9

6. Class Interface Description 9

REFERENCES 9

DOCUMENT HISTORY 10

# Introduction

## Purpose of this Document

The purpose of this document is to outline the detail and description on various parts of the design for the group project. This document should be read in a manner that takes into consideration the details of the group project assignment and the group project quality assurance plan [1].

## Scope

This document gives a breakdown of the group project design via various methods and descriptions. This document should be especially read by all members of the coding team, and by the rest of the group so they have an understanding of what is happening inside the program.

## Objectives

The objective of this document is to:

* Give a description of the group project components from a high-level point of view
* Illustrate the flow of data between the client and server
* Depict a basic overview of the classes and how they are linked through a UML class diagram
* Give an understanding of the relationships and dependencies between modules in the dependency description
* List and give details of the interface classes, so that it can be used by other coders.

# Architechtural Description

The monster program consists of three components:

* The DATABASE component
* The CLIENT component
* The SERVLET component

## The DATABASE component

The database provides the communication between the computer program and the databases which is executed through a database management system. We will be using Java Database which 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.

## The CLIENT component

The client is where the user can request webpage from the server. Our program web-pages will 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.

## The SERVLET component

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.

# Sequence Diagrams

:Client

:Server

SendUsernameAndPassword

response

Login

:Client

:Server

Register

SendsRegisterDetails

response

response

ConfirmIfCorrectFriend

SearchDatabaseForFriend

:Client

:Server

Add a friend

response

:Client

:Server

Accepting a Friend

Response

ConfimFriend

Deleting a Friend

Response

:Client

:Server

SearchAddedFriendDatabaseAndRemove

OfferAMonsterForSale

Response

PurchaseABreading Monster

BuyMonster

OfferAMonsterForBreading

Monster

Response

Response

ShowMonsterList

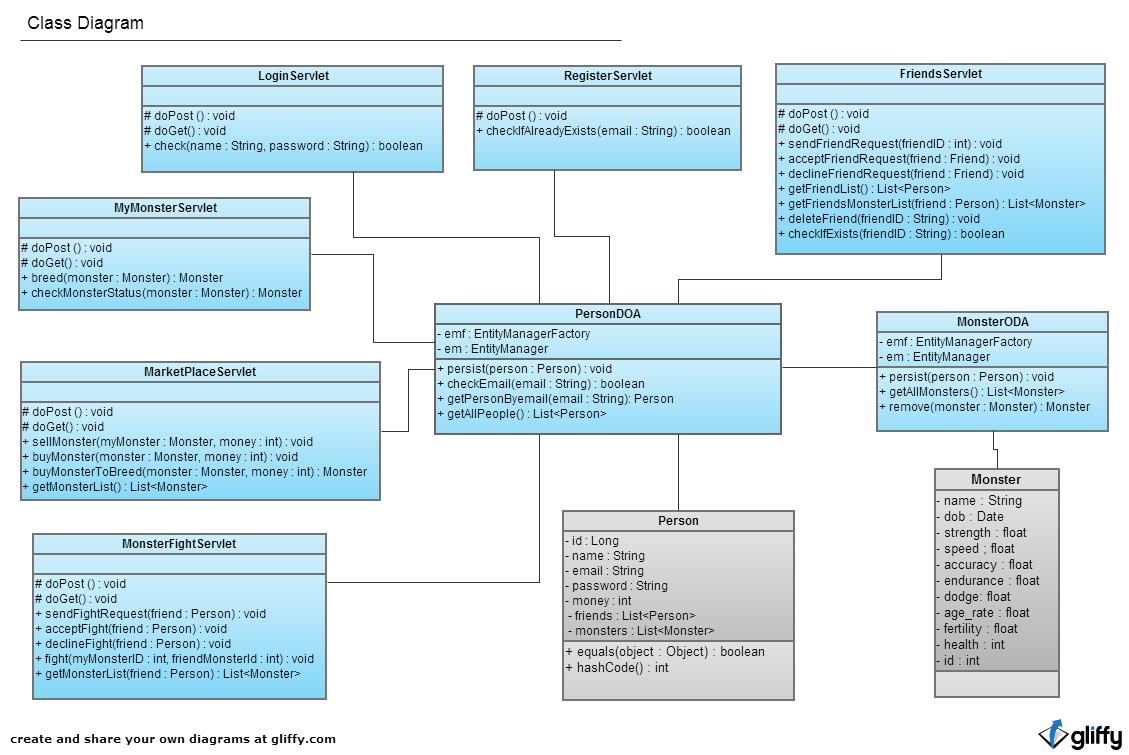
:Client

:Server

Response

Response

# Class Diagram



# Dependency Description

[Awaiting content]

# Class Interface Description

[Awaiting content]

REFERENCES

[1] Software Engineering Group Projects: General Documentation Standards. C. J. Price, N. W. Hardy. SE.QA.03. 1.5 Release

DOCUMENT HISTORY

| *Version* | *CCF No.* | *Date* | *Changes made to document* | *Changed by* |
| --- | --- | --- | --- | --- |
| 1.0 | N/A | 06/12/12 | N/A - original version | PW |
| 1.1 | N/A | 07/12/12 | Added architectural description, sequence diagram and class diagram | PW |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |