<GGRCE>

# Database Design Document

Version 1.0

11/04/2018

Document Number: 1

Contract Number: N/A

Table of Contents

[Database Design Document i](#_Toc508100475)

[Table of Contents ii](#_Toc508100476)

[List of Figures ii](#_Toc508100477)

[List of Tables ii](#_Toc508100478)

[1. Introduction 1](#_Toc508100479)

[2. Overview 2](#_Toc508100480)

[3. Assumptions/Constraints/Risks 3](#_Toc508100481)

[3.1 Assumptions 3](#_Toc508100482)

[3.2 Constraints 3](#_Toc508100483)

[3.3 Risks 3](#_Toc508100484)

[4. Design Decisions 4](#_Toc508100485)

[4.1 Key Factors Influencing Design 4](#_Toc508100486)

[4.2 Functional Design Decisions 4](#_Toc508100487)

[4.3 Security and Privacy Design Decisions 5](#_Toc508100489)

[4.4 Performance and Maintenance Design Decisions 5](#_Toc508100490)

[5. Detailed Database Design 6](#_Toc508100491)

[6. Database Administration and Monitoring 7](#_Toc508100493)

[6.1 Roles and Responsibilities 7](#_Toc508100494)

[6.2 Backup and Recovery 8](#_Toc508100496)

List of Figures

List of Tables

## Introduction

This document pertains to the existing automated system, labeled or termed with the provisional title “Generic RPG Example”, created using the Microsoft application “Access”. This documentation is meant to be used as reference during and after the implementation as a framework and justification for the choices made. The intended viewers are those involved in the design implementation and testing of the database. The ensuing material documented is transposable, due to altercation’s that may occur through-out development.

In relation to the DDD (Domain Driven Design) the current situation of the database, currently heralds information concerning Pre-set Enemy’s, User driven Heroes, Skills, Heroes Skills and Enemies Skills totaling five tables with one to one relations being formed throughout.

With regards to the security or privacy issues that may arise through applying DDD,the scope of what encompasses security in this instance narrows down to authentication and authorization.

## Overview

The origins of this project stemmed from the requirement to produce a functional data base or system that could benefit users and coincide with the creation of a game. The Data base was essentially designed as a design pivot for developers seeking to produce a game, aiding in the development and ideas for the entities set to be integrated into their game.

The database’s aesthetic look and function are set to be simplistic, this is partly due to improve the ease of use, though primarily due to limitations regarding time and software (noted below).

In regards to the application used to develop said database, as alliterated in the introduction is Microsoft access. Therefore it is plausible to link multiple access databases or use a back end database such as Microsoft SQL server to scale and support more data and user’s. Though in this instance the concluding decision is not absolute or decided, but it is increasly likely to interact and use a back end database such as Microsoft SQL to interact with other software (compatible) based on the purpose of the database.

## Assumptions/Constraints/Risks

### Assumptions

The dependencies regarding the database design for the system required to be noted is the compatibility, due to using an older version of Microsoft Access (2007) it is likely in newer versions for discrepancies to appear, in particularly in the UI format (Forms).

Overload of traffic could be cause of concern due to the database being of a small nature and all the information being stored on one file.

The use of Table partitioning (a database process where very large tables are divided into multiple smaller parts.) would mean that queries that access only a fraction of the data can run faster because there is less data to scan. Thus this could be a possible mitigation to the lack of storage and also useful during the time frame when there is a high user count.

### Constraints

Though not currently a constraint for this database, due to this constraints severity it is noted that Microsoft Access is finite meaning that due to the limitation of 2 GB storage if the data base is to be expanded changes may be necessary.

Further relating to the issue of expansion or building upon the current structure of the database, the software used in development is multi-user limited. This means that it is not possible to hold more than 225 concurrent users in regards to the technical limit, with the real world limit totaling 10-80 (depending on the application).

All the information in the database is saved into one file. Thus limiting options and how the data may be utilised; slowing down reports, queries, and forms. The performance will become slow as the user scales data size. Multimedia data can use up MS Access limited space quickly.

### Risks

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Risk | Probability | Severity | Risk Effect | Mitigation |
| 1 | Scope Creep | 30% | Low | Developer’s ideas may be of too lofty thus causing certain variables to spiral out of the developer’s control. | Through the use of SDLC, SCRUM and DDD (Models), though not definite if followed decreases the chances of said risk appearing. |
| 2 | Growth Prompt | 35% | Moderate | The technical limit of user’s is 225, thus a rapid rise in users could cause crashes or traffic overload. |  |
| 3 | Storage | 70% | High | The software used for development is Microsoft access, levying only 2GB, thus as the data base expands it will affect runtime. | An optimal solution would be to configure the database making altercations or use another database thus sharing the work load and increasing the storage size as two files are used to store information, rather than one. |
| 4 | Unauthorized Access / Security Issues | 40% | Catastrophic | The data base may experience a breach, thus compromising data stored within the database. | The issue of security stems from using Microsoft access to develop thus the system is dependent on that software regardless of the prompts or failsafe’s enacted. Whilst the use of DDD (domain driven design) during development itself is a cause for concern.  With regards to possible mitigation levels or layers of access and security may be enacted to make it increasly harder to crack along with monitorisation and updates thus ensuring and lowering the likely hood of a hack. |

## Design Decisions

### Key Factors Influencing Design

The whole structure or frame work of the database is based around pre-set or generated entities, there features and their relationships. Therefore it was important that the design of the database could accommodate these elements, ustilising tables to store said data and forms/ reports for the modification and access of data perusing from a user’s view.

Moreover due to the data base’s overall purpose, the design focused on simplicity rather than complexity, in an aesthetic and functional aspect. Thus alleviating the work from the user, as organisation is handled by the system and the addition of list boxes mean less exertion.

The key factors influencing the design are essentially all explicit in the requirements, thus leading to the aforementioned database.

### Functional Design Decisions

The functionality of the database from an application perspective, will entail a graphic user interface similarly seen on web sites, functioning to how websites may perform. The forms that correlate to the different tables, will be separated through and accessed by tabs (aiding navigation). This method or layout is similarly used for the reports as well.

The forms will enable input in addition to the deletion of data, thus giving the user a certain level of control over the information stored within the tables. In addition preferences or limitations will be set for each field ascertaining that inputs deemed invalid are not processed. This ensures that values are within respectable limits when configuring, ensuring the system does not break or the values are un- user -friendly.

Subsequently two buttons are placed in each form enabling the user to remove, add or update values stored within the tables. The inserts are generally sequential, with user input determining where the information is stored in the table. Through the altercations made by a user to the tables, the reports contents also update simultaneously (since the reports are not static but fluid), with the reports contents arranged in a condescending table format.

### Security and Privacy Design Decisions

The security and privacy of the database is partly dependent on the software used in implementation, however to further boost and decrease the chances of an attack or breach of data. The database will include a small level security layer; with password guards, thus only those whom have the rights or have been given admission will have dedicated access to the file. This implementation of a security level and user access control should bolster the security, though unauthorized access is still possible, thus leading to the ensuing idea of monitoring for unusual activity (detailed further under “Administration and Monitoring“).

### Performance and Maintenance Design Decisions

The Data base will be centralized due to Access constraints (thus requiring all information to be stored on one file) and scope. However if conceivable, a prospect is that the data base may be distributed, with portions of the database stored in multiple physical locations and processing is distributed among multiple database nodes. This would essentially enhanced security and alleviate the work load of the hardware (server).

In regards to the storage or the reorganisation of data, due to Access constraints of only being able to hold 2 GB of data, the information stored will most likely be purged after a pre-set amount of time or have to be stored client side rather than server to clear space.

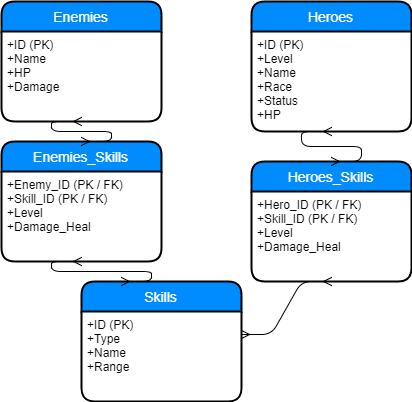
Furthermore the decision to delete data would probably require a SLA that notifies the user before use as to the time frame in which if not update the information will be deleted. Whilst information may be archived for a following few weeks until final deletion depending on user response.

The Database will have a backup or versions, in the event of loss of data or failure of hardware, however this will have significant impact on the availability, therefore frequent user service will be needed, since during this time the data may be inaccessible. This is also the case in the event of an update and therefore the user would have to be notified beforehand to make subsequent preparations.

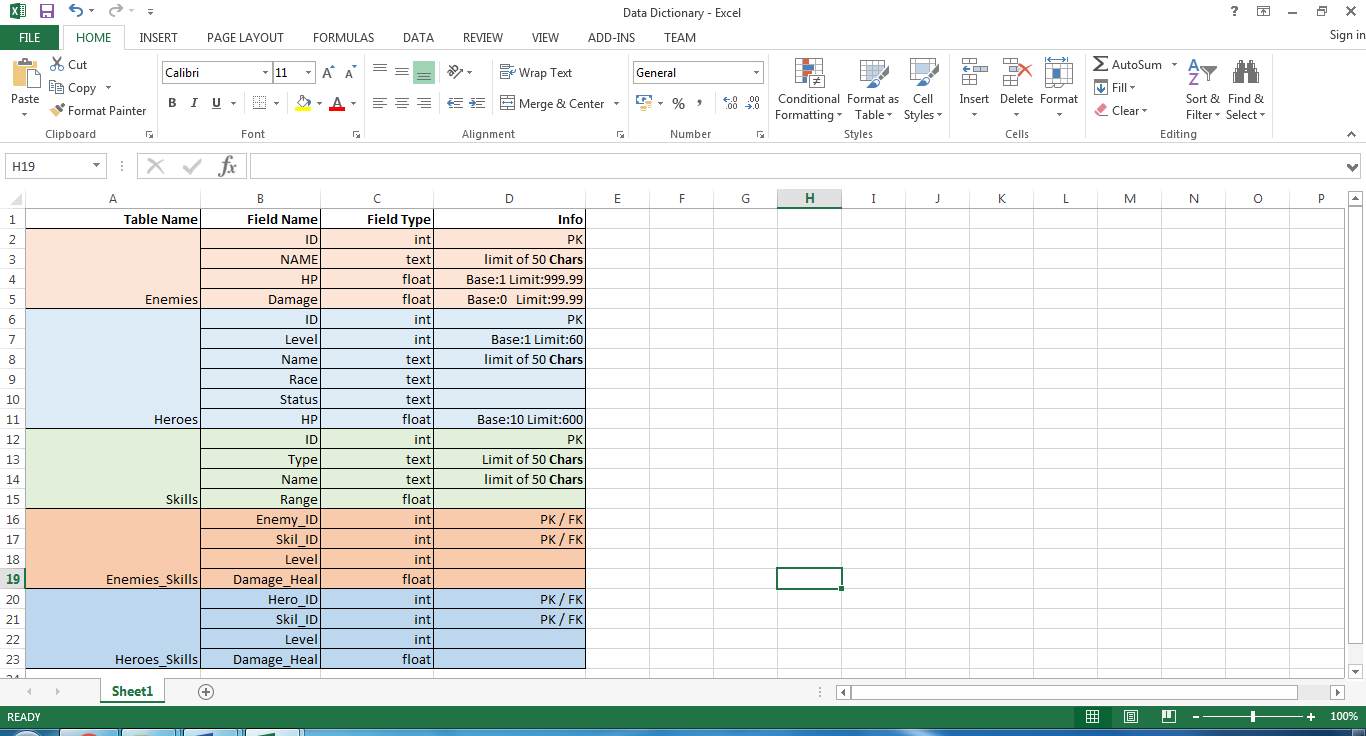
## Detailed Database Design

The below ERD (Entity Relationship Diagram) illustrates how the “entities”, in this instance or scenario; Hero, Enemies, Skills, etc., relate to each other within a system.

The diagram was subsequently produced though the use of an online flow chart maker termed “Draw.IO”. The tools and features available enabled the creation of said Diagram, include formatting, tables, shapes, text, the ability to link to a GitHub Repository, etc.



The Figure illustration below is a Data Dictionary, Containing the table names, field names, field type, and info, in principle pertaining to any information concerning the Database.



## Database Administration and Monitoring

### Roles and Responsibilities

The impending roles and responsibilities following the creation of said database have essentially fallen to; Matthew Sides, a core developer in the production of the database for the organisation Reboot Games. However the list detailed below and personal named are provisional as other people from different sectors in Reboot games may be added.

Data Base Administrator: Matthew Sides

System Administrator: Matthew Sides

Security Administrator: Matthew Sides

#### Security and Privacy

The security of the database is retained through security level access, using a sub schema in the user control component so that user’s may have full control on whom may access or see their data.

In regards to privacy, procedures will be needed to be put into place to ensure a certain level of trust between the organisation and user’s, ensuring integrity.

### Backup and Recovery

The Access Database Backup & Restore tool can be used to restore the Access configuration data, but it does not restore messages, which means that that it cannot be used to recover lost live messages (no applicable in this instance).

Whilst Versions or backups will also be retained, thus allowing for roll backs, in the event of any damage or data loss (copies).

Appendix A: Data Flow Diagrams

DFDs

Appendix B: Flowcharts