Warhammer 40k

Game Assistant App

Web API & App Design Specification

February 5th, 2017

Development Team: Matthew Cormier

**Table of Contents**

**Title Pagei**

**Table of Contentsii**

**Introduction Memo3**

**System Overview4-8**

System Definition4

System Requirements4

Mission Statement5

Mission Objectives5

Database and Application Design Summary6

System Architecture6-7

System Lifecycle7-8

Software Requirements7

Hardware Requirements7

Assumptions and Dependencies7

User Views8

CRUDE Diagram8

Security8

**System Design9-12**

**Database Design9-12**

Class Diagram9

Normalization10-11

Data Dictionary12

**Introduction**

To: End users

From: Matthew Cormier, Project Manager

Date: February 05, 2016

Re: Capstone Design Report

Dear possible users of this web API,

This report will give a detailed overview of the planning and design of the Warhammer 40k game assistant application and web API. These applications are being built for a personal capstone project that is a pre-requisite to my graduation of the Computer Programmer Analyst program at Durham College. All copyrighted and trademarked material is owned by Games Workshop Group PLC. The purpose of this application is purely for education and this work will not be used within a commercial environment.

Further explanation of this system’s analysis and design will be provided below. Any additional documents, diagrams, and system outputs may be added as an addendum.

**System Overview**

**System Definition**

This system will be broken down into three applications. The entire system will be built within the .NET framework. An ASP.NET web API will be created and deployed online along with a MySQL database. This web API will allow applications to send RESTful HTTP requests to create records, update records, or retrieve records. The second application will allow administrators to access the database and perform all necessary CRUD operations. The last application will be for end users. This app will allow users to create army lists and aggregate game rules to assist them while playing games of Warhammer 40k and its many expansions.

The CRUD and user applications will be desktop Windows form applications and will be coded with C#. All applications will be designed through a model, view, controller schema.

**System Requirements**

* HTTP/web accessible database.
* Database is normalized and well designed for scalability and data integrity.
* Storing codex, rulebook, and player list records.
* Printable army lists and other related reports.
* Dynamically generated reports.
* Allow users to create, update, and delete game and army list records.
* Allow administrators to update the database as new rule editions and game supplements are released.

**Mission Statement**

This system will assist end users in setting up and playing games of Warhammer 40k by providing digital record keeping of rules, army lists, games, and match results.

**Mission Objectives**

To maintain (enter, update, and delete) data on games

To maintain (enter, update, and delete) data on player lists

To maintain (enter, update, and delete) data on rule sets

To maintain (enter, update, and delete) data on codex

To maintain (enter, update, and delete) data on units

To maintain (enter, update, and delete) data on unit equipment and options

To perform searches on army lists

To perform searches on units

To perform searches on unit equipment and options

To track the status of games

To track the status of army lists

To report on games

To report on army lists

To report on units

To report on unit equipment and options

**Database and Application Design Summary**

To best meet the project requirements, the DBMS will be based on a web server and will utilize a relational MySQL model. MySQL will be used in place of other database services because it is the de-facto industry standard for web-based database applications. MySQL is a high performance options that will provide fast query speeds and allow for easy scalability for the business to grow future web functionality. MySQL is also open-sourced, allowing customization of the DBMS software to tailor to the project, if needed.

The applications will be C# windows applications because this will allow me to design and build the applications within the small 12-week time constraint of the capstone project. The web API for the database will allow the foundation work to be reusable for possible future Warhammer 40k applications. The web API and applications will be protected by a user authentication.

**System architecture**

Figure 1.1 shows the simplified architecture of the system.

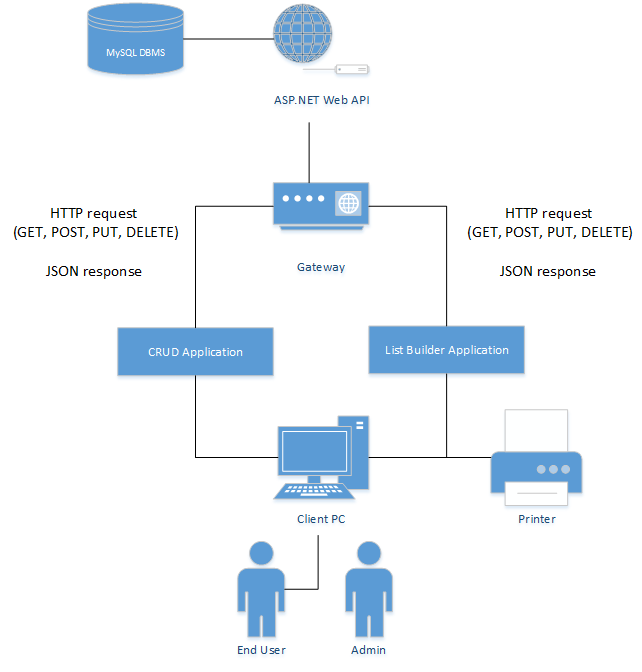


Figure 1.0 (System Architecture).

**System Lifecycle**

The development of the system will follow unstructured agile development. This is due to this project team only being one person and the sort 12-week turnover time. By the final deadline, a 1.0 prototype will be completed. Future prototypes may be expanded on.

**System Software Requirements**

Database: MySQL 5.7.15

Server: Apache 2.4

Framework: .NET

Languages: ASP.NET, C#, SQL

Repository: GitHub

Web server: GoDaddy

**System Hardware Requirements**

Client Windows PC

Client Printer

Internet access

**Assumptions and Dependencies**

This design of this system makes several assumptions about the software, hardware, and environment of both the customers and Aunt Rosie’s business. Including assumptions about the client machines and the web server.

Client machine assumptions:

* Client machine is in a fully functional state.
* Client machine has access to the internet.
* Client PC OS is Windows 7 or newer.

Server assumptions

* Server is online and accessible by users.
* The website has a registered domain name.
* The server is operating under normal (low) traffic.

**User Views**

The system will have two different levels of users. These users will have access to differing parts of the database through account logins and will affect what privileges the users have to read and write to records.

* Administrator – Access to full range of CRUD operations
* End User – Access to CRUD operations for their army lists and game records.

**CRUDE Analysis**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Admin | End User | Games | Missions | Lists | Units | Rules | Codex’s | Equipment |
| Admin | C, R, U | C, R, U, D | R, D | C, R, U, D | R, D | C, R, U, D | C, R, U, D | C, R, U, D | C, R, U, D |
| End user |  | C, R, U | C, R, U, D | R | C, R, U, D | R | R | R | R |

**Security**

Due to the simple nature of the web API and the lack of important information, the web API will only require basic security.

* Customized basic user account authentication
* Proper user views
* HTTPS for web API
* HTTP for MySQL server

These measures will protect the API and database from unauthorized access and cross-site attacks.

**Database Design**

**Class Diagram**

Figure 2.0 shows the class structure of the database.

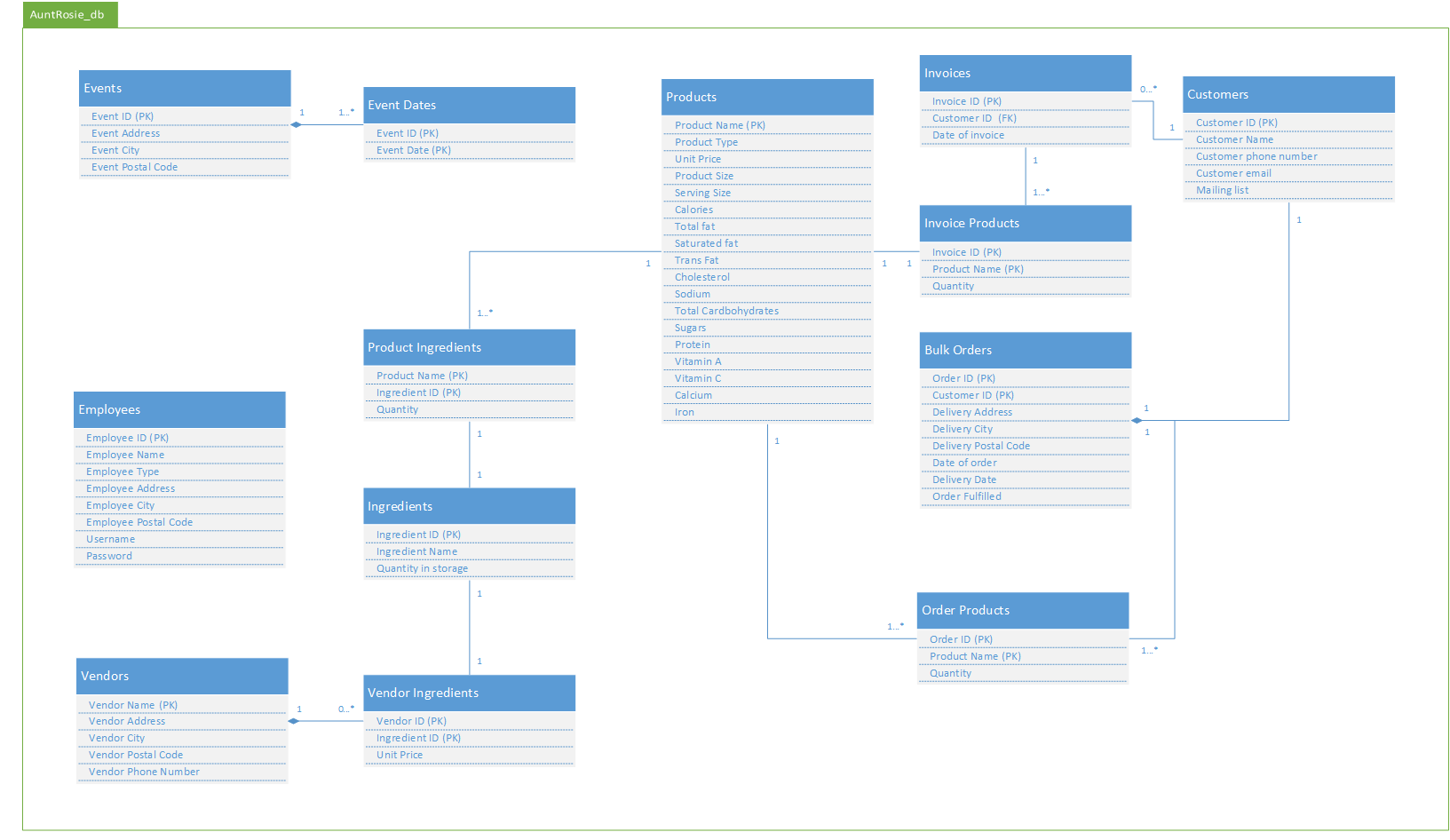


Figure 2.1 (Class Diagram)

**Normalization**

*3NF*

GAMES GAME\_ID, USER\_ID, MISSION\_ID, GAME\_NAME, GAME\_DESCRIPTION, NUMBER\_OF\_PLAYERS

MISSIONS MISSION\_ID, MISSION\_NAME, TABLE\_SIZE, MISSION\_DESCRIPTION, MISSION\_SPECIAL\_RULES, PRIMARY\_OBJECTIVES, MISSION\_LENGTH, DEPLOYMENT, SECONDARY OBJECTIVES, VICTORY\_OBJECTIVES

PLAYERS GAME\_ID, LIST\_ID, TACTICAL\_OBJECTIVES, FIRST\_TURN

SELECTED\_ GAME\_ID, MISSION\_RULE\_ID

MISSION\_RULES

MISSION\_RULES MISSION\_RULE\_ID, MISSION\_RULE\_DESCRIPTION

LISTS LIST\_ID, USER\_ID, LIST\_NAME, DATE\_CREATED, LAST\_UPDATED, MAX\_POINTS, RULE\_FORMAT, DETACHEMENT\_TYPE, WAR\_LORD\_TRAIT, MIN\_TROOP, MAX\_TROOP, MIN\_HQ, MAX\_HQ, MIN\_ELITE, MAX\_ELITE, MIN\_HEAVY\_SUPPORT, MAX\_HEAVY\_SUPPORT, MIN\_FORT, MAX\_FORT, MIN\_FAST\_ATTACK, MAX\_FAST\_ATTACK

SELECTED\_LIST\_RULES LIST\_ID, LIST\_RULE\_ID

LIST\_UNITS LIST\_ID, UNIT\_ID

LIST\_RULE LIST\_RULE\_ID, LIST\_RULE\_DECRIPTION

CODEXES CODEX\_ID, CODEX\_NAME, EDITION, DESCRIPTION

ALLIES CODEX\_ID, ALLY\_CODEX\_ID, TYPE\_OF\_ALLIANCE

UNITS UNIT\_ID, CODEX\_ID, UNIT\_NAME, UNIT\_CLASS, UNIT\_PRICE, UNIT\_COMPOSITION

UNIT\_CHARACTERS UNIT\_ID, CHARACTER\_ID

CHARACTERS CHARACTER\_ID, WS, BS, S, T, W, I, A, LD, SV, FRONT, SIDE, REAR, HP CHARACTER\_TYPE, MAY\_TAKE\_SPECIAL\_WARGEAR, MAY\_TAKE\_ARTIFACT

CHARACTER\_WEAPONS CHARACTER\_ID, WEAPON\_ID, QTY

WEAPONS WEAPON\_ID, WEAPON\_NAME, RANGE, S, AP, TYPE

CHARACTER\_ CHARACTER\_ID, SPECIAL\_WARGEAR\_ID

SPECIAL\_WARGEAR

SPECIAL\_WARGEAR SPECIAL\_WARGEAR\_ID, SPECIAL\_ISSUE\_WARGEAR\_PRICE, SPECIAL\_ISSUE\_WARGEAR\_DESCRIPTION, RANGE, S, AP, TYPE

CHARACTER\_WARGEAR CHARACTER\_ID, WARGEAR\_ID

WARGEAR WARGEAR\_ID, WARGEAR\_NAME, WARGEAR\_DESCRIPTION

CHARACTER\_ CHARACTER\_ID, SPECIAL\_RULE\_ID

SPECIAL\_RULES

SPECIAL\_RULES SPECIAL\_RULE\_ID, SPECIAL\_RULE\_NAME

SPECIAL\_RULE\_DESCRIPTION

CHARACTER\_ARTIFACTS CHARACTER\_ID, ARTIFACT\_ID,

ARTIFACTS ARTIFACT\_ID, ARTIFACE\_NAME, ARTIFACT\_DESCRIPTION,

ARTIFACT\_PRICE

USER USER\_ID, USERNAME, PASSWORD, DATE\_CREATED, LAST\_UPDATED

CHARACTER\_OPTIONS CHARACTER\_ID, OPTION\_ID

OPTIONS OPTION\_ID, WEAPON\_ID, OPTION\_DESCRIPTION, OPTION\_PRICE, MAX\_LIMIT, QTY, OPTION\_TYPE

CHARACTER\_RULES CHARACTER\_ID, RULE\_ID

RULES RULE\_ID, CHARACTER\_RULE\_NAME, CHARACTER\_RULE\_DESCRIPTION

**Data Dictionary**

See DataDictionary.pdf