

---

# System Design Document



for

FGG Creature Creator M<sup>3</sup>

Prepared by : Tristan Zippert

David DiFrumolo

Landon Thibodeau

Cedric Fahey

Matt Virgin

**Organization :** Moonwake Development

November 10, 2022

## **Table of Contents**

### [1 Introduction](#)

#### [1.1 Purpose of This Document](#)

#### [1.2 References](#)

### [2 System Architecture](#)

#### [2.1 Architectural Design](#)

#### [2.2 Decomposition Description](#)

### [3 Persistent Data Design](#)

#### [3.1 Database Descriptions](#)

#### [3.2 File Descriptions](#)

### [4 Requirements Matrix](#)

### [Appendix B – Team Review Sign-off](#)

### [Appendix C – Document Contributions](#)

# 1 Introduction

This is a capstone project for Frog God Games, in partial fulfillment of the Computer Science BS degree for the University of Maine. Frog God Games are accomplished publishers of role playing games, adventures, and addons for both modern and legacy gaming systems (Frog God Games Staff 2020). The company requires a front end for their database of creatures, M3. The front end of M3 will allow users to search, filter, and create creatures. The main objective of this application is to provide users a way to access the database and utilize it effectively for their game. The product is also meant to enable Frog God Games to bolster their sales.

## 1.1 Purpose of This Document

The purpose of this document is to describe the design of the M3 front-end system for use by Frog God Games and associated stakeholders. This document details the interaction between the system and the associated database. The classes, methods, and system components used to satisfy the functional requirements established in the SRS are also described. If the SRS describes what the system is going to do, this document describes how the system is going to do it.

## 1.2 References

“PL/SQL.” *PL/SQL for Developers*,

<https://www.oracle.com/database/technologies/appdev/plsql.html>.

Frog God Games Staff. (2020). About frog god games. Frog God Games. Retrieved October 20, 2022, from <https://www.froggodgames.com/frog-crew/>

Fowler, M., & Scott, K. (1997). Uml distilled: Applying the standard Object modeling language. Convert-O-Braille.

Badiuzzaman, M., & Rafiquzzaman, M. (2016). Development of “Virtual office model” software for entrepreneur and e-management. *International Journal of Innovation, Management and Technology*, 7(2), 41–45. <https://doi.org/10.18178/ijimt.2016.7.2.642>

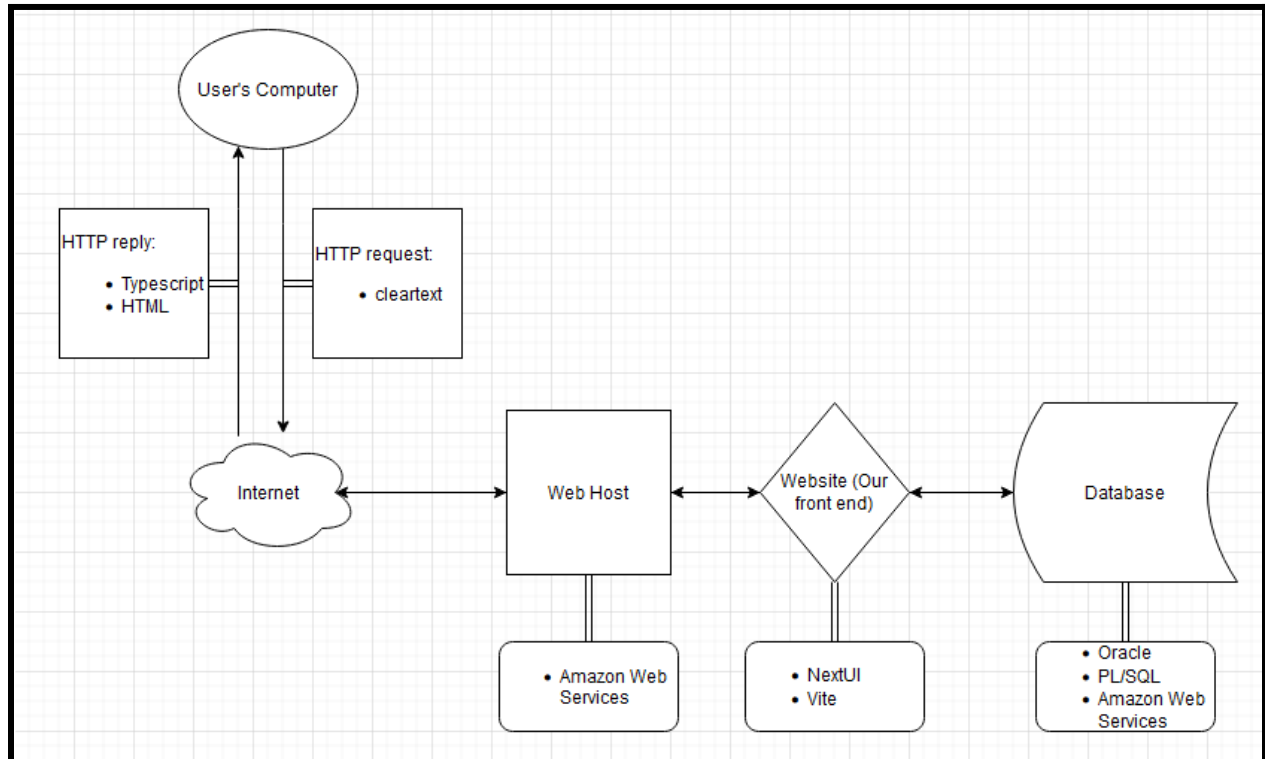
Moonwake Development (2022). System Requirements Specification. Retrieved October 20, 2022.

Moonwake Development (2022). User Interface Design Document. Retrieved: November 7, 2022.

## 2 System Architecture

This section provides an overview of the system, including design descriptions and a decomposition of the components included. It will also describe the technologies the system uses to interact with other components, including what's used to interact with the existing database.

### 2.1 Architectural Design

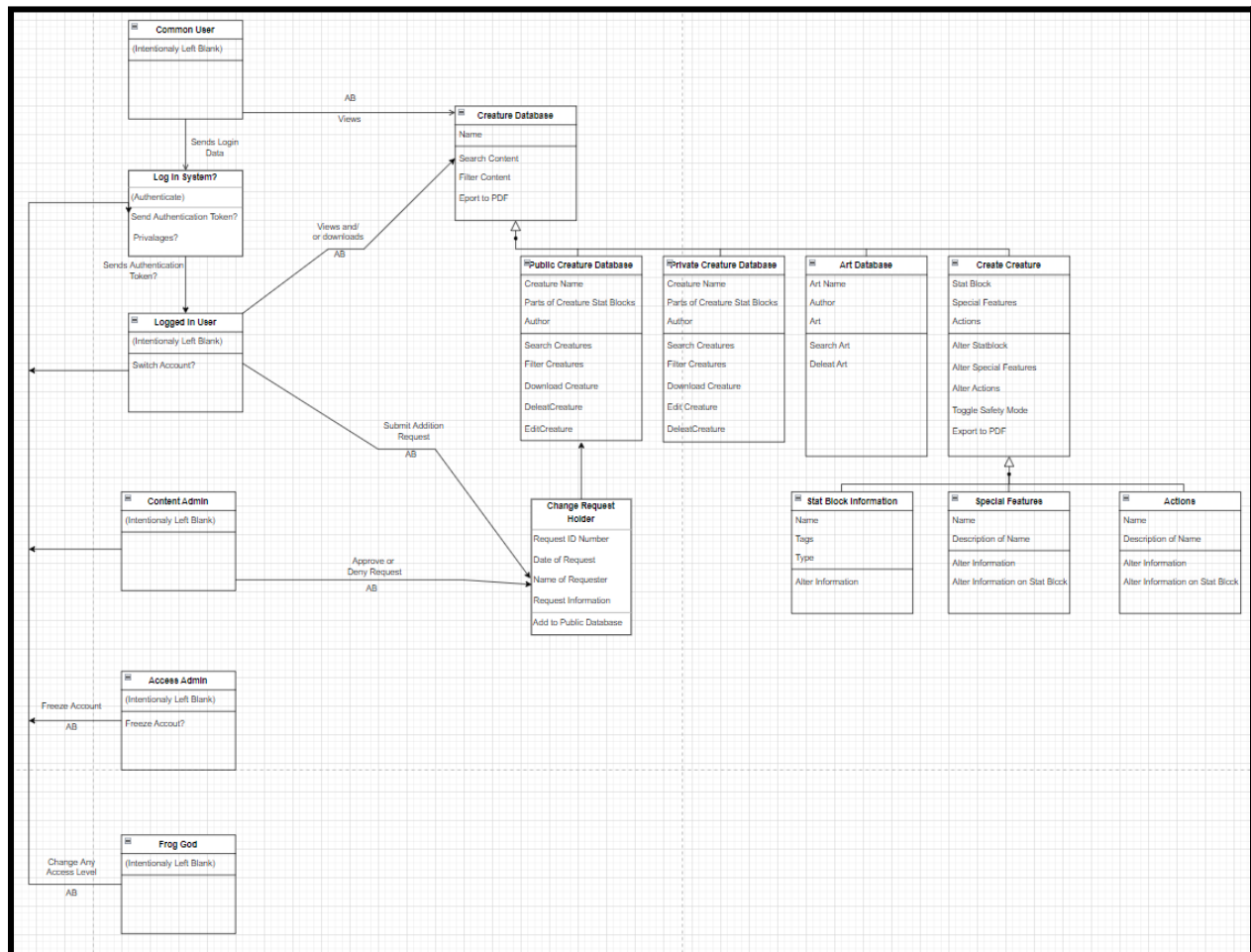


In this design, the user's computer will receive the information needed to display the front end through our Typescript and HTML code, sent to them after the web host for our website receives an HTTP request from the user. HTTP requests are typically in cleartext, which is simply data sent without encryption. The website itself will be designed with NextUI and Vite, tools made specifically for creating front end systems. NextUI is a Typescript library built on top of ReactUI, which allows us to quickly iterate through different designs and interactions for the user interface. Vite, on the other hand, is a utility for compiling web based code in Javascript/Typescript. These tools focus on the speed and performance of creating modern web applications.

The website will receive and transmit data to the database. Creature information is stored on the database, which the website will have to pull from to access and display. The website will also need to push to the database when a user creates a new creature that they want to store. The database itself is an oracle database built with PL/SQL, hosted on Amazon Web Services. The current plan is to call that PL/SQL from the database to modify it instead of writing totally new code. This plan will help speed up the development of the front end of M3.

## 2.2 Decomposition Description

This section describes the components that make up the system described in section 2.1 above. This system will take an object-oriented approach, where classes and methods will make up the bulk of the implementation for the front end. UML diagrams detailing these classes and their methods will be listed below. These diagrams describe the system as we know it at this point in time.



This is an Object Oriented system. The boxes in the diagram above represent the classes the system will use. Arrows represent the relationships between different classes. Hollow arrows from one class to another indicate that they are a part of (they inherit from) that class. For example, Public Creature Database, Privacy Creature Database, Art Database, and Create Creature are all part of Creature database. The first rectangle under a class name holds the attributes of the class. The second rectangle holds the methods. Attributes are the data that make up the class, and methods represent what the class can do with that data. Any arrow labeled AB means it represents a one way relationship. For example, the Content Admin can send to the

Change Request Holder, however not the other way around.

The question marks next to Log in System and its class attributes are there because we do not know how the Frog God Games account system is set up or how M<sup>3</sup> is going to pull the data from it for this project. The creator of the Frog God Games website must first be consulted to determine how they want their system to interact with ours.

### 3 Persistent Data Design

This section describes how data is going to be persistently stored across instances. The storage method is important for this project due to the fact that the M3 frontend will be accessed by multiple users at a time, all of which could pull and push information at separate times.

#### 3.1 Database Descriptions

The database used by the M3 system is an Oracle database hosted on AWS using PL/SQL. The database is separated into two major schemas: data schema and access schema. The access schema acts as an interface between the main table and data schema. The main data schema is centered around key joins via the “game\_system\_monster\_ID” key. This database already exists and is centered around the concept of third normal form. Moonwake development will not be creating any databases for this project, only accessing the one that already exists. Our front end will allow users to search their way through the database as well as add or remove their own creature entries, or allow users to base their own creatures off other entries. The front end will also have admin level users as described in the SRS that can freeze and manage another user’s access.

Schema diagram:

\* See M3 Documentation from Thom

I have not seen this documentation. EN

#### 3.2 File Descriptions

Our front end system uses no files as of 11/04/2022. Files are not needed because all data is stored in the database.



## 4 Requirements Matrix

The table below describes how use cases and their corresponding functional requirements (from the SRS) will be implemented. Each use case is listed alongside its number and any related functional requirements. The system component that will be used to implement that use case is listed in the final column.

#	Use Case	Functional Requirement(s)	System Component
1	Access Public Library	4. The system shall allow users to sort the public library by recently added public creatures	Method on Common User Class
2	Search Public Library	17. The system shall allow users to search the public library for keywords  18. The system shall allow users to sort the public library alphabetically	Method on Common User Class
3	Filter Public Library	1. The system shall allow users to filter the public library by the parts of a stat block, represented visually by buttons.  2. The system shall allow users to	Method on Common User Class

		<p>add stat block parts to their filter by clicking on buttons</p> <p>3. The system shall allow users to remove stat block parts from their filter by clicking on previously selected buttons</p> <p>16. The system shall allow users to filter the public library by author</p>	
4	Sort Public Library	4. The system shall allow users to sort the public library by recently added public creatures	Method on Common User Class
5	Account Access	6. The system shall allow users to log in with their Frog God Games Account	Method on Logged In User
6	Access own library	7. The system shall allow users to access their own library	Method on Logged In User
7	Search Own library	21. The system shall allow users to search their own library for keywords	Method on Logged In User

8	Filter Own Library	19. The system shall allow users to filter their own library	Method on Logged In User
9/15	Creature Generation	8. The system shall allow users to build creatures  12. The system shall suggest "default" attributes during creature creation to expedite the process	Method on Logged In User
10	Sort Own Library	20. The system shall allow users to sort their own library by recently added creatures  22. The system shall allow users to sort their own library alphabetically  23. The system shall allow users to filter their own library by author	Method on Logged In User
11	Redirect To Frog God Website (For Account Creation)		Method on Common User Class
12	Login	6. The system shall allow users to log in with their Frog God Games Account	Method on Logged In User

13	Freeze Account	15. The system shall allow account admins to freeze creature creation for specific accounts	Method on Content Admin
14	Change Account Privilege	15. The system shall allow account admins to freeze creature creation for specific accounts	Method on Account Admin
16	Edit Creature in own Library	<p>9. The system shall allow users to customize each part of a stat block for a creature they create</p> <p>10. The system shall allow users to edit creatures in their own library</p> <p>11. The system shall allow users to add a clone of a creature to their own library</p>	Method on Logged In User
17	Request Creature to Public Library	13. The system shall allow users to request a creature in their own library to be made public	Method on Logged In User
18	Approve or Deny Public Library Addition Request	14. The system shall allow content admins to approve or deny creature publication	Method on Content Admin

		requests.	
19	Add Creature to Global Library	<p>13. The system shall allow users to request a creature in their own library to be made public</p> <p>14. The system shall allow content admins to approve or deny creature publication requests.</p>	Method on Content Admin

## Appendix A – Agreement Between Customer and Contractor

By signing this document you are agreeing that all of the content above is accurate. You agree that the software specifications are accurate. The relations between the users as well as the terminology is accurate. All major functional and non-functional requirements will be implemented.

- In the future, if there are any updates or changes to this document that are agreed upon by the team the following procedure will take place. A review will be held with the client and the team members with the updated document. If, after the review, both parties agree upon the changes then the document must be resigned by each member before official use.

PRINT

SIGNATURE

DATE

Customer:

Edwin Nagy

X

Date 11/10/22

Comments: In the absence of seeing the attachment, which seems to be the most important new information in this submittal...

Team:

X

Date

X

Date

X

Date

X

Date

X

Date

## Appendix B – Team Review Sign-off

By signing this document each member of the development team determines that the collection of information withheld inside it has been reviewed and agreed upon. No member wishes to alter or change how it currently exists and there are no major points of contention within the team.

PRINT		SIGNATURE	DATE
-------	--	-----------	------

X	x		Date
---	---	--	------

Comments: N/A

X	x		Date
---	---	--	------

Comments: N/A

X	x		Date
---	---	--	------

Comments: N/A

X	x		Date
---	---	--	------

Comments: N/A

X	x		Date
---	---	--	------

Comments: N/A

## **Appendix C – Document Contributions**

- **Matt Virgin:** Section 2.1, 100%. Section 4, 10%. Section 2.2, 5%.
- **David:** Section 4, 90%.
- **Cedric:** Section 2.2, 95%
- **Landon:** Section 3, 50%
- **Tristan:** Section 3, 50%