

Moonwake Administrative Manual

**Developers: Cedric Fahey, Matt Virgin, David DiFrumolo,
Tristan Zippart, Landon Thibodeau.**

Figure 1: Moonwake Development Logo



Product: M³ A front end for Frog God Games' Creature Creator

30 Mar 23, 2023

M³ Front End
Administrator Manual

Table of Contents

	<u>Page</u>
Administrator Manual Template	1
(Adapted from Susan Mitchell)	1
1. Introduction	3
1.1 Purpose of This Document	3
References	3
2. System Overview	3
2.1 Background	3
2.2 Hardware and Software Requirements	3
3. Administrative Procedures	3
3.1 Installation	3
3.2 Routine Tasks	4
3.3 Periodic Administration	4
3.4 User Support	4
4. Troubleshooting	4
4.1 Dealing with Error Messages and Failures	4
4.2 Known Bugs and Limitations	4
Appendix A – Team Review Sign-off	4
Appendix B – Document Contributions	4

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 add-ons for both modern and legacy gaming systems (Frog God Games Staff 2020). The company requires a front end for its 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 with a way to access the database and utilize it effectively for its game. The product also enables Frog God Games to bolster their sales by introducing users to its website. Moonwake Development is a UMaine capstone team consisting of Cedric Fahey, Tristan Zippert, David DiFrumolo, Matthew Virgin, and Landon Thibodeau.

1.1 Purpose of This Document

watch switch of point
of view

repetitive

The purpose of this document is to outline the policies, procedures, and guidelines that a person or organization should follow to operate and manipulate the M3 product, including the front-end website application and the back-end Oracle database. This document is intended to serve as a guide for those who will be responsible for managing and supporting the system. It will provide a clear and concise guide that outlines the procedures, requirements, and tasks involved in the administration of the system. The document is divided into several sections that cover various aspects of the system, including its overview, hardware and software requirements, administrative procedures, and troubleshooting. Each section contains detailed information that will help you navigate different situations and scenarios in a consistent and appropriate manner, promoting efficient and effective operations within the system. By referring to this document, you will be equipped with the knowledge and tools you need to perform your administrative duties effectively and efficiently.

References

1. Moonwake Development (2022). System Design Document. Retrieved November 30, 2022.
2. Moonwake Development (2022). System Requirements Specification. Retrieved October 20, 2022.
3. Learning Software. "Oracle Instant Client Windows Installation Tutorial." *YouTube*, YouTube, 1 Jan. 2022, www.youtube.com/watch?v=v0TkVFG05c&list=PLkYi-U4Q0f_ulvDVTFVSAXdwmEDljM_Z4&index=5&t=286s.
4. Oracle Staff. "Oracle Instant Client Downloads." *Oracle*, Oracle, 2023, www.oracle.com/database/technologies/instant-client/downloads.html.
5. Nvm-Sh. "NVM-SH/NVM: Node Version Manager - POSIX-Compliant Bash Script to Manage Multiple Active Node.js Versions." *GitHub*, Git, 2023, github.com/nvm-sh/nvm
6. Triscuitcircuit. "Triscuitcircuit/Moonwake-FGG: Frontend Development for Dungeons and Dragons Creature Database Capstone Project." *GitHub*, Git, 2023, github.com/triscuitcircuit/Moonwake-FGG.
7. JetBrains Staff. "WebStorm: The Smartest Javascript IDE, by JetBrains." *JetBrains*, JetBrains, 2023, www.jetbrains.com/webstorm/

2. System Overview

2.1 Background



The M3 character creation web application is designed to provide users with a user-friendly and intuitive interface for creating and managing their monsters for roleplaying games. The application is intended to simplify the ~~character~~ ^{creature} creation process, allowing users to easily create and customize their characters. 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 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.

2.2 Hardware and Software Requirements

Discuss the hardware and software required for the installation and use of the system.

The system requires a computer using ~~either~~ Mac OS (12.0+, Intel or M-series), Windows (10+), or Ubuntu 22.04. Any IDE (Integrated Development Environment) will work for modifying the code, but Webstorm is recommended. Oracle Instant Client is used to interact with the Frog God Games database. (Docker? - not finalized). Node.js is also required, we recommend using some version of NVM (Node Version Manager) to make installation simpler. Once Node is installed, there are many other trivial requirements such as React, Vite, NextUI, and more that can be quickly and easily installed through the npm install command from the terminal in your IDE of choice. Git should also be installed in order to push any changes to the code to the codebase located at <https://github.com/triscuitcircuit/Moonwake-FGG>.

3. Administrative Procedures

The following describe the procedures that Administrators taking control of this system should know about.

3.1 Installation

General recommendation: [Webstorm](#)



For Mac OS machines:

- General requirements: [Brew](#), [Oracle Instant Client](#), [NodeJS](#), [Git](#)
- Mac (ARM) Requirements:



Back-end

- Self compiled Node Oracledb([here](#)), [Oracle Instant Client](#), [nvm](#)
 - After `nvm` is installed, follow [this](#) guide on how to set it up with Rosetta and ARM.
 - When `nvm` for ARM and Rosetta is set up, download the Oracle Instant Client for x86 Intel (as listed above), and copy the files to `/usr/local/lib`.
 - run: `npm -i` in the command terminal
 - Set `nvm` to intel, and run `node src/server.js`

Front-end

- Within the MacOS stable version of nvm, install [vite](#) globally.
- run `npm run start`

For Windows machines:

- Install Oracle Instant Client ([this](#) is a good tutorial)
- NVM (github [here](#) with installation instructions)
 - after install, run command prompt as administrator and enter command:
 - nvm install latest
- Open command prompt, cd into your desired folder, and pull the repo:
 - git pull <https://github.com/triscuitcircuit/Moonwake-FGG>
- Open webstorm and open the folder where the github repo resides
 - cd into the **fgg-front-end** folder with the terminal and run:
 - npm install
 - cd into the **fgg-back-end** folder and run the same command
 - if errors are encountered with oracle, run:
 - npm uninstall oracledb
 - npm install --save oracledb
 - Install Vite and yarn globally ('npm install --global vite' and same for yarn)
- For running the code under Windows, make sure the execution policy is set properly with this command (run in command prompt or windows powershell):
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope Process
- in the terminal, cd into **fgg-back-end** and run:
 - node .\src\server.js (establishes connection to database, must be re-run whenever code in fgg-back-end is changed)

```
Server is running on port 8080.  
Executing (default): SELECT 1+1 AS result FROM DUAL  
Connection has been established successfully.  
█
```

Figure 1: Successful back end connection

- open a new terminal in your IDE, cd into **fgg-front-end** and run:
 - npx tsc ; vite build ; vite preview (this will provide you with a localhost link to open the webpage, needs to be re-run whenever code in fgg-front-end is changed)

```
PS C:\Users\Matt\Desktop\School\Moonwake-FGG> Set-ExecutionPolicy RemoteSigned -Scope Process
PS C:\Users\Matt\Desktop\School\Moonwake-FGG> cd fgg-front-end
PS C:\Users\Matt\Desktop\School\Moonwake-FGG\fgg-front-end> npx tsc ; vite build ; vite preview
vite v4.1.1 building for production...
✓ 525 modules transformed.
dist/index.html 0.44 kB
dist/assets/index-f61be52c.css 12.73 kB | gzip: 2.68 kB
dist/assets/index-5049f29e.js 734.04 kB | gzip: 222.61 kB

(!) Some chunks are larger than 500 kB after minification. Consider:
- Using dynamic import() to code-split the application
- Use build.rollupOptions.output.manualChunks to improve chunking: https://rollupjs.org/configuration-options/#output-manualchunks
- Adjust chunk size limit for this warning via build.chunkSizeWarningLimit.
→ Local: http://localhost:4173/
→ Network: use --host to expose
```

Figure 2: Successful front end connection

For Ubuntu:

Git, [NodeJS](#), [global install vite](#), [Oracle Instant Client](#)

- **Front-end**
 - open the terminal to the front end directory and type the following command into terminal: `npm install`
 - If vite is setup correctly type the command: `npm run start` or `vite build`
- **back-end**
 - Open the terminal in the backend directory and type the following command:
 - `npm uninstall oracledb`
 - Followed by:
 - `npm install oracledb`
 - Then type the following command:
 - `npm install`
 - Once everything is installed run `node src/server.js` and the server should host

Server install?

For installing for server use on a specific domain, all references to the string "localhost:8080" have to be replaced with the domain name that is hosting the associated back-end project. Hosting a server can be done on platforms such as DigitalOcean or AWS.

3.2 Routine Tasks

Routine tasks for pushing code to the repository include but are not limited to: checking automated tests, checking github actions and manual testing. Automated tests occur with every push of code to the github repository and signify if the back-end server is running correctly. Github Actions occur when code is sent to the github repository and build the application. It is suggested to check the github actions status in a routine manner to make sure that the application is building correctly on the supported systems. Another routine task is in the form of manually testing API routing to verify that routes are working correctly after code alterations.

3.3 Periodic Administration

There is currently no periodic administration needed in the system created by Moonwake development. However, it is recommended that the npm packages used should be updated in accordance with security patches and updates.

3.4 User Support

User support will be only provided by the team handing out the access tokens in the form of the `.env` file to the receiving party. This file is responsible for a secure connection to DBMS system hosted by Frog God Games on AWS.

Oracle database has online support. Support for this project can be done in the form of pull requests to the github. Github pull requests can be verified by it building with github actions. If the client wants ownership of the github repository, it will be provided if Moonwake is contacted regarding that matter.

3.5 Adding Images (mark down where to change the placeholder images)

Images used by the project are kept in the `/fgg-front-end/src/` folder. The only image in there at the moment is `logo.png`, which is used in `/fgg-front-end/src/components/Navbar.tsx` to add the Frog God Games logo to the navigation bar. Additionally, the placeholder 'black void' images that can be seen are templates provided by NextUI. These can be changed in `/fgg-front-end/src/Display/Card.tsx`.

3.6 How our back-end code works

The back-end code hosts a Node instance that allows communication via web requests to the SQL database hosted by Frog God Games on AWS.

4. Troubleshooting

In this section, we outline the known bugs and limitations of the system that may impact end-user or administrator tasks and activities. We provide specific information such as code location, description of the bug or limitation, and reasons why it could not be fixed. Additionally, we discuss potential workarounds or ways to deal with these issues.

4.1 Dealing with Error Messages and Failures

- If an error arises with `oracledb`, try:
 - `npm uninstall oracledb`
 - `npm install --save oracledb`
 - if that doesn't work, make sure your oracle installation is added to your `PATH` (as described above in the installation section)
- If an error arises with Vite or Yarn, try:
 - `npm install --global vite`

- npm install --global yarn
- If the website does not display, right click on the page, select “inspect” and navigate to the console to see the error messages that tell you why the page did not display properly
- If vulnerabilities are discovered after running the npm install commands described above and in section 3.1, use the command:
 - npm audit fix --force
 and the vulnerabilities will typically be fixed

4.2 Known Bugs and Limitations

1. Limitation: Auto-Fill Feature for Creature Creation Form

Code Location: creature-creator.tsx, function fillFormWithPresets()


Description: In the creature creation system, a user is unable to use the auto-fill feature to populate the form with presets when creating a custom fantasy creature. This limitation may lead to additional time spent manually entering data into the form, especially for users who are unfamiliar with the available creature attributes. Additionally, some calculations / values used by the creature creation form may be inaccurate.

Reason for Limitation: The auto-fill feature could not be implemented due to time constraints and shifting priorities during the development phase. The focus was placed on refining the core functionality of the creature creation system and ensuring a smooth user experience.

How to Deal with the Limitation: Administrators and end-users should familiarize themselves with the creature attributes and options available in the form. To expedite the process of entering data, users are encouraged to create their own reference documents or templates that can be used to quickly copy and paste information into the form fields. Although this workaround may not be as convenient as an auto-fill feature, it will help minimize the time spent on manual data entry. The variables in the code can be changed however you like to better fit the rules of tabletop RPG games.

2. Limitation: Inconsistent Search and Filter Functionality for Creature Attributes

Code Location: search_and_filter.tsx, function searchAndFilter(),
GASYMO_GAME_SYSTEM_MONSTER.route.js

 **Description:** Due to the architecture of the database and the way data was initially structured, the search and filter system has a limitation in handling specific attributes of the GASYMO_GAME_SYSTEM_MONSTER. The lack of a consistent structure for these attributes makes it difficult to search and filter them in the same manner as other attributes, which may lead to incomplete or inaccurate search results. To work around this, the attributes of a monster (strength, dexterity, constitution, intelligence, wisdom, and charisma) are ORed

with each other, even when global AND is enabled.

Reason for Limitation: When the database was designed, the attribute structure was not standardized, leading to inconsistencies in how attributes are stored and retrieved. This limitation was not addressed during the development phase due to the complexity of restructuring the database and the potential impact on existing data and system functionality. Within GASMYO_GAME_SYSTEM_MONSTER.route.js in fgg-back-end\src\routes, each attribute is found by comparing the range of attributes the user is searching for with MOAB_MONSTER_ATTRIBUTES.MOAB_DISPLAY_TEXT. Because str, dex, etc, all use this column to store their value, ANDing them would not make sense, as MOAB_MONSTER_ATTRIBUTES.MOAB_DISPLAY_TEXT = "STR 12" AND MOAB_MONSTER_ATTRIBUTES.MOAB_DISPLAY_TEXT = "CON 13" can never be true.

How to Deal with the Limitation: To mitigate the impact of this limitation, administrators should educate users on the specific attributes that may not be searchable or filterable within the system. Users should be made aware that the search results might not be comprehensive and that manual cross-referencing may be necessary to ensure all relevant information is considered.



To address this limitation in the long term, a potential solution would be to restructure the database and standardize the attribute structure. This would require a thorough analysis of the existing data and potential impacts on system functionality. Careful planning and execution would be necessary to minimize disruptions and ensure a smooth transition to the updated architecture. It's also possible that a solution could be made simply by modifying GASMYO_GAME_SYSTEM_MONSTER.route.js, perhaps by implementing a count where the creature is only displayed if that count equals the # of attributes the user searched for. However, the user can currently tell if a creature has both the STR and DEX (for example) values they searched for by hovering over the creature name in the tabular view. If only STR is displayed, that creature does not meet the DEX constraint the user set.

3. Bug: Pagination produces extra, blank pages

Code Location: helpers.js in fgg-back-end/src/Database/config/

Description: Creature database and search pages display extra pages that users can click on, but they will be blank. Only the pages that present a loading symbol and eventual load creature data are real. Everything is there, but there are just extra, blank pages.

Reason: Calculation of the totalPages in getPagingData is higher than actual

Potential Solution: Recalculate how to get pages / use debugger to step through how totalPages is being calculated

4. Limitation: Pagination doesn't work when associated values are being searched for

Code location: GASMYO_GAME_SYSTEM_MONSTER.route.js in fgg-back-end\src\routes

Description: The pagination system that is used to speed up the loading time of

displaying the database is turned off when associated values are being searched for. This means that the pagination system works as expected for the creature database page, but will be turned off on the search and filter page if a user searches for attributes associated with GASMYO_GAME_SYSTEM_MONSTER instead of attributes directly on it. (These associated attributes include str, con, dex, int, wis, cha, hp, and size). This means that searches that fetch a lot of data could take a long time. For global AND, this isn't typically an issue, but can cause long loading times when global AND is disabled.

Reason for Limitation: Adding 'limit' and 'offset' to the findAndCountAll function call that fetches the data being searched for in GASMYO_GAME_SYSTEM_MONSTER.route.js makes the identifier for associated column names invalid - "MOAB_MONSTER_ATTRIBUTES.MOAB_DISPLAY_TEXT" becomes an invalid identifier, for example. (This column is used for finding the attributes of a monster).

How to Deal with the Limitation: We've had some success with setting subQuery: false in the call to findAndCountAll in GASMYO_GAME_SYSTEM_MONSTER.route.js. However, this made each creature card display only 1 row of a creature, and a single creature would take up multiple pages. It's possible that the creature-db-modal.tsx and creature-database.tsx components could be reworked to work around this issue.

5. Bug: Modal window refuses to reopen after opening the same one consecutively

Code location: fgg-front-end/src/pages/creature-db-modal.tsx, fgg-front-end/src/pages/creature-database.tsx

Description: Opening the same creature modal window consecutively will cause the user to be unable to open any more modal windows until they click out of them and search again or refresh the page. This applies to both the creature database page and the search and filter page.

Reason for bug: Something is wrong with the boolean for opening and closing the modal window

Potential solution: use debugger / console.log to identify how boolean is changing when windows are opened and closed

6. Limitation: Home page is a placeholder

Code location: fgg-front-end/src/components/Home-Demo.tsx

Description: The home page just shows some pretty pictures with some example buttons.

Reason: We don't have much to put on a home page, since the meat and potatoes of our application is in search and creation

How to Deal with the Limitation: Redesign Home-Demo.tsx however you like

In the event that new bugs or limitations are discovered, they will be documented and addressed in subsequent updates.

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 instructions and information above are clear.

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

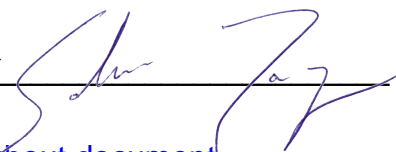
SIGNATURE

DATE

Customer:

Edwin Nagy

x



Date Apr. 10, 2023

Comments:

Comments added throughout document.

Team:

Cedric Fahey

x

Cedric Fahey

Date 4/10/23

David DiFumolo

x

David DiFumolo

Date 4/10/23

Landon Thibodeau

x

Landon Thibodeau

Date 4/10/23

Tristan Zippert

x

Tristan Zippert

Date 4/10/23

Matthew Virgin

x

Matthew Virgin

Date 4/10/23

Appendix B – Team Review Sign-off

By signing this document you are agreeing that all of the content above is accurate. 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 Cedric Fahey Comments: N/A	x <i>Cedric Fahey</i>	Date 4/10/23
X David DiFrumolo Comments: N/A	x <i>David DiFrumolo</i>	Date 4/10/23
X Landon Thibodeau Comments: N/A	x <i>Landon Thibodeau</i>	Date 4/10/23
X Tristan Zippert Comments: N/A	x <i>Tristan Zippert</i>	Date 4/10/23
X Matthew Virgin Comments: N/A	x <i>Matthew Virgin</i>	Date 4/10/23

Appendix C – Document Contributions

Landon Thibodeau - Sections 3 and 4

David DiFrumolo - Cover, Introduction, 2.1, Appendices A,B,C

Tristan Zippert - Mac and Linux install guides. server setup

Cedric Lahey - Sections 3 and 4

Matt Virgin - Windows install guide, section 3