SOFTWARE REQUIREMENTS SPECIFICATION

for

FGG Creature Creator M^3

Prepared by : Tristan Zippert
David DiFrumolo
Landon Thibodeau
Cedric Fahey
Matt Virgin

Organization : Moonwake Development

October 20, 2022

Contents

1 Introduction 3
 1.1 Purpose of This Document 3
  1.1.1 References 3
 1.2 Purpose of the Product 3
 1.3 Product Scope 4
2 Functional Requirements 5
3 Non-Functional Requirements 23
 3.1 Performance Requirements 23
 3.2 Safety Requirements 23
 3.3 Security Requirements 23
 3.4 Privacy Requirements 23
 3.4 Software quality attributes 23
 3.5 Non-Functional Requirements Testing 23
4 User Interface 25
5 Deliverables 26
6 Open Issues 26
Appendix A: Agreement Between Customer and Contractor 27
Appendix B: Team Review Sign-off 28
Appendix C: Document Contributions 29

1 Introduction

Frog God Games, a publisher of RPG style books, has a database of creatures that they want accessed by a website front-end creation application. The application would allow users to access the existing database, while at the same time allowing them to make creatures that can be uploaded publicly or privately. The users would also be able to acquire a custom made PDF file of any public creature, or any creatures created by the user. themselves

1.1 Purpose of This Document

✓

This document specifies what the system created by Moonwake Development for Frog God Games will do. The document's intended audience is Frog God Games. This document also acts as an agreement between the Moonwake Development team and the Frog God Games team on what Moonwake Development will be responsible for in regards to M^3. This document can be updated in the future with additional requirements if agreed upon and signed by both parties.

1.1.1 References

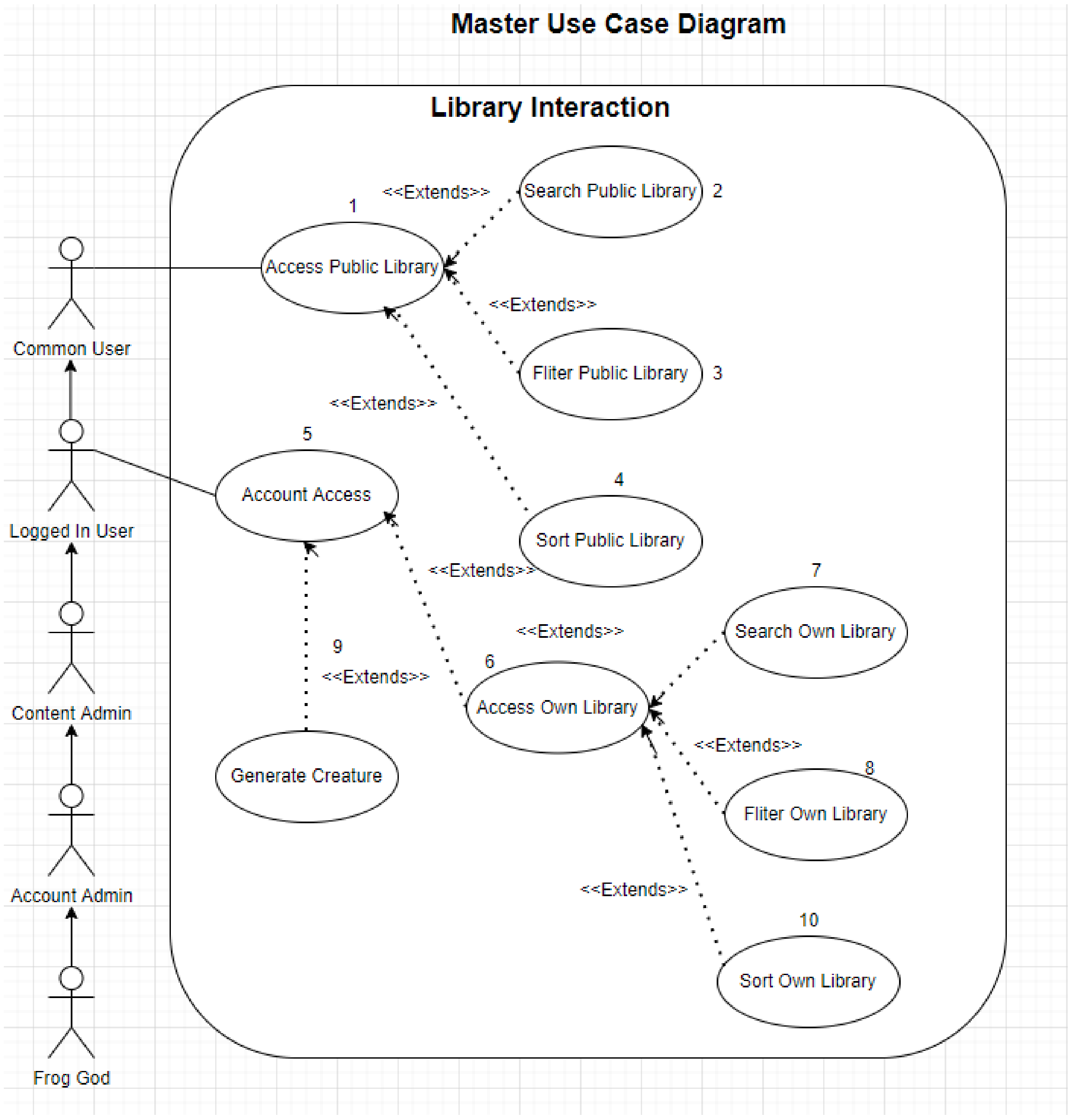
Frog God Games Staff. (2020). *About frog god games*. Frog God Games. Retrieved October 20, 2022, from https://www.froggodgames.com/frog-crew/

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

1.2 Purpose of the Product

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, M^3. The front end of M^3 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. ✓

1.3 Product Scope



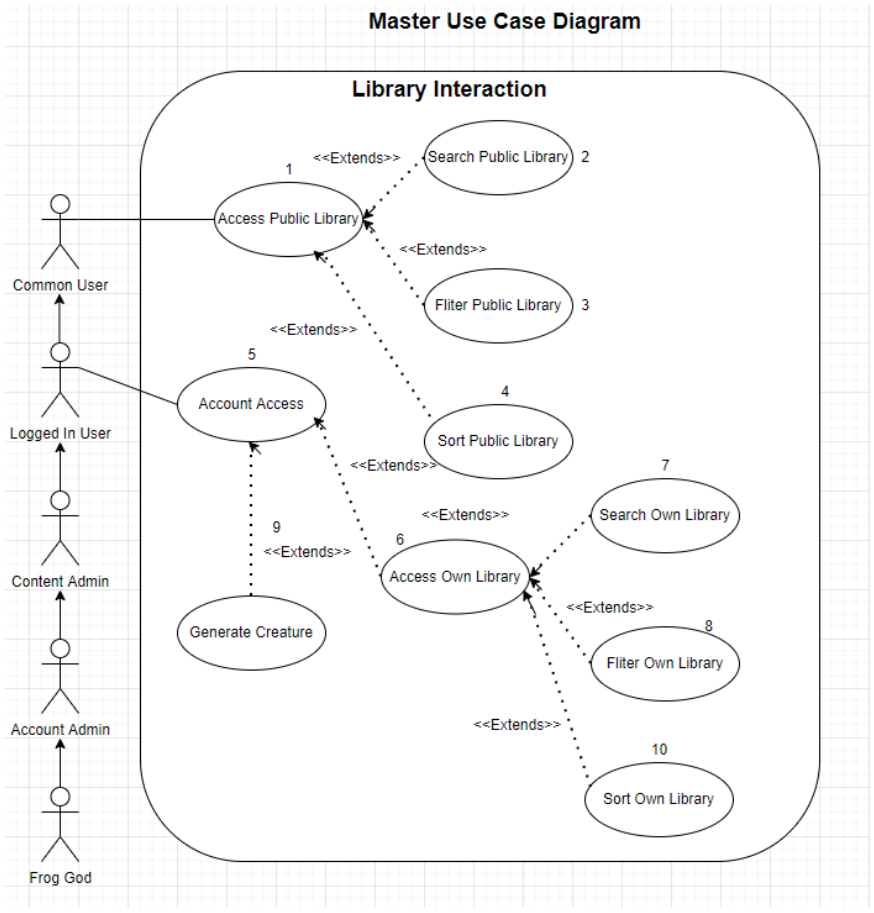
The above diagram demonstrates the boundary between the front end of M^3 and the outside world. Users are of the outside world, and interact with the system through various use cases. Our product will have four different user types. They will be a God user type for the owners of Frog God Games and developers; there will be an admin user type that can allow or deny database edits; there will be a logged in common user that has accessi~~bility~~ to their previous uploads; lastly an unlogged in common user that may only look up creatures. The system will allow ^ users to make their creations public upon request if approved by an admin.

logged on

2 Functional Requirements

The requirements listed below represent what the front end of M^3 shall be capable of. The requirements are numbered and appear in order of importance. Following the requirements will be use cases detailing how users shall interact with the system, not listed in any particular order. Beneath each use case will be a table that provides more details to the associated diagram. Each use case will also have a priority level from 1 to 5, 5 being the highest priority and 1 being the lowest priority.

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 add stat block parts to their filter by clicking on buttons
3. The system shall allow users to remove stat block parts from their filter by clicking on previously selected buttons
4. The system shall allow users to sort the public library by recently added public creatures
5. The system shall allow users to export stat blocks to a pdf
6. The system shall allow users to log in with their Frog God Games Account
7. The system shall allow users to access their own library
8. The system shall allow users to build creatures
9. The system shall allow users to customize each part of a stat block for a creature they create
10. The system shall allow users to edit creatures in their own library
11. The system shall allow users to add a clone of a creature to their own library
12. The system shall suggest "default" attributes during creature creation to expedite the process
13. The system shall allow users to request a creature in their own library to be made public
14. The system shall allow content admins to approve or deny creature publication requests.
15. The system shall allow account admins to freeze creature creation for specific accounts
16. The system shall allow users to filter the public library by author
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
19. The system shall allow users to filter their own library
20. The system shall allow users to sort their own library by recently added creatures
21. The system shall allow users to search their own library for keywords
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



Number	1	
Name	Access Public Library	
Summary	A common user is able to access the public library of content	
Priority	3	
Preconditions	A common user is able to access the website	
Postconditions	A common user is able to view the content in the public library	
Primary Actor	Common user	
Secondary Actors	None	
Trigger	A common user enters the website	
Main Scenario	**Step**	**Action**
	1	The system displays the public facing front page
	2	The common user selects the “Public Library” tab

	3	The system displays the public library of content
Extensions	**Step**	**Branching Action**
Open Issues	Lack of access to current database	

Number	2	
Name	Search Public Library	
Summary	A common user is able to search the public library for content	
Priority	3	
Preconditions	A common user is able to access the public library	
Postconditions	The common user is able to view a library of searched content	
Primary Actor	Common User	
Secondary Actors		
Trigger	The common users selects the “Public Library” tab	
Main Scenario	**Step**	**Action**
	1	The system displays the public library of content
	2	The common user enters a search term into the search box
	3	The system displays content related to the search term
Extensions	**Step**	**Branching Action**
Open Issues	Lack of access to current database	

Number	3
Name	Filter Public Library
Summary	Allow common users to filter the database based on stat blocks and creature

	attributes.	
Priority	3	
Preconditions	User accesses public database	
Postconditions		
Primary Actor	Common User	
Secondary Actors		
Trigger	User selects filter button	
Main Scenario	**Step**	**Action**
	1	System returns a viable list of creature stats to search
	2	User selects parts of a stat block to search
	3	System returns list of creatures with matching stats
	4	User selects creature
Extensions	**Step**	**Branching Action**
Open Issues	Lack of access to current database	

Number	4	
Name	Sort Public Library	
Summary	Allow common user to sort public creature creatures alphabetically or chronologically	
Priority	2	
Preconditions	Common User accesses public database	
Postconditions	A list of sorted creatures are returned to the user	
Primary Actor	Common User	
Secondary Actors		
Trigger	User selects “sort by” button	
Main Scenario	**Step**	**Action**

	1	System returns alphabetical and chronological option
	2	User selects
	2a	User selects alphabetical option
	2b	User selects chronological option
	3	System returns list of creatures based on sort option
Extensions	**Step**	**Branching Action**
Open Issues	Lack of access to current database	

Number	5	
Name	Account Access	
Summary	A user is able to access their account through their Frog God Games account	
Priority	5	
Preconditions	The Common User already has a Frog God Games Account	
Postconditions	The user is able to access Logged In User privileges and their own content	
Primary Actor	Common User and Logged In User	
Secondary Actors	Common user and Logged in User	
Trigger	A common user clicks “sign in” button	
Main Scenario	**Step**	**Action**
	1	The system displays the sign in section
	2	The user types in their username and password
	3a	The system grants the user logged in user stats and access to that accounts local library
	3b	The system returns an error “Incorrect username or password”
Extensions	**Step**	**Branching Action**
	3a	User clicks “my own library” Access Own Library

	3a	User clicks “Create my own creature” Generate Creature
Open Issues	Database schema(s)	
	Design and layout of front end	
	Lack of access to the current database	

Number	6	
Name	Access own library	
Summary	Allows Logged In User to access their own library of unpublicized creatures	
Priority	5	
Preconditions	Account Access	
Postconditions	User sees their library of creatures	
Primary Actor	Logged In User	
Secondary Actors		
Trigger	User logs in and selects “own library”	
Main Scenario	**Step**	**Action**
	1	System presents user with created library
	2	User sees their own library of creatures
Extensions	**Step**	**Branching Action**
8	2	Filter own library
7	2	Search Own Library
10	2	Sort Own Library
Open Issues	Database schema(s) Design and Layout of front end Lack of access to current database	

Number	7	
Name	Search Own library	
Summary	The logged in user is able to search their own library for content	
Priority	3	
Preconditions	The logged in user is able to access their own library of content	
Postconditions	The logged in user is able to view their library of searched content	
Primary Actor	Logged in user	
Secondary Actors	None	
Trigger	The logged in user clicks “Private library” tab	
Main Scenario	**Step**	
	1	The system displays the private library of the user
	2	The logged in user enters a search term into the search box
	3	The system displays content related to the search term from the users private library
Extensions	**Step**	
		N/A
Open Issues		
	Database schema(s)	
	Design and layout of front end	
	Lack of access to the current database	

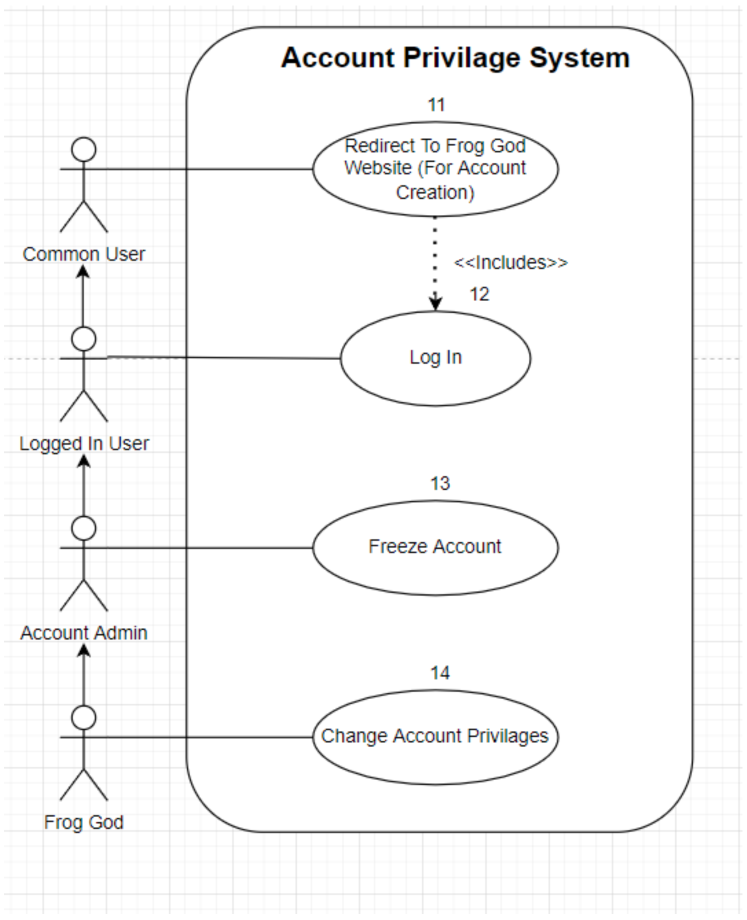
Number	8
Name	Filter Own Library
Summary	A logged in user is able to filter their own library based on stat blocks and creature
Priority	3
Preconditions	Logged in user is able to access their private library
Postconditions	The logged in user views a filtered version of their private library
Primary Actor	Logged in user

Secondary Actors	None	
Trigger	The logged in user clicks “private library” tab	
Main Scenario	**Step**	
	1	The system displays the private library of the user
	2	User selects stat blocks to filter creatures
	3	The system displays a list of creatures matching selected stat block
Extensions	**Step**	
		N/A
Open Issues		
	Database schema(s)	
	Design and layout of front end	
	Lack of access to the current database	

Number	9	
Name	Creature Generation	
Summary	A Logged In User creates a creature by filling out the parts of a stat block	
Priority	4	
Preconditions	The user is logged in and has access to their account. The user is looking at the “generate creature” page	
Postconditions	Creature is created locally in user’s own library	
Primary Actor	Logged In User	
Secondary Actors	None	
Trigger	User selects “edit creature” from home page	
Main Scenario	**Step**	**Action**
	1	The system displays sample of what the creature’s page will look like
	2	User edits any parts of the stat block they would like to and presses “confirm”
	3	System displays message confirming the creation of a creature
Extensions	**Step**	**Branching Action**

	3a	User clicks “Edit Creature” after returning to their own library: Edit Creature in Local Library
	3b	User clicks “Request Publication” Request Creature to Global Library
Open Issues		
	The Moonwake Development team can not access the database at this time ✓	

Number	10	
Name	Sort Own Library	
Summary	A logged in user is able to sort their own library based alphabetical order or chronological order	
Priority	1	
Preconditions	A logged in user is viewing a search or filtered list of creatures	
Postconditions	The logged in user has soreded a filtered or searched list of creatures	
Primary Actor	Logged in user	
Secondary Actors	None	
Trigger	The logged in user clicks “sort”	
Main Scenario	**Step**	
	1	The system displays the list to be sorted
	2	User selects what they want to sort the list by
	3	The system displays the list in the selected sorted order
Extensions	**Step**	
		N/A
Open Issues		
	Database schema(s)	
	Design and layout of front end	
	Lack of access to the current database	



Number	11	
Name	Redirect To Frog God Website (For Account Creation)	
Summary	Common User creates account via Frog God Games website	
Priority	4	
Preconditions	Common user doesn’t already have a Frog God Games account	
Postconditions	Common user becomes a registered user	
Primary Actor	Common user	
Secondary Actors	None	
Trigger	User clicks “Create Account” button on public facing front page	
Main Scenario	**Step**	**Action**
	1	The system redirects the user to the Frog God Games website

Extensions	**Step**	**Branching Action**
		N/A
Open Issues	How to use current Frog God account system	

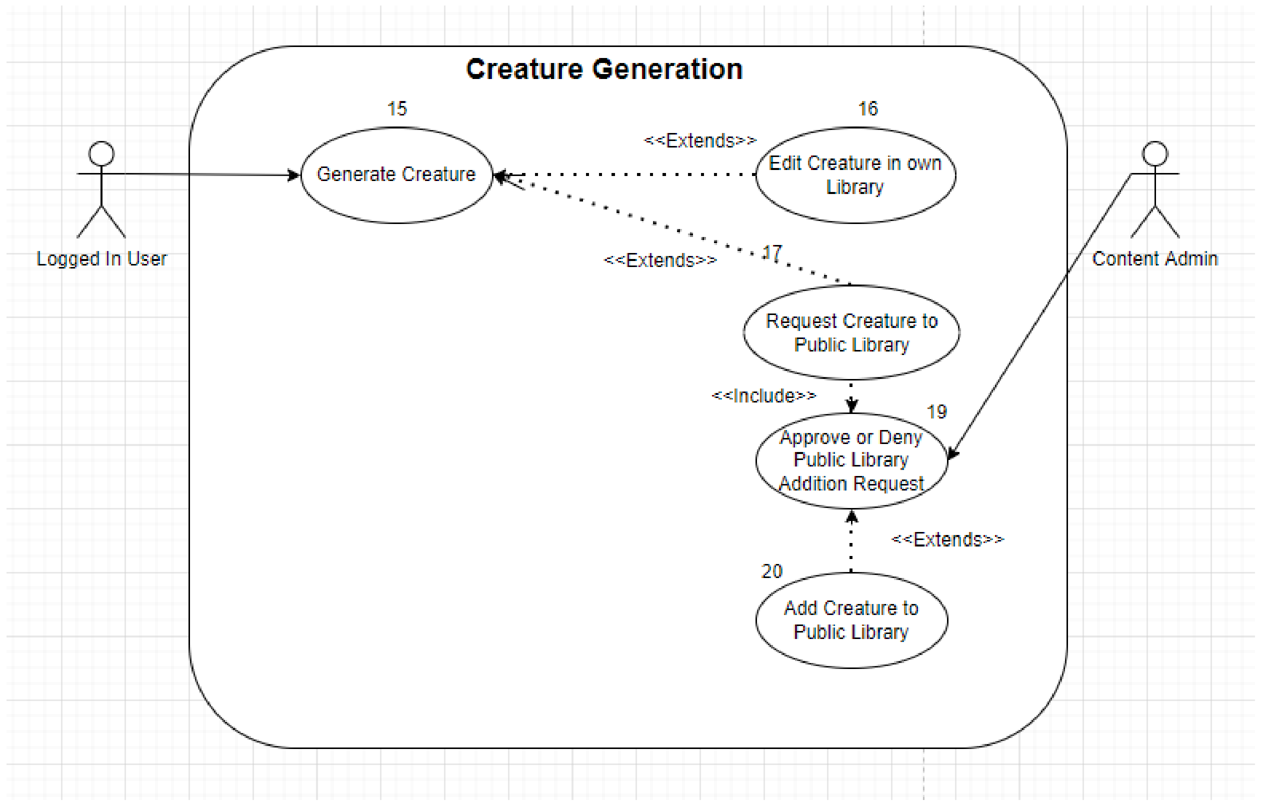
Number	12	
Name	Login	
Summary	Common user logs in to become a logged in user	
Priority	5	
Preconditions	Common user already has a Frog God Games account	
Postconditions	Common user becomes a logged in user	
Primary Actor	Common user	
Secondary Actors	None	
Trigger	User clicks “login” from public facing front page	
Main Scenario	**Step**	**Action**
	1	The system displays login page with email and password form
	2a	User enters their Frog God Games account information
	2b	User clicks “cancel” button
	3a	The system displays “login successful” and redirects user to home page
	3b	The system redirects the user to public front page
Extensions	**Step**	**Branching Action**
		N/A
Open Issues	How to use current Frog God account system	

Number	13
Name	Freeze Account
Summary	Account Admin is able to freeze a users account
Priority	4

Preconditions	Account Admin is logged in to the website	
Postconditions	A user is no longer able to create creatures using their account	
Primary Actor	Account Admin	
Secondary Actors	None	
Trigger	Account Admin clicks “Users” tab	
Main Scenario	**Step**	**Action**
	1	The system displays a list of all user accounts
	2a	Account Admin selects a user from the list
	2b	Account Admin searches for a user from the list and selects a user
	2c	Account Admin selects a user from flagged users list
	3	The system displays menu of actions
	4	The Account Admin selects freeze account
	5	The system displays “Account Frozen” message
Extensions	**Step**	**Branching Action**
		N/A
Open Issues	How to use current Frog God account system	

Number	14	
Name	Change Account Privilege	
Summary	A Frog God is able to change the account privileges of a users account	
Priority	3	
Preconditions	A Frog God is logged in	
Postconditions	The account privileges of a user are changed	
Primary Actor	Frog God	
Secondary Actors	None	
Trigger	Frog God clicks “Users” tab	
Main Scenario	**Step**	**Action**
	1	The system displays a list of all user accounts

	2a	The Frog God selects a user from the list
	2b	The Frog God searches for a user from the list and selects a user
	3	The displays menu of actions
	4a	The Frog God selects “Add Account Admin privileges”
	4b	The Frog God selects “Add Content Admin privileges”
	5a	The system displays “User is now an Account Admin”
	5b	The system displays “User is now a Content Admin”
Extensions	**Step**	**Branching Action**
		N/A
Open Issues	How to use current Frog God account system	


Creature Generation
15
Generate Creature
16
Edit Creature in own Library
<<Extends>>
<<Extends>>
17
Request Creature to Public Library
<<Include>>
19
Approve or Deny Public Library Addition Request
<<Extends>>
20
Add Creature to Public Library
Logged In User
Content Admin

Number	15	
Name	Creature Generation	
Summary	A Logged In User creates a creature by filling out the parts of a stat block	
Priority	4	
Preconditions	The user is logged in and has access to their account. The user is looking at the "generate creature" page	
Postconditions	Creature is created locally in user's own library	
Primary Actor	Logged In User	
Secondary Actors	None	
Trigger	User selects "edit creature" from home page	
Main Scenario	**Step**	**Action**
	1	The system displays sample of what the creature's page will look like
	2	User edits any parts of the stat block they would like to and presses "confirm"
	3	System displays message confirming the creation of a creature

Extensions	Step	Branching Action
	3a	User clicks “Edit Creature” after returning to their own library: Edit Creature in Local Library
	3b	User clicks “Request Publication” Request Creature to Global Library
Open Issues	The Moonwake Development team can not access the database at this time	

Number	16	
Name	Edit Creature in own Library	
Summary	Logged In User edits an already existing creature that has not yet been added to the database	
Priority	4	
Preconditions	User is logged in, the creature being edited exists	
Postconditions	Edited creature will be updated to match edits	
Primary Actor	Logged In User	
Secondary Actors	N/A	
Trigger	User selects “edit creature” from home page	
Main Scenario	**Step**	**Action**
	1	System displays list of creatures stored locally that can be edited
	2	User selects a creature from the list
	3	System displays the form for editing selected creature
	4	User fills out the form
	5	System asks for confirmation
	6	User selects “confirm”
	7	System displays updated creature page
	8	User sees updated page
Extensions	**Step**	**Branching Action**
		N/A
Open Issues	The Moonwake Development team can not access the database at this time	

Number	17	
Name	Request Creature to Public Library	
Summary	Logged In User submits a request to have a creature in their own library made public	
Priority	4	
Preconditions	User is logged in, the creature being requested exists	
Postconditions	A request is sent to an admin to make User's creature public	
Primary Actor	Logged In User	
Secondary Actors	N/A	
Trigger	User selects "request publication" from home page	
Main Scenario	**Step**	**Action**
	1	System displays request form
	2	User completes request form and presses "submit"
	3	System displays message confirming their request has been sent
	4	User sees confirmation message
Extensions	**Step**	**Branching Action**
		N/A
Open Issues	The Moonwake Development team can not access the database at this time	

Number	19
Name	Approve or Deny Public Library Addition Request
Summary	An admin can approve or deny a request sent from a logged In user
Priority	4
Preconditions	User is an Admin, is Logged In, and the request being looked into exists
Postconditions	The user that sent the request will be notified of their approval/denial
Primary Actor	Admin
Secondary Actors	N/A

Trigger	Admin selects "view publication requests" from home page	
Main Scenario	**Step**	**Action**
	1	System displays list of publication requests
	2	Admin selects a request
	3	System displays details of said request
	4a	Admin selects "approve"
	4b	Admin selects "deny"
	5a	System notifies admin and the user that sent the request that the request has been approved
	5b	System notifies admin and the user that send the request that the request has been denied
	6	Admin sees notification
Extensions	**Step**	**Branching Action**
	4a. 1	Upon approval, admin selects "make public" Add Creature to Public Library
Open Issues	The Moonwake Development team can not access the database at this time	

Number	20	
Name	Add Creature to Global Library Font change...	
Summary	Admin creates a creature and publishes it to the global library	
Priority	4	
Preconditions	User must be an Admin, Admin must be logged in	
Postconditions	Creature will be published and visible in global library of creatures	
Primary Actor	Admin	
Secondary Actors		
Trigger	Admin selects "publish creature" from home page	
Main Scenario	**Step**	**Action**
	1	System displays creature creation form

	2	Admin completes creature creation form
	3	System asks for confirmation
	4	Admin selects “confirm”
	5	System displays message confirming creature has been created and published to the global library
	6	Admin sees message
Extensions	**Step**	**Branching Action**
		N/A
Open Issues	• Lack of access to current database • How to communicate with database • Database schema(s)	

3 Non-Functional Requirements

The Front End of Frog God Games creature database will offer a fast and easy to use front end for the public to view, create, and share creatures with each other as well as export a PDF datasheet of selected creatures. The following non-functional requirements are to describe performance requirements, safety requirements, security requirements, privacy requirements, and software quality attributes. Each requirement will have a priority level from 1 to 5, with 5 being the highest priority and 1 being the lowest.

3.1 Performance Requirements

- **REQ1:** Priority Level 5: The system shall be available 23 hrs a day, 7 days a week to all users excluding when an exterior tool is down (Suchs as AWS or Shopify)
- **REQ2:** Priority Level 4: The system shall not take more than 1 seconds to respond to a user request 90% of the time[1]
- **REQ3:** Priority Level 2: The system shall not take more than 5 seconds to export a PDF of stat blocks for selected creatures
- **REQ4:** Priority Level 4: The system shall not request information from the server unless specifically requested by the user

3.2 Safety Requirements

- **REQ5:** Priority Level 1: The system shall flag inappropriate terms using a profanity detector to warn admins of possible inappropriate content before publication.
- **REQ6:** Priority Level ~~3~~: The system shall require an admin check in order to accept a users request to publicize their creature
  5

3.3 Security Requirements

- **REQ7:** Priority Level 5: The system shall prevent unwanted changes to the database by individuals without proper permission

3.4 Privacy Requirements

- **REQ8:** Priority Level 2: The system shall not collect location data
- **REQ9:** Priority Level 1: The system shall not collect use time data

Not sure this is meaningful. What is the size of "potential users"

3.4 Software quality attributes

- **REQ10:** Priority Level 3: The system shall allow 90% of potential users to use the application at once

3.5 Non-Functional Requirements Testing

- **TEST1:** A test to see if the system is responsive every 30 minutes by pinging every part of the system
- **TEST2:** A selenium tester will test the system as a user for responsiveness every 30 minutes by taking an action on the system
- **TEST3:** A selenium tester will test the system as a user by attempting to export a creature to PDF every 30 minutes
- **TEST4:** A test will count the number of times the server is pinged from a developer version that has no pings in that time frame.
- **TEST5:** The system shall make a creature with an inappropriate string as a name and see if the profanity detector picks it up

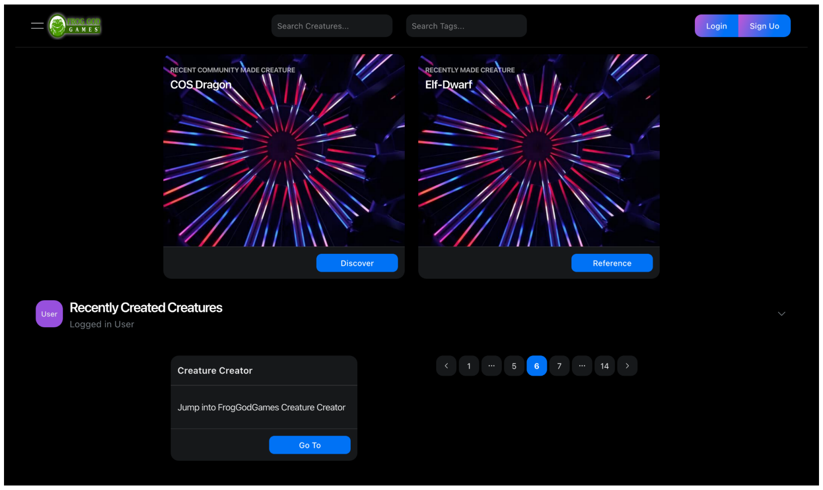
I'll need this term explained. Also if this pinging/testing is a permanent function.

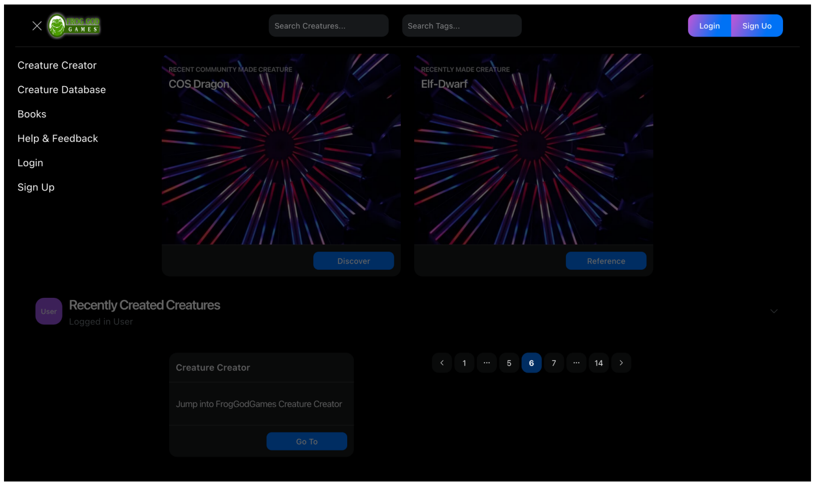
- **TEST6:** A test to see if the system is responsive every 30 minutes by pinging every part of the system
- **TEST7:** A test to ensure that the common user can not alter the public database
    - **TEST7.1:** A test to ensure that a content admin can not freeze accounts
- **TEST8:** A selenium tester that will test to see if the server requests location data
- **TEST9:** A selenium tester that will test to see if the server requests use time data
- **TEST10:** A tester will see if the system completes

We add discussed and I would like to add back in a fuse that freezes an account from uploading data if a certain rate and volume quantity is exceeded. We'd also discussed having a volume cap for non-paying users, with pay per month option for users that want to store more data (including possibly their own art).

4 User Interface

A demo application is available on our development Github repository. Provided screenshots describe a possible user interface to maximize user experience, while at the same time making the application simple to use.





We've discussed this prototype and expect that it will change significantly.

5 Deliverables

Deliverables include everything that will be produced during this project. Deliverables include documents as well as source code and the final product. The following table lists out the deliverables that are expected to be produced from this project. The table specifies the name of the deliverable as well as the date each deliverable will be produced by and the format that it will be available as.

Deliverable:	Format:	Date:
System Requirement Specifications	Hardcopy/PDF	10/20/22
System Design Document	Hardcopy/PDF	11/1/22
User Interface Design Document	Hardcopy/PDF	11/29/22
Critical Design Review Document	Hardcopy/PDF	12/15/22
User Manual	Hardcopy/PDF	Spring 2023
Administrator Manual	Hardcopy/PDF	Spring 2023
Bi-Weekly Reports	PDF or JPEG	Bi-Weely Starting 10/25/22
All Source Code	Electronic via GitHub	Updated continuously
The Final Product	Website	Spring 2023

6 Open Issues

Open issues are issues that have been raised and do not yet have a conclusion. These issues will be addressed later in the development process. Some of these issues are included below.

- Lack of access to current database
- How to communicate with database
- Database schema(s)
- How the front end will be hosted
- Final decision regarding language for front end
- Design and Layout of front end
- ORM (Object Relational Mapper) for JavaScript/TypeScript use for database (for security - to avoid sql injections)
- How to use Frog God Website accounts

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* [illegible]	Date 10/24/2022

Comments: With changes as noted in document.

Team:		
Matthew Virgin	x Matthew Virgin	Date 10/25/2022
Cedric Fahey	x [illegible]	Date 10/25/2022
David DiFrancalo	x [illegible]	Date 10/26/2022
Tristan Zippert	x Tristan Zippert	Date 10/25/2022
Landon Thibodeau	x [illegible]	Date 10/26/2022

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 Matthew Virgin	x_*Matthew Virgin*	Date 10/20/2022
Comments: N/A		
X Tristan Zippert	x_*Tristan Zippert*	Date 10/20/2022
Comments: N/A		
X Landon Thibodeau	x *Landon Thibodeau*	Date 10/20/2022
Comments: N/A		
X__Cedric Fahey	x__*Cedric Fahey*	Date 10/20/2022
Comments: N/A		
X__David DiFrumolo	x__*David DiFrumolo*	Date 10/20/22
Comments: N/A		

Appendix C: Document Contributions

- **Matt Virgin:** Use Case Descriptions, 33%. Functional Requirements, 60%.
- **David:** Non-Functional Requirements 30% Appendix B 100%. Appendix A 50%
- **Cedric:** Use case Diagrams 100%, Use Case Descriptions 33%. Functional Requirements, 20%. Appendix A 50%
- **Landon:** Non-Functional Requirements 70%, Use Case Descriptions 33%, Open Issues 100%, Deliverables 20%
- **Tristan:** Deliverables, 20%. User-interface, 100%. Functional Requirements, 35%, Introduction 100%, Use Cases: 30%