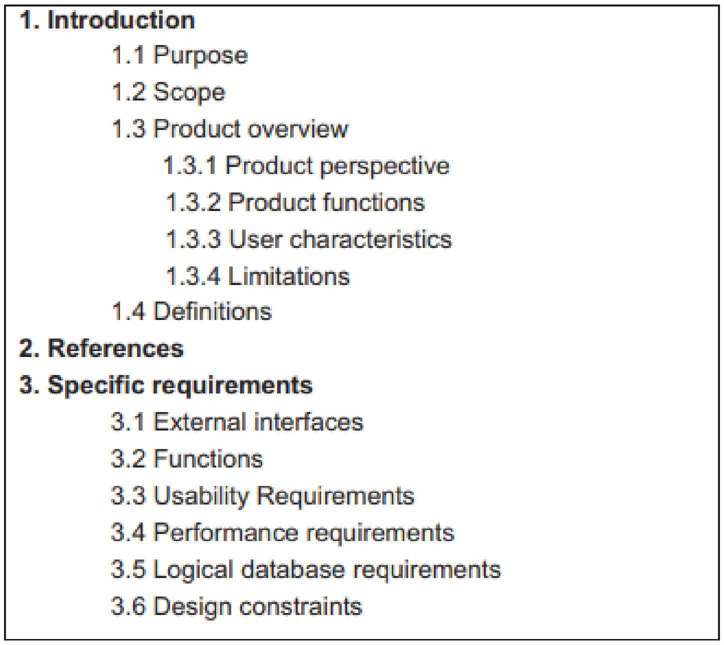
Universidad EAFIT
Ingeniería de Sistemas
Proyecto Integrador 1

Deliverable 1: Project definition and Software Requirements Specification

Create a GitHub Wiki page (NOT an attachment) with the items requested in this template.

All notes containing figures, explanations, and examples for filling out the template should be deleted and replaced by the project information. This template is based on ISO/IEC/IEEE 29148:2011, which is an international standard that indicates how to define system and software requirements.

The software requirements specification (SRS) is a specification for a particular software product that performs certain functions in a specific environment. The content is organized as follows:



The following is a summary of what each section should contain. For more information, please refer to the documents:

- 29148-2011-Requirements specification template -> section 8.4 Software requirements specification document (page 53) and section 9.5 Software requirements specification (SRS) document (page 62)
- Towards a new template for the specification of requirements in semi-structured natural language (available at: https://www.researchgate.net/publication/339396006_Towards_a_new_template_for_the_specification_of_requirements_in_semi-structured_natural_language)

For examples see the following documents:

- Example - EIP_RequirementsSpecificationGLA_ V2-5
- 2020-Example Software Requirements Specification Document for ReqView

1. Introduction

1.1. Purpose

It is widely recognized that fish are animals with very specific care needs. Most people, at times, not even the owners, are aware of how to provide them with the care they require. That's why we are committed to developing a platform that automates your aquarium environment based on the types of fish you have. We will implement artificial intelligence to care for your fish, detecting any potential diseases or aquarium malfunctions. Furthermore, in our fish store, we will offer a variety of products such as live fish, aquariums, and any other accessory a fish owner might need with the added benefit that our store will provide you with all the specifications for the product and will align perfectly with the automated aquarium environment. With this project we aim to simplify and enhance the experience that can be owning a fish and caring for it, providing users with the tools needed to ensure the well-being of their aquatic pets.

1.2. Scope

The scope of the software under consideration, the Aquatech Management System (AMS), encompasses a comprehensive suite of functionalities aimed at facilitating efficient aquarium management. AMS will automate control over environmental parameters like temperature and lighting, monitor water quality, integrate AI for fish health analysis, and offer remote access for management tasks. By providing these capabilities, AMS seeks to streamline aquarium management processes for both hobbyists and professionals, enhancing user convenience, promoting fish health, and delivering a seamless shopping experience through its integrated virtual store. This scope is consistent with higher-level specifications, ensuring coherence and alignment across all project documentation.

1.3. Product Overview

1.3.1. Product perspective

1. System Interfaces

- Our Django project will interact with the SQLite3 database for user authentication and storage of application data.
- Integration with the external bank's payment gateway will involve interfacing with their API for processing online payments.

2. User Interfaces

- We’ll ensure that the user interface of the Django application is seamless and consistent with Django's built-in authentication system for user login, registration, and profile management.
- The payment interface will provide a smooth checkout experience for users, integrating with the external bank's payment gateway for secure transactions.

3. Software Interfaces

- Our Django project may utilize various Django packages and libraries for additional functionality, such as
- crispy_forms
- django

- django-crispy-forms
- crispy_bootstrap4
- Pillow

- Integration with the external bank's payment gateway will require adherence to their API specifications.

4. Communications Interfaces

- Communication between **the** Django application and the SQLite3 database will typically be handled through Django's ORM (Object-Relational Mapping) layer.
- Integration with the external bank's payment gateway will involve HTTP requests to their API endpoints for processing payment transactions.

5. Operations

- Our Django application will be designed for scalability, with considerations for handling increased user traffic and database load.
- We'll implement error handling and logging mechanisms to monitor and troubleshoot issues related to database operations and external API integrations.
- We'll ensure that my application meets security standards for handling sensitive user data and processing online payments.

1.3.2. Product functions

1. User Authentication

- Allow users to register new accounts.
- Enable existing users to log in securely.
- Implement password recovery mechanisms for forgotten passwords.

2. Store Functionality

- Display a catalog of available fish and accessories for sale.
- Provide filtering and sorting options for easy navigation.
- Allow users to add items to their shopping cart.
- Support secure checkout and payment processing through the integrated payment gateway.

3. Aquarium Management

- Enable users to view their aquarium(s) and its current conditions.
- Allow users to add and update aquarium conditions, such as temperature, pH level, etc.
- Provide recommendations for suitable fish based on the aquarium's conditions.
- Support the addition and management of multiple aquariums for users with multiple setups.

4. Profile Management

- Allow users to view and update their profile information.
- Provide options for users to manage their preferences and settings.
- Implement user roles and permissions for admin functionalities (if applicable).

5. Customer Service Interaction

- Offer a chat feature for users to communicate with customer service representatives.
- Provide a form or interface for users to submit questions or support requests.

6. Home Page Navigation

- Present information about the company, its mission, and vision on the home page.
- Offer clear navigation options to access different sections of the website (store, aquarium, profile, services).

7. Integration with External Systems

- Interface seamlessly with the SQLite3 database for user data management and authentication.
- Communicate with the external bank's payment gateway for processing online payments securely.

8. System Administration

- Provide administrative functionalities for managing user accounts, products, and orders.
- Support monitoring and logging capabilities for system administrators to track user activities and system performance.

9. Monitoring Sick Fish

- Integrate functionality for automatic monitoring of fish in smart aquariums using artificial intelligence (AI).
- Utilize an embedded camera to capture real-time images of the fish.
- Implement image recognition algorithms to detect signs of diseases or abnormalities in the fish.
- Send automatic alerts to users when a sick fish is detected, indicating the possible illness and recommending treatment actions.

10. Recommendations Based on Aquarium Specifications

- Analyze the specifications of users' aquariums, including size, water parameters, and existing fish population.
- Provide personalized recommendations for compatible fish species and accessories that suit the user's aquarium setup.
- Assist users in making informed decisions about adding new fish or accessories to their aquariums based on their specific requirements and constraints.

1.3.3. User characteristics

Aquarium Enthusiasts

- Individuals with a keen interest in fishkeeping and aquarium maintenance. These users may have varying levels of experience, from beginners to seasoned hobbyists.

Pet Owners

- Individuals who own fish as pets and are invested in their well-being. They may be seeking solutions to effectively manage their aquariums and ensure the health and happiness of their fish.

Tech-Savvy Users

- Users who are comfortable with technology may appreciate advanced features and functionalities. They may seek innovative solutions for monitoring and managing their aquariums using digital tools.

Novice Fish Keepers

- Individuals who are new to fishkeeping may require guidance and support in setting up and maintaining their aquariums. These users may benefit from user-friendly interfaces and educational resources to enhance their fishkeeping experience.

1.3.4. Limitations

Regulatory Policies:

Due to regulatory policies in our country, there are limitations on the types of fish that can be sold through our platform. Certain species of fish may be protected or restricted from sale due to conservation efforts, endangered status, or potential ecological impact. Compliance with these regulatory policies is essential to ensure legal compliance and environmental sustainability. As a result, our platform will only offer fish species that are permitted for sale according to the regulations set forth by relevant authorities. We will implement measures to verify the legality of fish species offered for sale on our platform, including collaboration with reputable suppliers and adherence to documentation requirements. Users will be informed of the restricted availability of certain fish species and encouraged to make responsible choices in their purchases to support conservation efforts and comply with regulatory policies.

Limitation: Safety and Security Considerations

Ensuring the safety and security of users' personal information and financial transactions is paramount. Compliance with data protection laws and regulations imposes constraints on data handling and storage practices. Implementation of robust security measures, including encryption, access controls, and regular security audits, may limit certain options in terms of system architecture and technology selection. User authentication and authorization mechanisms must be carefully designed to prevent unauthorized access and protect user privacy. Vulnerability management and incident response procedures are essential to mitigate security risks and ensure the integrity of the system. Balancing security requirements with usability and performance considerations may pose challenges in system design and implementation.

Limitation: Hardware Limitations

The hardware capabilities of users devices may impose limitations on the performance and functionality of the software. Compatibility with various devices, including computers, smartphones, and tablets, may require optimization efforts to ensure consistent user experience across different platforms. Users without a domotic aquarium setup may experience limited options and functionalities compared to those with such systems. Certain features and functionalities specific to domotic aquarium management, such as automated monitoring and control, may not be available for users without compatible hardware. Compatibility with older or less powerful devices may limit the use of advanced features or require fallback options to ensure usability for all users. Consideration of hardware limitations is essential to provide an inclusive and accessible experience for users with diverse devices and capabilities, while also managing expectations regarding the availability of certain features based on the user's setup.

2. References

Input format:

- User credentials to access their account that keeps activity history to improve recommendations.
- A secure payment system for the user to make purchases all in one place.
- User inputs for the desired parameters ranges of the aquarium, e.g., water temperature, turbidity, or ambience lights, through the web dashboard.
- Sensor data from the aquarium that monitors the fish tank environment and sends it to a database.
- A list of products that the user selects to purchase while navigating the store.
- A map location with a public aquarium that the user chooses to see the details of

Output format:

- Recommendations on different care products depending on the state of the fish tank.
- A world map that shows the locations of all the public fish tanks
- Real time visualization of the fish tank variables through the web dashboard.
- Alert notifications in case of abnormal sensor readings.
- Details of a public fish tank from the map including a user defined description.

The software aims to address the following challenges that fish owners may face when taking care of their tank environment:

- Maintaining an optimal water environment. Too many fluctuations in temperature, pH levels, or turbidity may lead to stress or even harm to the fish.
- Separating the time needed to make proper monitoring and maintenance to the tank.
- Delayed response to possible issues due to not being able to always monitor the environment.
- Failure to control the parameters of the environment accurately manually.
- Having to outsource all the components and products from multiple places with varying quality.

The storage of sensitive user data like login credentials or bank data is to be encrypted and transferred using secure communication protocols to protect it from unauthorized access.
The location data of each aquarium is to always be stored; however, it shall not be posted publicly without the user's authorization, this includes the world map.
Furthermore, the user must always have the option to disable and to delete their history tracking.

The code that powers the software is open source and falls under the GPL-3 license, modification or redistribution of the software is allowed, only if you're providing the same freedom that was given to you, patenting the program to render it non-free is specially forbidden.

3. Specific requirements

Each requirement shall possess the following characteristics:

- **Necessary.** The requirement defines an essential capability, characteristic, constraint, and/or quality factor. If it is removed or deleted, a deficiency will exist, which cannot be fulfilled by other capabilities of the product or process.
- **Implementation Free.** The requirement, while addressing what is necessary and sufficient in the system, avoids placing unnecessary constraints on the architectural design. The objective is to be implementation independent. The requirement states what is required, not how the requirement should be met.
- **Unambiguous.** The requirement is stated in such a way so that it can be interpreted in only one way. The requirement is stated simply and is easy to understand.

- **Consistent.** The requirement is free of conflicts with other requirements.
- **Complete.** The stated requirement needs no further amplification because it is measurable and sufficiently describes the capability and characteristics to meet the stakeholder's need.
- **Singular.** The requirement statement includes only one requirement with no use of conjunctions.
- **Feasible.** The requirement is technically achievable, does not require major technological advances, and fits within system constraints (e.g., cost, schedule, technical, legal, regulatory) with acceptable risk.
- **Traceable.** The requirement is upwards traceable to specific documented stakeholder statement(s) of need, higher tier requirement, or other source (e.g., a trade or design study). The requirement is also downwards traceable to the specific requirements in the lower tier requirements specification or other system definition artefacts. That is, all parent-child relationships for the requirement are identified in tracing such that the requirement traces to its source and implementation.
- **Verifiable.** The requirement has the means to prove that the system satisfies the specified requirement. Evidence may be collected that proves that the system can satisfy the specified requirement. Verifiability is enhanced when the requirement is measurable.

Requirements should state 'what' is needed, not 'how'. Vague and general terms shall be avoided. They result in requirements that are often difficult or even impossible to verify or may allow for multiple interpretations. The following are types of unbounded or ambiguous terms:

- Superlatives (such as 'best', 'most')
- Subjective language (such as 'user friendly', 'easy to use', 'cost effective')
- Vague pronouns (such as 'it', 'this', 'that')
- Ambiguous adverbs and adjectives (such as 'almost always', 'significant', 'minimal')
- Open-ended, non-verifiable terms (such as 'provide support', 'but not limited to', 'as a minimum')
- Comparative phrases (such as 'better than', 'higher quality')
- Loopholes (such as 'if possible', 'as appropriate', 'as applicable')
- Negative statements (such as statements of system capability not to be provided)

To describe each requirement, you should use the requirements writing template presented in section 5 Proposing a new requirements specification template (see: Towards a new template for the specification of requirements in semi-structured natural language or https://www.researchgate.net/publication/339396006_Towards_a_new_template_for_the_specification_of_requirements_in_semi-structured_natural_language).

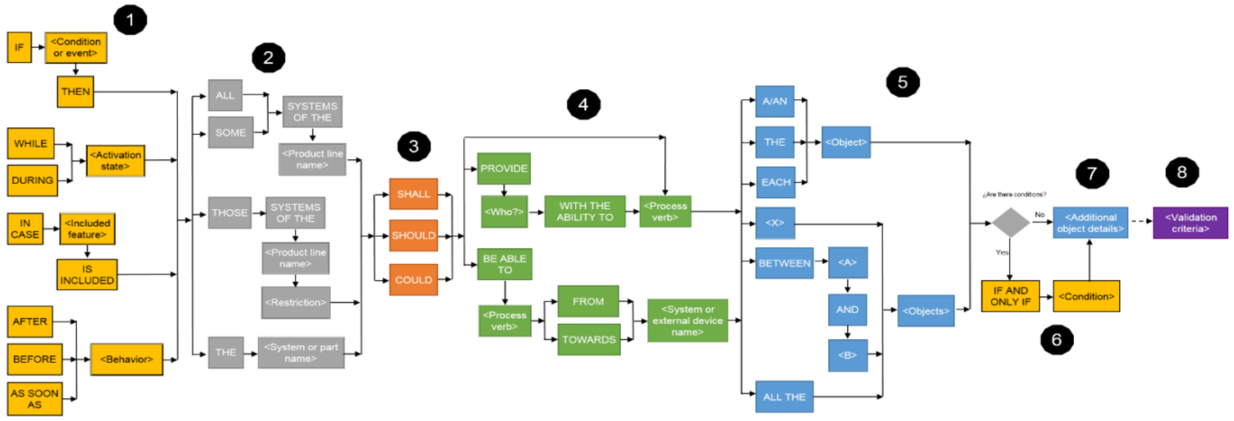


Figure 3. Mazo & Jaramillo template.

Identify each requirement with a unique code. For example, FR01 (for functional requirements), PR02 (for performance requirements), UR03 (for usability requirements) ...

3.1. External interfaces

1. Login Interface
a) Item Name: Login Interface
b) Purpose Description: Allows users to authenticate into the system.
c) Input Source or Output Destination: Users
d) Valid Range, Accuracy, and/or Tolerance: N/A
e) Units of Measure: N/A
f) Timing: Activated at user session initiation.
g) Relationships to Other Inputs/Outputs: Successful authentication grants access to other system functionalities.
h) Screen Formats/Organization: Form with fields for username and password.
i) Window Formats/Organization: Modal popup interface or dedicated page.
j) Data Formats: Plain text for username and password.

k) Command Formats: N/A
l) End Messages: Login success or error messages.

2. Store Interface
a) Item Name: Store Interface
b) Purpose Description: Allows users to purchase fish and accessories for their aquarium.
c) Input Source or Output Destination: Users
d) Valid Range, Accuracy, and/or Tolerance: N/A
e) Units of Measure: N/A
f) Timing: Available after login.
g) Relationships to Other Inputs/Outputs: Users can add products to the cart and proceed to checkout.
h) Screen Formats/Organization: Product catalog with filtering and search options.
i) Window Formats/Organization: Dedicated page with product listing.
j) Data Formats: Product information (name, description, price, etc.).
k) Command Formats: Buttons for adding to cart, proceeding to checkout, etc.
l) End Messages: Purchase confirmation, error messages, etc.

3. Aquarium Interface
a) Item Name: Aquarium Interface
b) Purpose Description: Allows users to view and manage their aquarium.
c) Input Source or Output Destination: Users
d) Valid Range, Accuracy, and/or Tolerance: N/A
e) Units of Measure: N/A
f) Timing: Available after login.
g) Relationships to Other Inputs/Outputs: Users can add aquarium conditions and receive fish recommendations.
h) Screen Formats/Organization: Aquarium view with options to add conditions and view recommendations.
i) Window Formats/Organization: Dedicated page with aquarium information.
j) Data Formats: Aquarium conditions (temperature, pH, etc.), fish recommendations.
k) Command Formats: Buttons for adding conditions, saving changes, etc.
l) End Messages: Confirmation of changes, error messages, etc.

4. Services Interface
a) Item Name: Services Interface
b) Purpose Description: Allows users to chat with customer service representatives and ask

questions.
c) Input Source or Output Destination: Users
d) Valid Range, Accuracy, and/or Tolerance: N/A
e) Units of Measure: N/A
f) Timing: Available on the home page or through a dedicated section.
g) Relationships to Other Inputs/Outputs: Users can interact with customer service representatives in real-time.
h) Screen Formats/Organization: Chat interface or form for submitting questions.
i) Window Formats/Organization: Popup chat window or dedicated page.
j) Data Formats: Text input for chatting or submitting questions.
k) Command Formats: Send button for sending messages or questions.
l) End Messages: Chat session end confirmation or response from customer service.

5. Home Interface
a) Item Name: Home Interface
b) Purpose Description: Provides information about the company and options for accessing various sections of the website.
c) Input Source or Output Destination: Users
d) Valid Range, Accuracy, and/or Tolerance: N/A
e) Units of Measure: N/A
f) Timing: Available upon visiting the website.
g) Relationships to Other Inputs/Outputs: Serves as the main navigation hub for accessing other sections.
h) Screen Formats/Organization: Introduction to the company, with navigation options displayed prominently.
i) Window Formats/Organization: Landing page or homepage layout.
j) Data Formats: Text and images for company information.
k) Command Formats: Clickable buttons or links for accessing different sections (store, aquarium, profile, services).
l) End Messages: N/A (It's a static page with informational content and navigation options).

3.2. Functions

WHILE creating a user account the system shall autonomously verify and ensure the uniqueness of the chosen username
WHILE filtering products the virtual store system shall display results based on applied filters.
WHEN reviewing a product the system shall display user-provided reviews and ratings.
In case of payment the system shall utilize either an API or an artisanal method based on user preference.
IN CASE the IA detects dirty water the aquarius system should be able to send an alert to the web system
DURING the exploration of the virtual store all fish in store should have detailed description
AS SOON as a user configures their aquarium the system should recommend products based on user preferences.
AFTER detecting abnormal fish behavior the syste shall alert users and identify potential diseases.
DURING the configuration of a fishtank the system shall allow users to customize tank specifications.
AFTER configuring privacy settings the users shall receive immediate confirmation of their configured preferences.

WHILE seeking assistance from chatbot the system shall promptly respond with the IA generated chatbot
AFTER assistance is provided the system shall request user feedback for continuous improvement.
WHILE configuring sensors the system shall offer real-time status updates to ensure seamless integration.
AS SOON as the user configures the feeding schedule the aquarium system shall automatically feed the fish at specified intervals, ensuring timely and consistent nourishment.
AFTER the user realice the payment THE point of sale subsystem shall automatically send the payment recimenet verification
The system shall provide the guest with the ability to place his order
THE point of sale subsystem SHALL be able to read a valid credit card from a branch's dataphone
IF the user enters incorrect credentials during login THEN THE system SHALL display an authentication error message
THE web application SHALL support user authentication including sign-up functionality
THE web application COULD PROVIDE THE users WITH THE ABILITY TO edit their profile information

THE system MUST complete the user authentication within 5 seconds under normal lad conditions

3.3. Usability requirements

Define usability (quality in use) requirements. Usability requirements and objectives for the software system include measurable effectiveness, efficiency, and satisfaction criteria in specific contexts of use.

1. Effectiveness
   - The system shall allow users to perform common tasks related to aquarium management, such as adding or updating aquarium conditions, with an accuracy rate of at least 95%.
2. Efficiency
   - The system shall enable users to complete tasks related to purchasing fish or accessories from the store within an average time of 3 minutes per transaction.
3. Learnability
   - New users shall be able to navigate the system and perform basic tasks, such as logging in and viewing their aquariums, with minimal assistance or training within 5 minutes of initial use.
4. Satisfaction
   - Users shall rate their overall satisfaction with the system's interface and features as "satisfied" or "very satisfied" on a post-interaction survey administered after each session.
5. Error Tolerance
   - The system shall provide clear error messages and guidance for recovering from errors or invalid inputs, resulting in a reduction of user errors.

3.4. Performance requirements

1. Static Numerical Requirements

a) Terminal Support

The system shall support a minimum of 1000 concurrent user terminals accessing the application simultaneously.

b) Simultaneous Users

The system shall support a minimum of 500 simultaneous users interacting with the system's various functionalities concurrently.

c) Information Handling

The system shall handle a maximum of 100 gb of data , including user profiles, aquarium configurations, and transaction records, ML models .

2. Dynamic Numerical Requirements

- The system shall process a minimum of 25 transactions per minute during normal workload conditions.
- During peak workload conditions, the system shall be capable of processing up to 100 per minute to accommodate increased user activity.
- The system shall be able to retrieve and display aquarium information for a single user within 2 seconds, even under peak load conditions.
- User authentication and authorization processes shall have a response time of no more than 2 seconds, ensuring swift access to system functionalities.
- The system shall be capable of handling a peak data load of 20 megabytes per second during periods of high user activity without degradation in performance.

3.5. Logical database requirements

Specify the logical requirements for any information that is to be placed into a database, including:
a) Types of information used by various functions;
b) Frequency of use;
c) Accessing capabilities;
d) Data entities and their relationships;
e) Integrity constraints;
f) Data retention requirements.

Each team must define **at least 5 logical database requirements**.

1. Types of Information Used by Various Functions
   - The database shall store information related to user profiles, including usernames, passwords (encrypted), email addresses, and profile preferences, to support user authentication and personalization functionalities.
   - Information about aquariums, including specifications such as size, water parameters, and fish population, shall be stored to facilitate aquarium management features.

2. Frequency of Use:
   - User profile information shall be accessed frequently during login and profile management activities, requiring efficient retrieval and update capabilities.
   - Aquarium specifications and conditions shall be accessed and updated regularly as users interact with the system to monitor and manage their aquariums.

3. Accessing Capabilities
   - The database shall support concurrent access by multiple users to retrieve and update their respective user profiles and aquarium data simultaneously.
   - Access controls shall be implemented to ensure that users can only access and modify their own profile and aquarium information, maintaining data privacy and security.

4. Data Entities and Their Relationships
   - The database shall include entities for users, aquariums, fish species, and transactions, with relationships established to link users to their respective aquariums and transactions to user accounts.
   - User profiles shall be associated with one or more aquariums, while transactions shall be linked to specific user accounts for order tracking and management purposes.

5. Integrity Constraints
   - The database shall enforce integrity constraints to maintain data consistency and accuracy, such as ensuring that each user has a unique username and that aquarium specifications fall within predefined ranges.
   - Referential integrity constraints shall be enforced to maintain relationships between entities, preventing orphaned records and ensuring data integrity.

6. Data Retention Requirements
   - User profile information shall be retained indefinitely to support ongoing user interactions and account management.
   - Aquarium data shall be retained for a minimum of 6 months to allow users to track changes in their aquarium conditions and maintain historical records of their setups.

3.6. Design constraints

A requirement that limits the options open to a designer of a solution by imposing immovable boundaries and limits (e.g., the system shall incorporate a legacy or provided system element, or certain data shall be maintained in an on-line repository). Specify constraints on the system design imposed by external standards, regulatory requirements, or project limitations

1. Compatibility with Legacy Systems
   - The system shall incorporate certain legacy components or interfaces to ensure compatibility with existing systems used by the organization.

2. Technological Constraints
   - The system design must be compatible with specific technologies or platforms mandated by the organization or project stakeholders.

3. Resource Limitations
   - The system design must consider limitations in hardware resources, such as server capacity, memory, and processing power.

4. Timeline Constraints
   - The design and development of the system must be completed within specified timeframes to meet project deadlines and organizational objectives.

5. Scalability Requirements
   - The system design must accommodate future scalability requirements, allowing for growth in user base, data volume, and system complexity.

6. Security Constraints
   - The system design must incorporate security measures to protect against unauthorized access, data breaches, and other security threats, in compliance with industry standards and best practices.

7. Interoperability Requirements
   - The system design must ensure interoperability with other systems or applications used within the organization's ecosystem, facilitating data exchange and integration.

8. User Experience Constraints
   - The system design must prioritize user experience considerations, ensuring intuitive interfaces, accessibility features, and responsiveness across different devices and platforms.

4. Video

Add the link to the video, ensuring access: YouTube, Vimeo.
All members must appear in the video.
Video length: minimum 5 minutes - maximum 10 minutes.

Content of the video:

- Name of project or team.
- Problem / opportunity / need.
- Team members.
- Top 12 functional requirements (prioritization activity must be done first).

5. Project management

Create the project in GitHub and record top 20 requirements.
Record weekly meetings (each student should answer: What did you do last week? What are you going to do this week? Are there any obstacles in the way?).
Report retrospective (the team must answer: What should we continue to do (best practices)? What should we start doing (process improvements)? What should we stop doing (process problems and bottlenecks)?).

Develop class assignments and add them to the wiki.