CPS 731 - Project Plan
Automatic Fish Feeding System

Matthew Clair	500333467
Warren Pinto	500396119
Sundar Murugesan	500415649
Brandon Cardoso	500400444

Table of Contents

1. Project Plan

1.1 Introduction

1.1.1 The Problem

Currently, the only way to feed fish is by manually walking up to the tank, determining how much food each fish needs, measuring out each portion and distributing each portion over the entire surface of the water. This requires the presence of a worker to spend a lot of time doing something that could easily be automated.

1.1.2 Client Profile

This problem challenges anyone who owns a fish tank, from small tanks at home to large display tanks at a zoo or aquarium. Everyone who owns fish must repeatedly face the task of determining how much of any given food is required to feed any amount of fish multiple times a week.

1.1.3 Proposed Solution

Our proposed solution is a system that automates the task of feeding fish by automatically measuring, and distributing each type of food for each fish on a predefined schedule. This system would be able to work on any size and shape of aquarium with little extra effort. The system would be able to store information regarding all types of fish in the aquarium, the type of food they require, the amount of food they require per feeding, and how often/when each feeding should occur.

1.1.4 Basic Strategy for Development of Solution

Our basic strategy for developing this system involves having the hardware and software components designed, developed, and tested side by side so that any problems that may arise are immediately dealt with.

1.1.5 Critical Appraisal of Solution

The system we plan on developing will successfully handle and essentially remove the problem all fish tank and aquarium owners face by automating their fish feeding. The system will automatically measure out the correct amount of food for all fish in the tank, and properly distribute the food on time so that the fish are kept healthy and and the tank well maintained.

1.2 Constraints

The users of this system will be the fish keepers at the aquarium. The keepers need to be able to select a type of food, a quantity to be fed and a frequency to be fed. They also need to have a system in place to warn them before the food runs out. The hardware system is expected to be in place for a minimum of 6 years and up to a possible of 18 years. The aquarium aims to have a major renovation in 6 to 18 years so the hardware will need to function until the renovation. The software should be able to continue to be used with a new hardware system.

As the aquarium has many exotic and expensive aquatic animals, it is critical that the system is very reliable. Short term overfeeding (maximum of 5 days) or underfeeding (maximum of 2 days) is acceptable but any longer risks the health of the fish and is not acceptable.

The users of the system will be the fish feeding staff. These users are experienced with the requirements of the aquatic life and their handling but have no prior experience with technological feeding systems. Adequate documentation and intuitive design is an important factor to make sure that the users are able to use the system without issues. After initial setup for the fish and food types, the system should be able to function without user interaction other than maintaining food levels in the store.

The system has to be compatible with existing tanks. There is no existing system for feeding the fish. Feeding times, food types and quantity are currently tracked on paper. The system will need to allow fish keepers to input the values, store them and updated them to reduce paper required. No future extensions are required. The system should support both French and English.

All fish keepers will have to have training on how to use the new system. They will be required to monitor the function of the hardware and control it through the software. Hardware installation will be performed at the end of development but initial software data will need to be inputted by the customer from their paper records. The customer environment will be accessible on week nights from 11PM to 6AM with prior notice. All tank and aquarium dimensions will available at all times.

The only alternative solution is to continue to manually feed the fish. This costs less in the short term as the existing staff is trained to feed fish. However as more types of fish are added to the inventory the keepers will have an increased workload and likely require additional keepers and most training. An upfront cost for an automated system will allow the aquarium to maintain the current number of fish keepers and expand the aquarium without increasing their workload.

1.3 Work Packages, Schedule and Budget

1.3.1 Resource Requirements

To develop the system, there are some requirements that need to be met. The developing environment will consist of Dell Inspiron Desktop computers integrated with the Linux OS and Windows 8. The software will be developed in the C programming language on computers running the Linux operating system. Each developer will have their own station to work at. All accounting will be done using Windows computers. Section 1.3.4 describes resource allocations and budget costs.

Section 1.3.3 describes the staffing positions and its quantity. The project manager will be the lead for the project. The team will consist of software developers, testers and technicians. The software will be developed by the Software Developer team, the Testers will perform Quality Assurance and Technicians will handle hardware related issues and maintenance. Project quality and testers team will be lead by the Quality Assurance Manager. Customer Service/Tech Support will be responsible for handling customer related queries.

1.3.2 Schedules

Phase	Description	Planned Date
User Requirements	Approval of User Requirements Document by Project Manager and Analyst	December 16
Software Development	Approval of Software Development by Project Manager and Analyst	January 2
Software Requirements & Hardware Requirements	Approval of Software Requirements Document	January 15
Architectural Design	Approval of Architectural Design Document	February 15
Preliminary Design	Approval of Preliminary Design prototype	March 1
Detailed Design	Approval of Software and Hardware Detailed Desgin	May 12
System Integration & Testing	Integration of system and testing	June 15
Deployment	Deployment	June 30

1.3.3 Staffing and Organization

Position	Quantity
Analysts	1
Project Manager	1
Software Developer	4
Quality Assurance Manager	1
Software Tester	3
Human Resource	1
Customer Service/Tech Support	2

Breakdown of each employee’s responsibilities are shown in Section 1.4.7

1.3.4 Budget and Resource Allocation

Resource allocation will be spent on the total man hours to complete the project at specific phases. Estimated man hours are based on the total hours combined for Software Developers and Testers.

Phase	Estimate (man hours)
User Requirements	160
Software Requirements	200
Architectural Design	270
Detailed Design	550
Margin	250
Total	1430

Other additions into the budget includes workspaces, computers for document processing and software development, a server, chairs, printers and a whiteboard.

Hardware Resources	Quantity	Cost ea.
Dell Inspiron 660S Desktop computer	10	$379.99
Cube Solutions Full Height Jr. Executive Cubicle, Single Cubicle	10	$1,849.99
Work Center 6605 printer	2	$749.00
PowerEdge GT320 Tower Server	1	$1,819.00
Office Star Mesh Multifunction Task Chair, Black	10	$169.00
TOTAL		$40,697.80

Software Requirements
For the development of the feeder system, all code is to be developed using the C Language in the Linux OS environment. Items not related to the development of software, such as any accounting/financing services, should will be done in Windows 8 OS using Microsoft Office Tools.

Minimum system requirements to develop the system:

- 2GB RAM
- 500GB HardDrive Space
- 2 MBit/s network

1.3.5 Cost/Benefit Analysis

To reduce costs, the project is required to use automated scripts and reusable code.
Section 1.3.4 provides the cost of total required man hours as well as additional resources. Additional risks that the project and/or system may encounter are listed in Section 1.3.6. Costs for the project can include replacing and/or fixing malfunctioning hardware. The project will not be using existing code, as there are no code from previous projects that were reuseable.

Benefits to the customer includes automation of feeding system without the need for extra effort towards employees. The system can also keep track and maintain the right dosage of food for each tank.

1.3.6 Risk Analysis and Management

Risk analysis should include the identification of potential risks, its damages and the means to reduce the impact of such risks imposed to the system. The risks should include risks associated with the system as well as risks associated with the management.

1.3.6.1 System Risks

The system may encounter certain risks during its lifetime. The table below identifies each risk towards the system, as well as the probability and severity of the risk.

Identified Risk	Probability	Severity	Estimated Risk	Acceptability
Power Failure	Medium	Medium	Medium	**ALARP**
Hardware Failure	Medium	Medium	Medium	**ALARP**
Overdose of food	Medium	High	High	**Intolerable**
Under dose of Food	Medium	High	High	**Intolerable**
Electrical	Medium	High	High	**Intolerable**

In addition to the identified risks above, each risk has a proposed solution. The table below shows each each associated with its solution. It is up to the project manager and analyst to decide which solution is to be implemented.

Identified Risk	Proposed Solution
Power Failure	- Implement power generator. Implement built-in battery
Hardware Failure	- Remove faulty hardware from system. - Replace faulty hardware with spare - Send faulty hardware to technician
Overdose of food	- Implement filtration system - Manually clean tank - If hardware issue, remove faulty hardware from system. Replace faulty hardware with spare. Send faulty hardware to technician - If software issue, suspend entire feeding system. Report problem to developers. Implement changes. Resume system

Underdose of food	- If hardware issue, remove faulty hardware from system. Replace faulty hardware with spare. Send faulty hardware to technician - If software issue, report problem to developers. Implement changes. Resume system
Electrical	- Implement fuses into hardware.

1.3.6.2 Management Risks

In addition to the risks associated with the system, there are also risks that can occur within management.

Missed Days. Risks associated with management includes missed days such as illnesses. Manager should inform the CEO and the team, if possible, before the absence. If the manager cannot inform the CEO and team, the team leader should be able to take over for a short period of time.

1.3.6.3 Resource Risks

Risks towards resources may have an impact on the project. Each risk is identified along with its proposed solution.

Damaged computers: Computers should be backed up regularly. If a computer is severely damaged, the computer is to be replaced. All files recoverable are to be transferred to the new computer.

Project Time Shortage: Developers may experience project time low. The project manager is responsible for adjusting the project plan. Times could be early or late completion. For late completion, lowest priority requirement is to be done last or to be not included in system design.

System Design Errors: Developers may experience faults or errors in the system program design. It is the responsibility of the Quality Assurance Manager and its team to properly find and document all faults and errors in the system and to notify development team of errors. Errors should be corrected as soon as possible to avoid time shortage.

1.3.7 Deliverables

Phase	Deliverables	To whom	Format
User Requirements	User Requirements Document Software Project Management Plan Software Quality Assurance Plan	Senior Manager/customer Senior Manager Senior Manager	Paper and electronic
Software	Software Requirements Document	Senior Manager/customer	Paper

Requirements	Software Test Plan Hardware Test Plan	Senior Manager/customer Senior Manager/customer	and electronic
Architectural Design	Architectural Design Document Integration Test Plan	Senior Manager/customer Senior Manager/customer	Paper and electronic
Detailed Design	Detailed Design Document Unit Test Plan Software User Manual Source code	Senior Manager/customer Senior Manager/customer Senior Manager/customer Senior Manager/customer	Paper and electronic

1.3.8 Software Methods, Techniques and Tools Needed

System Design should include flow charts, flow diagram, case and pseudocode.

Test data will be generated using closed box and open box testing. For black-box testing, all possible input will be tested into the system with the expected results according to user requirements. Tests should include statement testing, branch testing and path testing.

Automated tools for testing will be used for static and dynamic analysis. Such tools that should be used for static analyzing can be code analyzers, data analyzers and sequence checkers. Dynamic analysis can be used for concurrency. Tests should ensure correctness of desired results with user requirements and system specifications.

1.4 Managerial Procedures

1.4.1 Organizational Structure

This project will follow an organizational structure based on the following roles stated in the table below.

Position	Quantity
Senior Management	1
Analysts	1
Project Manager	1
Software Developer	4
Quality Assurance Manager	1

Software Tester	3
Human Resource	1

Senior Management is employed by Aquarium while all other members are part of Software Consulting firm. It is the responsibility to maintain and set deliverables by Project Manager and communicate to Senior Management. With that said, if Project Manager is on absence of leave it is the duty of an Analyst to report to Senior Management on the project.

1.4.2 Process Model

The model used for making the Fish Feeding System is the System Lifecycle (DoD Standard 2167a). This model puts emphasis in developing software and hardware at the same time and then integrating both phases as a whole system where rigorous testing is performed on the system. This is the ideal model for this project because the System needs a machine and software to perform tasks in feeding fish.

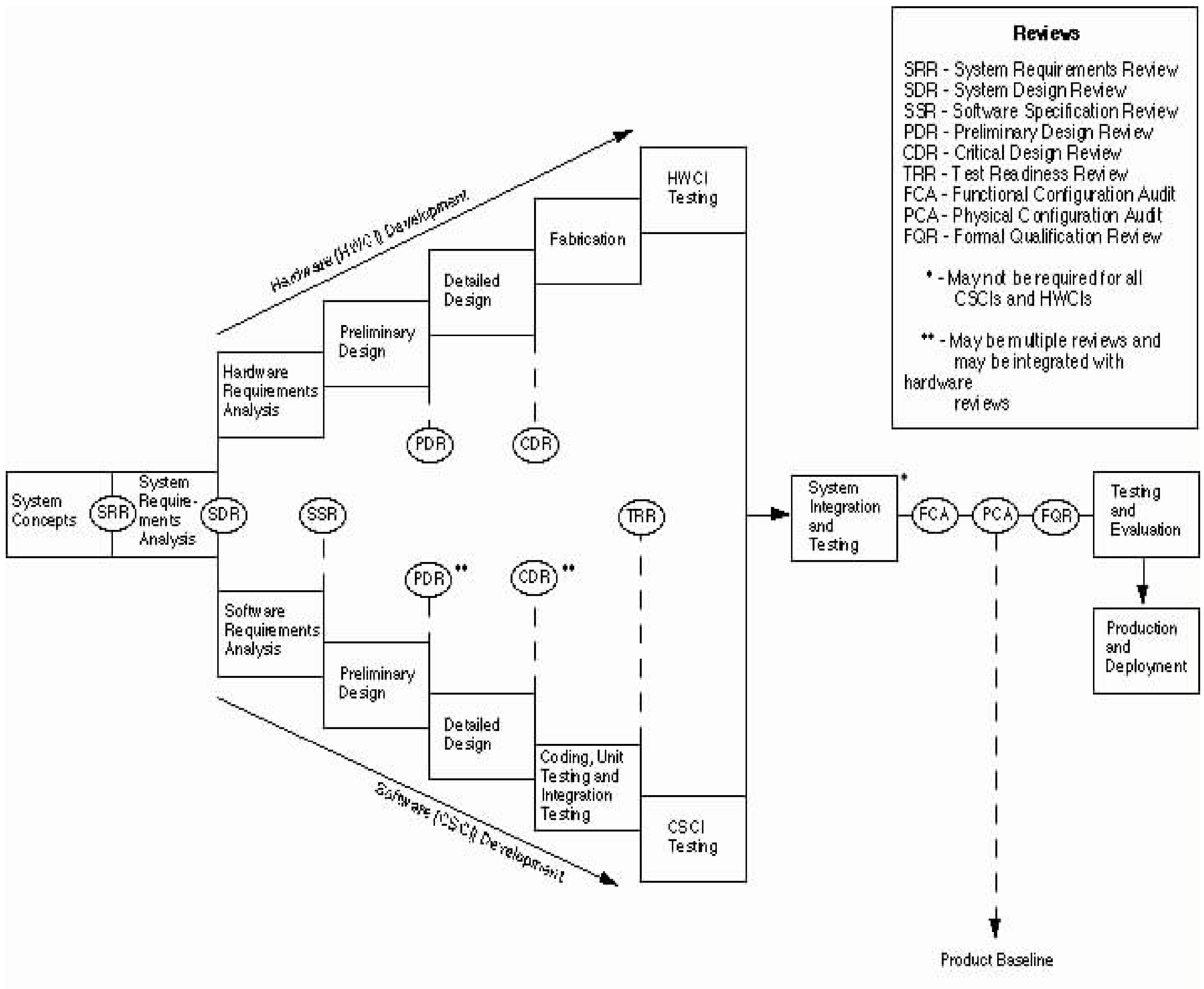

Reviews
SRR - System Requirements Review
SDR - System Design Review
SSR - Software Specification Review
PDR - Preliminary Design Review
CDR - Critical Design Review
TRR - Test Readiness Review
FCA - Functional Configuration Audit
PCA - Physical Configuration Audit
FQR - Formal Qualification Review
* - May not be required for all CSCIs and HWCIs
** - May be multiple reviews and may be integrated with hardware reviews
Hardware (HWCI) Development
HWCI Testing
Fabrication
Detailed Design
Preliminary Design
Hardware Requirements Analysis
PDR
CDR
System Concepts
SRR
System Require-ments Analysis
SDR
SSR
TRR
PDR **
CDR **
System Integration and Testing *
FCA
PCA
FQR
Testing and Evaluation
Production and Deployment
Software Requirements Analysis
Preliminary Design
Detailed Design
Coding, Unit Testing and Integration Testing
CSCI Testing
Software (CSCI) Development
Product Baseline

Figure 1-8 DoD-Std-2167A Software Development Lifecycle Example

1.4.3 Methodologies and Notations

The methodologies used for this project come from DoD-STD-2167A which contains a System's Development Cycle. It is in good practice to follow common conventions to make readability and maintainability of code easier. A Higher order language will be considered and its naming conventions will be adhered too. Other good practices include a modular approach of the system and code re-use. In addition to that, our database will go through normalization to reduce redundancy and contain E-R diagrams explaining the system(symbolic notation).

1.4.4 Boundaries & Interfaces

The fish feeding system team interacts with the following people throughout the project:

- senior manager: project manager maintains communication with senior manager on a weekly basis
- CEO: project manager and senior manager maintain communication with CEO bi-weekly
- Aquarium Staff: project manager may meet and research tasks and efficiency issues and knowledge from Staff

1.4.5 Accountability Monitoring

It is the responsibility of the senior management and project manager to maintain accurate accounts of project costs and deliverables. With that said, bi-weekly meetings with the CEO are done for their work to be monitored and reviewed.

1.4.6 Product Control

Project manager is responsible for keeping log books of expenditures and financial reporting to higher levels. Keeping target budget in mind; project manager must control all factors throughout the project to ensure spending has reached a reasonable amount.

1.4.7 Responsibilities

The following is a brief summary of roles that are fulfilled by employees of Software Consulting Firm:

Project Manager

- Produce and maintain a project plan
- Motivating team
- Defining work packages and goals
- Maintaining a time budget
- Feedback to senior management by reporting progress to senior management

Quality Assurance Manager

- Guarantee that product will be delivered to quality standards
- Checking all project documents
- Monitoring and reviewing all testing activities

Analyst
- Writing formal business reports
- Communicating to clients
- Communicate to and instruct software developer and software tester
- Update issues and progress to project manager

Software Developer
- Executing plans made by project manager
- Develop software
- Document software
- Adhere to best programming practices
- Update issues and progress to project manager

Software Tester
- Executing plans made by project manager and quality assurance manager
- Write unit tests
- Write reports
- Give feedback to software developer
- Update issues and progress to project manager and quality assurance manager

Human Resources
- Setting up payroll and contracts
- Account entry of expenditures and credit
- Setting up employee ID and building access

1.5 List of Definitions

ALARP	As Low As Reasonably Possible

2. Software Requirements Specifications

2.1 Introduction: Overview

Fish are currently fed manually by aquarium staff. This leads to two different issues. The first issue is that staff have to manually measure food for the fish. Some fish have very strict diets and mismeasured food will harm them which can be costly or fatal to the fish. The second issue is that staff have to get the information about each fish from paper files which is very time consuming.

2.2 Software Functions

2.2.1 Process Descriptions - Product Functions

2.2.1.1 ManageTanks

Select Tanks: Select one or more tanks that needs feeding.
Priority:High

Set Food Type: Set the required food type for a tank.
Priority:High

Set Food Count: Set the required food quantity needed for a tan.
Priority:High

2.2.1.2 ManageFeeder

Select Feeder: The software should keep track of each feeder.
Priority:High

Select Food: The software should keep track of each tank with its associated food type.
Priority:High

Move to tank: The feeder should move to a specified fish tank.
Priority:High

Dispense Food: The feeder hardware mechanism selects the right amount of food for tank and dispenses the food.
Priority: High

Log Feeding Time: The software must record the times at which the tanks were fed by the feeder.
Priority: High

Food Count Status: Three light system. Green - Full food capacity, Orange - medium food capacity, Red - Low food capacity
Priority: Medium

Halt: The feeder should suspend hardware functions if any fault/error has occured. Flashing red - the hardware has halted due to error/faults in the system.
Priority: Medium

2.2.1.3 ManageInventory

View Food Quantity: The system must be able to keep track of the quantity of a food type in the inventory.
Priority: Medium

View Food Type: The system must be able to keep track of all the food types in the inventory.
Priority: Medium

Update Food Quantity: Every Time a food type has been added or removed, the system must be able to update the food quantity of a food type in the inventory.
Priority: Medium

2.2.1.4 ManageMaintenace

View Maintenance Time: Keep track of all maintenance times made to the system.
Priority: Medium

View Maintenance Log: Keep track of a detailed description of the maintenance done to the system.
Priority: Medium

2.2.1.5 MaintainUsers

Employee: Keep track of the employees who has access to the feeder system.
Priority: Medium

Usage Time: Keep track of the time when the Employee used the system
Priority: Medium

2.2.2 Data Descriptions

Data	Type	Description
TankID	Integer	Tank Identification Number
FeederID	Integer	Feeder Identification Number
FoodType	String	Type of food
FoodCount	Integer	Count of food type
FeedTime	String	Time stamp for when a tank is fed.
MaintenanceID	Integer	Maintenance Identification Number
MaintenanceTime	String	Time stamp for when the maintenance was performed to the system
DetailedDescription	String	A detailed description of the maintenance
EmployeeID	Integer	Employee Identification Number
EmployeeName	String	Employee’s name
UsageTime	String	Time stamp for when an employee interacts with the system.
ErrorCode	Integer	Error code number
ErrorDesc	String	Error code description

2.2.2.1 Feeder Data Description

Data	Type
TankID	Integer
FeederID	Integer

Data	Type
FeederID	Integer
FoodType	Integer
FoodCount	Integer
FeedTime	String

2.2.2.2 Tank Data Description

Data	Type
TankID	Integer
FoodType	String
FoodCount	Integer

2.2.2.3 Maintenance Description

Data	Type
MaintenanceID	Integer
MaintenancTime	String
DetailedDescription	String

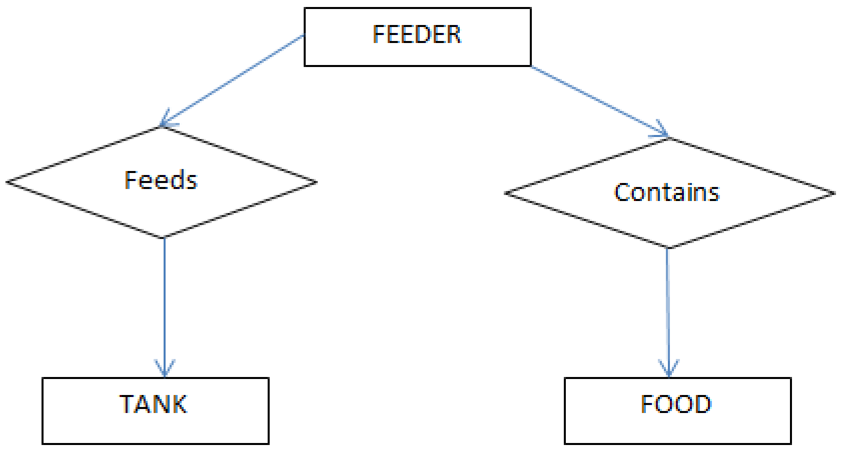
2.2.2.3 Employee Description

Data	Type
EmployeeID	Integer

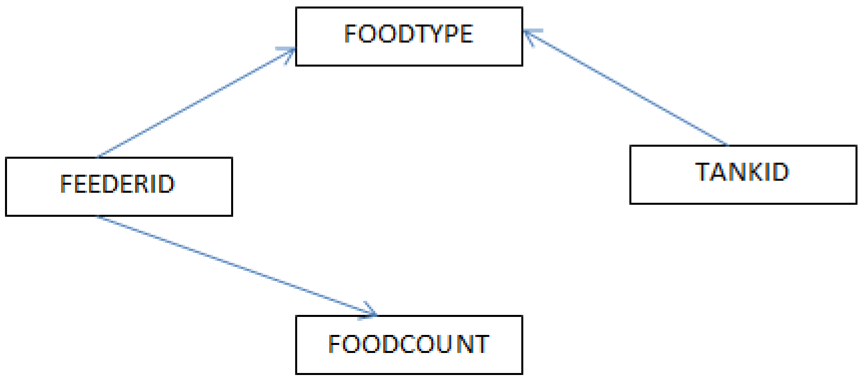
EmployeeName	String
UsageTime	String

2.2.3 Data Relationships

2.2.3.1 ER-Diagram



2.2.3.2 Data Structure Diagram



2.3 Constraints

2.3.1 System Interfaces (Hardware, Operating System, Network)

Dell inspiron 660 is the choice of hardware that will be used in developing software for the feeder system. Any issues with hardware needs to be consulted with Dell technical support resulting in time lost on project. If it is found that a hardware components needs to be replaced than it has cost money and a loss of a developers work.

The Feeder System has many hardware components that need to work with each other. A major constraint is these parts are ordered from others and any issues with them requires expertise that is not in scope of project team. This could halt progress altogether until just this issue is resolved.

All work done by project manager and software developers are uploaded onto central server. If the server has failed than all work has been lost even with regular back-ups it may not contain the current work done.

Troubleshooting in Linux operating system and Windows 8 environments can take some research and is a minor constraint since it takes time away from the project.

2.3.2 Performance Requirements: Timing and Processing Rates

If it takes the system time to process entered data than this can cause problems when the fish in the aquarium get feed. This results in the whole system unable to perform its intended function

2.3.3 Memory Constraints

C is the programming language of choice in developing the Feeder System. In C, pointers point to memory locations where having the wrong declarations and usage of memory can cause segmentation faults and system to crash.

2.3.4 External Interfaces (Screens, Data, Command, and Message Formats)

A screen is connected to the Feeder System for Aquarium Staff to view and edit data. A major constraints can happen if the screen is not properly held in place with the whole system. Since the Feeder Tank moves back and forth on a railing it creates some force and tension. Ordering just one screen could cause a delay in the project if it became damaged in testing its stability during use.

2.3.5 Code and Data Size, Processing Volume

Efficiency of the software is highly dependant on the data structures chosen during development. If over time data size has increased due to growth of the Aquarium business the software could suffer by being very slow. It can result in errors or total failure to the system making it unusable to the user.

2.3.6 Portability Aspects, Accuracy, and Security

Delivering the system will require special machinery to install into an aquarium. Issues while placing the system in place can result in damage to the system leading to a major failure in the project.

Any operations done on software side of the Feeder System needs to checked for accuracy. Detecting such errors can be difficult and time consuming process. If it takes 4 months to detect an inconsistency in the data then we have had 4 months of inaccurate data that can cause inefficiency in ordering supplies.

2.4 Reliability: Assumptions and Dependencies

This system is a soft real-time system. Deadlines for events do exist but they are not extremely strict. A missed deadline will not cause a total system failure as long as the event still occurs within a timeframe of approximately one hour. However continuous missed deadlines can become a serious problem.

2.4.1 Exception Handling

The system will need to be able to handle some exceptions to operate correctly. Hardware failures and software faults need to be reported on-screen so the appropriate action can be taken by the aquarium staff. Data faults will need to be reported on data entry and offer the option of correction so the user is able to edit the incorrect data without restarting the entry entirely.

If a feeding deadline is missed significantly it can be combined into the next deadline twice before manual intervention will be required. Therefore for the first two deadline exceptions, the feeding action can be joined with the following action if the actions are too close in time to allow both feeding actions to occur on their own. If a deadline is frequently missed, user interaction will be required to manually feed the fish and adjust the feeding schedule so the feeding system can handle the quantity of fish to be fed at each deadline.

2.4.2 Hardware Failures and Responses

There are a few main ways the hardware can fail. The worst failure would be if the suspension broke resulting in the feeding car falling into the water. This will likely cause serious damage to the aquatic life and the tanks. Recovery would be extremely expensive.

Other hardware failures are not as major. These failures will result in a system that is unable to perform its action but will not cause any environmental damage. Some examples are food car motor failure, dispenser door failure and screen failure. As there is a current manual system is in-place for feeding the fish, it can serve as a fall-back in the event of a total system failure.

2.4.3 Software Faults and Responses

The most likely software faults are related to the timing of feeding events. The highest chance of a mis-timing happening will be based around daylight savings time changes. To prevent this we will make our system aware of time changes and make sure that they do not trigger duplicate events or skip events.

2.4.4 Data Faults and Responses

Data will all be inputted manually by the staff. As new fish are added and old fish are removed, the staff will be required to update the database through the interface provided. All data will be verified before being saved into the database. A feeding example will also be shown upon data entry to confirm that it is correct.

2.5 Software Life Cycle

2.5.1 Description and Justification of the Life Cycle Used

DoD Standard 2167a is document from Department of Defense containing a Systems Development Cycle; this is the life cycle of choice as both hardware and software components of system are designed separately, but remain dependant on each other(Please see Section 3.3 Alternatives on why this Cycle fits our System). This allows both the hardware and software to be developed at the same time and be tested together to ensure compatibility as opposed to developing them separately, where unknown issues and complexities could be created that affect the other component without our knowledge.

2.5.2 Standards and Methodologies for the Development

The standards we will use are included in the definition of DoD-STD-2167A. Following are some examples.

2.5.2.1 Higher Order Language

All code shall be written in the higher order language specified in the System Specifications.

2.5.2.2 Control Constructs

Code shall be written using the constructs SEQUENCE, IF-THEN-ELSE, DO-WHILE, DO-UNTIL, and CASE.

2.5.2.3 Modularity

Source code for each unit must not exceed 100 executable statements. Also, units shall follow the following characteristics:

A. Local variables within separate units must not share the same storage locations.
B. Each unit must perform a single function.
C. Modifying a unit’s code during execution is prohibited.

D. Each unit must be uniquely named.
E. All units must follow a standard format made up of prologue, declarative statements, and executable statements or comments.
F. Aside from error exits, each unit must only have one entry and one exit point.
G. Coding style conventions must be consistent between units.

2.5.2.4 Symbolic Parameters

Symbolic parameters must be used, in lieu of specific numeric values, to represent constants, relative location within a table, and size of data structure.

2.5.2.5 Naming

Naming conventions must be uniform throughout the computer software configuration item and must employ meaningful names which clearly identify the constant, variable, function performed, and any other objects used in the computer software configuration item, to the reader. Language keywords must not be used as identifies

2.5.3 Predicted Modifications and Maintenance

Depending on the shape and size of the tank the system is to be installed on, the rail system used may have to be custom built to accomodate.

Maintenance includes ensuring the electrical equipment does not get wet, and the wheels of the dispenser remain well lubricated.

2.5.4 Acceptance Procedures

Upon acceptance of our system, an installation crew will arrive to install the system to the desired fish tank, and fully instruct the staff on how to operate the system so that they may program it for their needs.

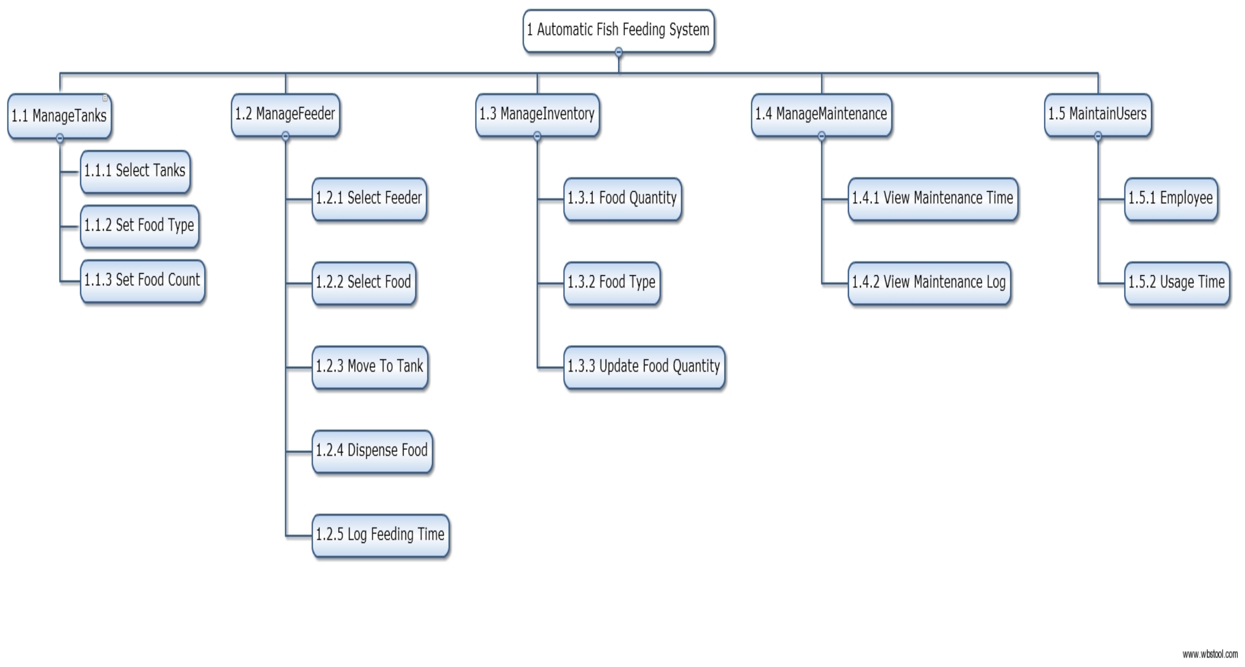
2.6 References

Documents

DoD-STD-2617A
http://www.product-lifecycle-management.com/download/DOD-STD-2167A.pdf

3. Design Specifications

3.1 Architectural Design



3.2 Detailed Designs

3.2.1 Data Dictionary

Refer to Section 2.2.2

3.2.2 ER-Diagrams

Refer to Section 2.2.3

3.3 Alternatives

The fish feeder system is a critical system and therefore it's development has to be considered in a manner that allows rigorous testing during development. Other Critical systems used online that was considered was developing Medical software in various devices(the link is attached here-http://eprints.dkit.ie/237/1/IntegratingAgileMedicalMMHFMCVC.pdf)
The critical system above uses a hybrid of V-model and agile practices to introduce 2 extra stages during development that was chosen to satisfy laws made for medical software. Therefore the system will not undergo any changes with the design process and will continue using the DoD Standard 2167a document for building an effective critical system. The DoD documentation best fits the design process as this system uses hardware and software. The hardware and software design can run in parallel and therefore decrease the total time costs for

implementing the system. Once the testing of each, hardware and software, is complete, it can finally be integrated. Once the integration process has been completed, final testing and evaluation can be performed on the system as a whole.

4. Detailed Design: The Unit Folder

4.1 Module Design

Refer to Module Designs at the end of the report.

4.2 Unit Status

4.2.1 Sign-Off Sheet

ASSIGNMENT GROUP: ________________________________

TITLE:

LOCATION:	

REQUIRED COMPLETION DATE: ________________________________

EMPLOYEE NAME	COMPLETION DATE	INIT	COMMENTS

4.2.2 Planning and Actual Scheduling

4.2.2.1 Phase 1

POSITION	**TASK**	**DURATION** (hours)
Senior Manager	Manage	160
Analyst	Analyse User Requirements	160

4.2.2.2 Phase 2

POSITION	**TASK**	**DURATION** (hours)
Senior Manager	Manage	100
Analyst	Software Requirements	100
Software Developer #1	Software Requirements	100

4.2.2.3 Phase 3

POSITION	**TASK**	**DURATION** (hours)
Senior Manager	Manage	270
Analyst	Architectural Design	90
	Detailed Design	180
Software Developer #1	Architectural Design	90
	Detailed Design	180
Software Developer #2	Architectural Design	90
	Detailed Design	180

4.2.2.4 Phase 4

POSITION	**TASK**	**DURATION** (hours)
Senior Manager	Manage	50
Software Developer #1	Software Development	50
Software Developer #2	Software Development	50
Software Developer #3	Software Development	50
Software Developer #4	Software Development	50
Quality Assurance Manager	Assure Quality	50

4.2.2.5 Phase 5

POSITION	**TASK**	**DURATION** (hours)
Senior Manager	Manage	50
Software Developer #1	Software Development	50
Software Developer #2	Software Development	50
Quality Assurance Manager	Assure Quality	50
Software Tester #1	Testing	50
Software Tester #2	Testing	50
Software Tester #3	Testing	50

4.2.3 Acquisition of Reusable Components and Tools

We will be designing this system from scratch, not introducing any existing components or tools. Our development will allow up to recreate the system for other fish tanks easily, only requiring to acquire a custom set of rails