Software Requirement Specification Document
Nursery System

David Adel, Youssef Alaa Eldin, Mahmoud Ahmed, John Monir

March 11, 2018

1 Introduction

1.1 Purpose of this document

In this document we're stating the main (non)functional requirements as well as targeting our main audience; Prof. Ayman Ezzat and nursery owners.

1.2 Scope of this document

As per this document, functionalities had been elicited by our Customer Mrs. Anahid, we've agreed on developing the web application upon her requirements stating a software system that provides support for the development, repair, and enhancement of software, and for the management and control of these activities. Our system requires a personal computer having any internet-browser for online operations or a software to launch the local host database. This application will be held by August, 2018 and will costs zero pounds; since its done for academic purpose and the developers who worked on this projects David Adel, Youssef Alaa Eldin, Mahmoud Ahmed, John Monir.

1.3 Overview

This Nursery web application will get rid of the excess paper work usage as it will organize and make it much easier to retrieve reports of each child or worker in the constitution.

1.4 Business Context

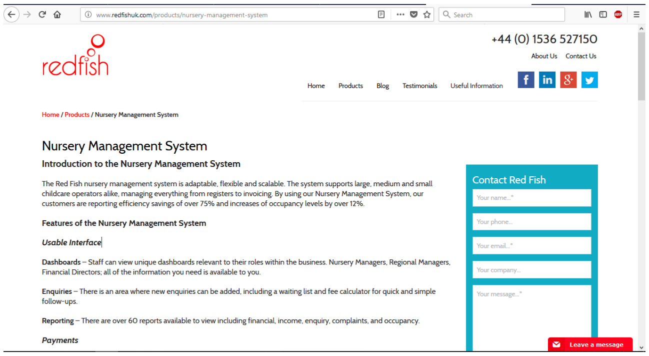
As per the Nursery web application, mainly will harshly decrease the amount of time preparing the required data. It will eliminate the existent of any paper-work as everything will be automatically stored on the database. It will allow the users to retrieve any kind of reports they require with the minimum effort and time.

2 General Description

2.1 Product Functions

This software contains different kind of modules; User Manipulation Module, Attendance System, Check list of child's everyday activities, User Application Manipulation Module, Event Creation Module.

2.2 Similar System Information



Nursery Management System

Introduction to the Nursery Management System

The Red Fish nursery management system is adaptable, flexible and scalable. The system supports large, medium and small childcare operators alike, managing everything from registers to invoicing. By using our Nursery Management System, our customers are reporting efficiency savings of over 75% and increases of occupancy levels by over 12%.

Features of the Nursery Management System

Usable Interface

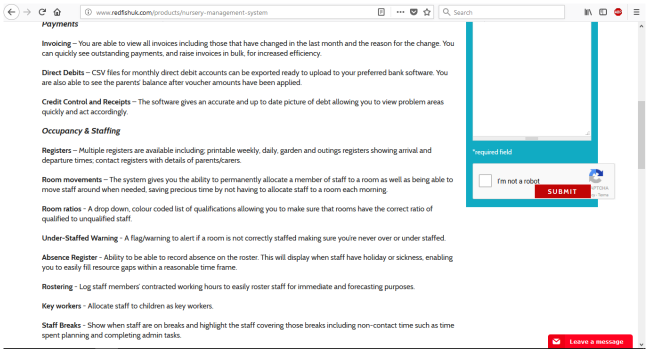
Dashboards – Staff can view unique dashboards relevant to their roles within the business. Nursery Managers, Regional Managers, Financial Directors; all of the information you need is available to you.

Enquiries – There is an area where new enquiries can be added, including a waiting list and fee calculator for quick and simple follow-ups.

Reporting – There are over 60 reports available to view including financial, income, enquiry, complaints, and occupancy.

Payments

Figure 1: Similar System showing included functionalities.



Payments

Invoicing – You are able to view all invoices including those that have changed in the last month and the reason for the change. You can quickly see outstanding payments, and raise invoices in bulk, for increased efficiency.

Direct Debits – CSV files for monthly direct debit accounts can be exported ready to upload to your preferred bank software. You are also able to see the parents' balance after voucher amounts have been applied.

Credit Control and Receipts – The software gives an accurate and up to date picture of debt allowing you to view problem areas quickly and act accordingly.

Occupancy & Staffing

Registers – Multiple registers are available including; printable weekly, daily, garden and outings registers showing arrival and departure times; contact registers with details of parents/carers.

Room movements – The system gives you the ability to permanently allocate a member of staff to a room as well as being able to move staff around when needed, saving precious time by not having to allocate staff to a room each morning.

Room ratios - A drop down, colour coded list of qualifications allowing you to make sure that rooms have the correct ratio of qualified to unqualified staff.

Under-Staffed Warning - A flag/warning to alert if a room is not correctly staffed making sure you're never over or under staffed.

Absence Register - Ability to be able to record absence on the roster. This will display when staff have holiday or sickness, enabling you to easily fill resource gaps within a reasonable time frame.

Rostering - Log staff members' contracted working hours to easily roster staff for immediate and forecasting purposes.

Key workers - Allocate staff to children as key workers.

Staff Breaks - Show when staff are on breaks and highlight the staff covering those breaks including non-contact time such as time spent planning and completing admin tasks.

Figure 2: Similar System showing included functionalities cont.

2.3 User Characteristics

This system will be used by the Nursery's Management Department; The user should have a 4 hour training to be able to deal smoothly with the system as well as having basic computer knowledge and English commands.

2.4 User Problem Statement

The main essential problem is getting rid of excess paper work, and lost time.

2.5 User Objectives

An organized flow of data, user-friendly system, a well handled time management system and minimizing the complexity of the work flow.

2.6 General Constraints

The constitution should have a modern computer to process the system, as well as a fingerprint scanner for the workers attendance.

3 Functional Requirements

Code	F.01
Name	login
Type	Functional
Critically	High
Input	Username, Password
Output	Redirect to specific
Description	The user input his/her username and password to check them in the database
Priority	10/10
Expected Risk	Database size, bugs, wrong email or password
Preconditions	Open website
Post-Conditions	Fill session with user id
Dependencies	none

Code	F.02
Name	hashing
Type	Functional
Critically	High
Input	Password
Output	Hashed Password
Description	The password will be hashed and inserted in the database
Priority	10/10
Expected Risk	Hashing can be predicted
Preconditions	none
Post-Conditions	none
Dependencies	Fid.01

Code	F.03
Name	showNav
Type	Functional
Critically	High
Input	none
Output	show pages available for this role
Description	When the user logs in, his role will be retrieved from the database and the pages assigned to his role will be showed
Priority	10/10
Expected Risk	Database size
Preconditions	user must be logged in
Post-Conditions	pages will be shown
Dependencies	Fid. 01, Fid. 02

Code	F.04
Name	addAnotherUser
Type	Functional
Critically	High
Input	User
Output	none
Description	The admin will enter the user info and then it will be inserted in the database
Priority	10/10
Expected Risk	the data entered may be wrong
Preconditions	user must be logged in and of type admin
Post-Conditions	a new user will be created
Dependencies	Fid. 01, Fid. 02, Fid.03

Code	F.05
Name	editAnotherUser
Type	Functional
Critically	High
Input	User
Output	none
Description	The admin changes another user's information thought the parameter given
Priority	10/10
Expected Risk	Incorrect input might be entered
Preconditions	Admin must be logged in
Post-Conditions	User will be updated
Dependencies	Fid. 01, Fid. 02 Fid. 03

Code	F.06
Name	viewAnotherUserInfo
Type	Functional
Critically	High
Input	User
Output	A page with the User input data
Description	After selecting a certain user, will be directed to a page with their information
Priority	10/10
Expected Risk	none
Preconditions	A user of type admin has to be logged in
Post-Conditions	Will be directed to the specified user information page
Dependencies	Fid. 01, Fid. 02 Fid. 03

Code	F.07
Name	add
Type	Functional
Critically	High
Input	none
Output	none
Description	An object is added depending on the class implementing the function
Priority	10/10
Expected Risk	Incorrect data might be entered
Preconditions	A User must be logged in and of type Admin
Post-Conditions	A new object of the implimenting class will be inserted in a specific table
Dependencies	Fid. 01, Fid. 02 Fid. 03

Code	F.08
Name	edit
Type	Functional
Critically	High
Input	none
Output Description	none An object is updated depending on the class implementing the function
Priority	10/10
Expected Risk	Incorrect data might be entered
Preconditions	A User must be logged in and of type Admin
Post-Conditions	User will be updated
Dependencies	Fid. 01, Fid. 02 Fid. 03

Code	F.09
Name	delete
Type	Functional
Critically	High
Input	none
Output	none
Description	An object will be moved from the current table to a specified log table depending on the class implementing the function
Priority	10/10
Expected Risk	none
Preconditions	A User must be logged in and of type Admin
Post-Conditions	User will be moved to a specified log table, and removed from the current table
Dependencies	Fid. 01, Fid. 02 Fid. 03

Code	F.10
Name	view
Type	Functional
Critically	High
Input	none
Output Description	none An object is viewed in a specific page linked from the database depending on the class implementing the function
Priority	10/10
Expected Risk	Incorrect data might be entered
Preconditions	A User must be logged in and of type Admin
Post-Conditions	User will be viewed in a specific page
Dependencies	Fid. 01, Fid. 02 Fid. 03

Code	F.11
Name	alert
Type	Functional
Critically	High
Input	none
Output	A pop-up message that there are children with absence of 3 consecutive days or more
Description	A box with a message stating that there are children with absence of 3 consecutive days or more that has a redirect button to the page with the information of absent children and a cancel button
Priority	8/10
Expected Risk	Pressing the wrong button unintentionally
Preconditions	A User must be logged in and of type Admin, and at a child has been recorded as 'not attended' in the attendance table
Post-Conditions	If canceled, nothing happens. Else, redirecting to the list of Absent Children page
Dependencies	Fid. 01, Fid. 02, Fid. 03

Code	F.12
Name	threeDaysOfAbsense
Type	Functional
Critically	High
Input	none
Output	A page with a list of children that have been listed in the database with absence of 3 consecutive days or more
Description	A page with a list of children that have been listed in the database with absence of 3 consecutive days or more with their parent mobile phone and an option of removing the record, the record is moved from the current table to a specified log table
Priority	10/10
Expected Risk	Incorrect data entry while moving the record to the log table
Preconditions	A User must be logged in and of type Admin, and at a child has been recorded as 'not attended' in the attendance table
Post-Conditions	Listing Absent Children
Dependencies	Fid. 01, Fid. 02, Fid. 03, Fid. 11

Code	F.13
Name	Encrypt
Type	Functional
Critically	High
Input	String
Output	none
Description	The given string is encrypted with Caesar ciphering algorithm
Priority	10/10
Expected Risk	Incorrect encryption algorithm that could be decrypted
Preconditions	A User must be logged in and of type Admin, and at a child has been recorded as 'not attended' in the attendance table
Post-Conditions	Encrypted string is assigned to a specific attribute and sent to the database
Dependencies	None

Code	F.14
Name	Decrypt
Type	Functional
Critically	High
Input	String
Output	none
Description	The given string is decrypted with Caesar ciphering algorithm
Priority	10/10
Expected Risk	Incorrect decryption algorithm
Preconditions	A User must be logged in and of type Admin, and at a child has been recorded as 'not attended' in the attendance table
Post-Conditions	String is decrypted and assigned to a specific attribute for use.
Dependencies	None

Code	F.15
Name	addUserDailyHours
Type	Functional
Critically	High
Input	Onleave, WorkersHoursSalary
Output	none
Description	Onleave is an object containing attendance information(the attended hours of the user) which contains the User information. Onleave as well contains leaving time of the attended user. WorkersHoursSalary is an object containing information of a specific role stating the price of the basic hour(price per hour), extra hours(price of extra attended hours), deduction hours(price of deducted hours) and normal hours(the amount of hours a role has to fulfill per day). The function extracts the working hours of the user per day along the month and compares it with the ideal hours per month, that is extracted from WorkersHoursSalary multiplied by the number of working days per month. The absolute value of the calculation of the compared results with the minimum requirement hours is stored in an attribute(hours), if the resulted value is negative or zero, a boolean attribute(isExtra) is set to False, Else, will be set to True.
Priority	9/10
Expected Risk	Incorrect extra of data from the sent parameters.
Preconditions	The attendance of the specific user information has to be valid(exist), and WorkersHoursSalary record which contains the ideal record of working hours.
Post-Conditions	Listing User Role attendance sheet
Dependencies	Fid. 01, Fid. 02, Fid. 03, Fid. 13, Fid. 14

Code	F.16
Name	valueCalculation
Type	Functional
Critically	High
Input	ExperienceSalary, SalaryManipulation, UserDailyHours, WorkedHoursSalary
Output	None
Description	ExperienceSalary is an object that has information of a specific user, their money to be paid depends on their experience that is always fixed. SalaryManipulation is an object that has information of a specific user, an extra money paid depending on the admin point of view, it's not fixed. UserDailyHours is an object to extract the information of the total calculated hours compared to the ideal working hours. WorkedHoursSalary is an object that has the price of basic hour, extra hour, and deduction hour. The value to be paid is the calculated if the isExtra (boolean) of the UserDailyHours is true , then the equation is ExperienceSalary value+ SalaryManipulation +(BasicHour (from WorkedHoursSalary)*Number Of Working Days In The Month)+(Hours(is extracted from UserDailyHours)*ExtraHour(price of extra hour extracted from WorkedHoursSalary)) and if the value of isExtra is False then the equation is calculated by this equation: ExperienceSalary value+ SalaryManipulation +(BasicHour (from WorkedHoursSalary)*Number Of Working Days In The Month)+(Hours(is extracted from UserDailyHours)*DeductionHour(price of extra hour extracted from WorkedHoursSalary))
Priority	10/10
Expected Risk	Incorrect calculation
Preconditions	The required tables (UserDailyHours, ExperienceSalary, SalaryManipulation, and WorkersHourSalary) have to be filled with the needed values to fulfill

Code	F.17
Name	changeStatus
Type	Functional
Critically	High
Input	User
Output	none
Description	This function allows the admin to manipulate the status attribute of the given user in order to be able to categorize certain type of status together for different purposes.
Priority	10/10
Expected Risk	If incorrect User was manipulated
Preconditions	A User must be logged in and of type Admin, and an existent user to be manipulated
Post-Conditions	Status is changed of a certain user.
Dependencies	Fid. 01, Fid. 02, Fid. 03

Code	F.18
Name	assignUser
Type	Functional
Critically	High
Input	User, Class, Course
Output	none
Description	Through this function the admin is able to link a user which is in this case a child to a certain course(having a name) which belongs to a certain class(having a name).
Priority	10/10
Expected Risk	Incorrect providing of data
Preconditions	A User must be logged in and of type Admin, and the existent of records of courses, classes, and users.
Post-Conditions	A record is added in the database to fulfill the linkage between each the Child, Class, and the Course.
Dependencies	Fid. 01, Fid. 02, Fid. 03

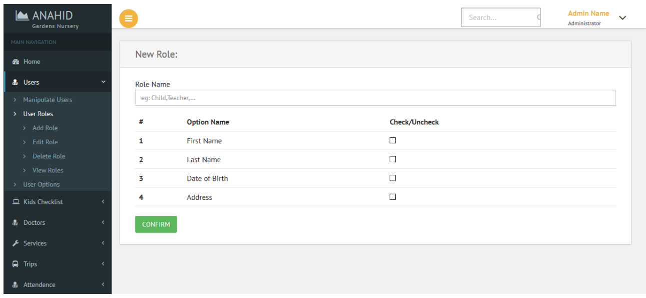
Code	F.19
Name	addDriverCar
Type	Functional
Critically	High
Input	User, Car
Output	none
Description	Through the given user, as well as the car object made, stating the plate number, color, year, model of the vehicle, will be able to assign a certain car to a certain user in order to keep track of each driver details.
Priority	8/10
Expected Risk	Incorrect entry of data
Preconditions	A User must be logged in and of type Admin, and the existent of another user to be able to assign a car to them.
Post-Conditions	Linkage between a user and a car.
Dependencies	None

4 Interface Requirements

This section describes how the software interfaces with other software products or users for input or output. Examples of such interfaces include library routines, token streams, shared memory, data streams, and so forth.

4.1 User Interfaces

4.1.1 GUI



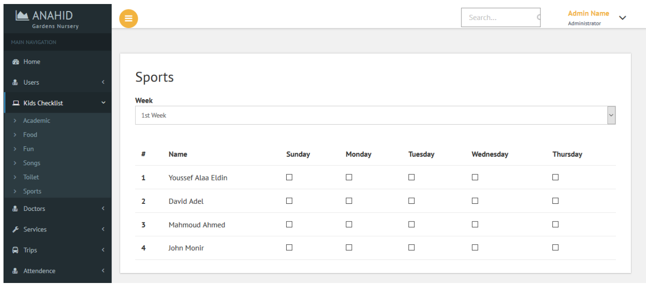
New Role:

Role Name

eg: Child,Teacher,...

#	Option Name	Check/Uncheck
1	First Name	□
2	Last Name	□
3	Date of Birth	□
4	Address	□

Figure 3: Explanation of adding a new Role



Sports

Week

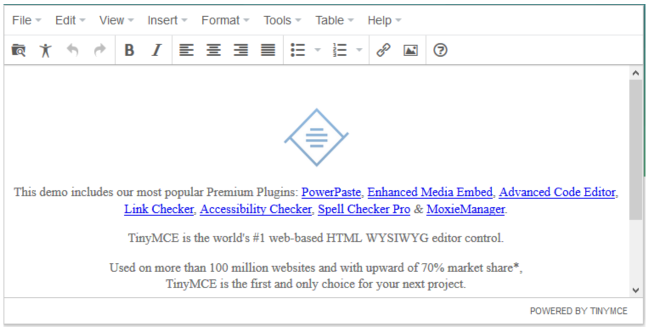
1st Week

#	Name	Sunday	Monday	Tuesday	Wednesday	Thursday
1	Youssef Alaa Eldin	□	□	□	□	□
2	David Adel	□	□	□	□	□
3	Mahmoud Ahmed	□	□	□	□	□
4	John Monir	□	□	□	□	□

Figure 4: Explanation of the checklist functionality

4.1.2 API

TinyMCE



This demo includes our most popular Premium Plugins: PowerPaste, Enhanced Media Embed, Advanced Code Editor, Link Checker, Accessibility Checker, Spell Checker Pro & MoxieManager.

TinyMCE is the world's #1 web-based HTML WYSIWYG editor control.

Used on more than 100 million websites and with upward of 70% market share*,
TinyMCE is the first and only choice for your next project.

Figure 5: a tool used to edit web interfaces easily through few clicks

Github, a tool used make a version control environment

4.1.3 CLI

NA

4.1.4 Diagnostics or ROM

NA

4.2 Hardware Interfaces

4.2.1 Fingerprint Scanner

Hardware used to scan the workers' fingerprint for the attendance system.

4.2.2 Modern Image Scanner

Hardware used to scan the child's documents to enter it in the application.

4.3 Communications Interfaces

NA

4.4 Software Interfaces

NA

5 Performance Requirements

The system requires a minimum of 512MB and 100MB free disk space.

6 Design Constraints

6.1 Standards Compliance

NA

6.2 Hardware Limitations

The system should be connected to a Modern Image Scanner and a fingerprint scanner.

6.3 others as appropriate

NA

7 Other non-functional attributes

7.1 Security

The user's password will be hashed as a kind of security and referred in Fid. 02. As well as encypting the user's sensitive data as referred in Fid.

7.2 Maintainability

The system will be easy to maintain as we are using EAV model in the user details, so it will be easy to add new roles. Referred in Fid. 8, Fid. 9, Fid. 10, Fid. 11

7.3 Portability

The system is a web based application so it is portable and doesn't have OS limitations as it can work on any web browser on any device due its responsiveness.

8 Preliminary Object-Oriented Domain Analysis

8.1 Inheritance Relationships

8.2 Class descriptions

8.2.1 Class name

EAV - Concrete Class

8.2.2 List of Superclasses:

NA

8.2.3 List of Subclasses:

NA

8.2.4 Purpose:

This is the main class in the project through which it will dynamically operate.

8.2.5 Collaborations:

There are few classes required in order to fulfill the correct performance of the EAV which are the ”Entity Class” which is responsible for stating the name of a certain Entity being added(dynamically) to the system, will not co-operate without the existent of the ”Attribute Class” which has the attributes(options) that an entity should have. Along with the last and the most important class which is the ”Values Class” in order to get the desired value of the attribute given of the specified Entity, and this is how everything will dynamically work.

8.2.6 Attributes:

It has the following attributes, ID(1,2,3,etc..), an object from the class Entity(including a name(Application,payment, role and so on) and an attribute(option that would specify this entity such as Firstname, Age, etc..), and an object from the class Values(values specifying the option(Attribute) value such as ”Ahmed”, or ”19” depending on the option data type given via the inserted data in the form)

8.2.7 Operations

There are an amount of operations that might be needed if new requirements are met which will be a new instance of an attribute type(needing to insert a new data type needed in the website), then assigning this new type to an option that you'd like to have or that already exists within the website. And same goes for the Entity along with its name that would be saved in the form of another table considered as a look-up table for the given options(”visa”,”paypal”,etc.) to have reference of it. There is a main side-effect of using this module which would be the term of **complexity**, it has very high complexity in terms of implementation.

8.2.8 Constraints:

There are certain restrictions for the usage of this class and mainly would occur in any other class which would be the repetition of an already existent value which the EAV has already. For example an attribute with the value ”FirstName”, and requiring a new entry with the same name.

9 Operational Scenarios

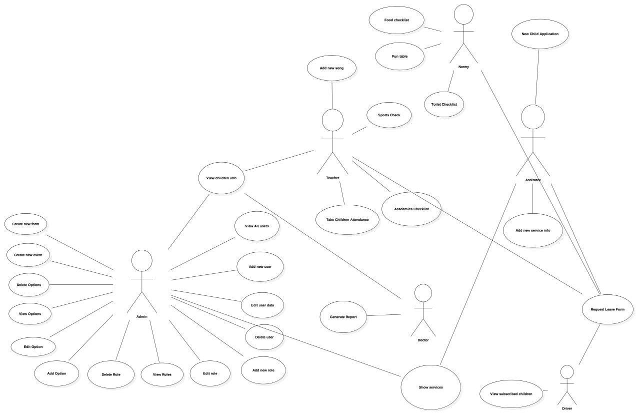


Figure 6: Use Diagram explains how the system works

10 Preliminary Schedule Adjusted

Through the following chart, will state the acquired steps in order to develop the project and deliver it at a certain time.

11 Preliminary Budget Adjusted

This project will cost zero pounds as it is for academic purpose

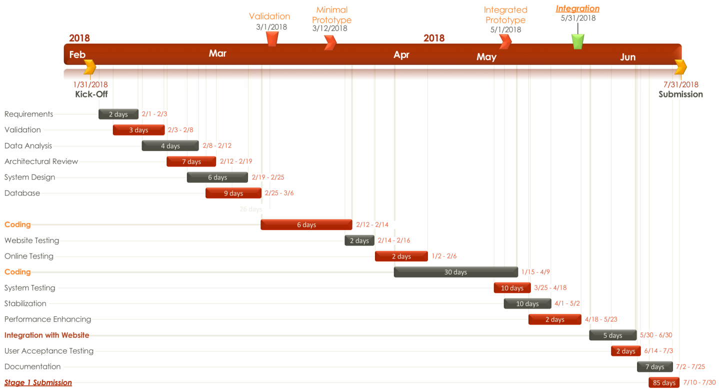


Figure 7: Pert Chart

12 Appendices

12.1 Definitions, Acronyms, Abbreviations

NA

12.2 Collected Material

NA

13 References

http://www.redfishuk.com/products/nursery-management-system
http://www.nurserymanagement.com.au/software-modules/