****

**Bachelors in Information Technology (C & N)**

**Subject Name: Software Engineering (CP2013)**

**Taught By: Dr. Liu Dasheng**

**Project Report on:**

**Zoo Management**

**By**

**Group Name: Roger That**

**Group Member:**

**James Das: 12669110**

**Kevin Leslie: 12780283**

**Nguyen Ha Bao Duy: 12669176**

**Pritam Soni: 12661855**

**Rushabh Padalia: 12766953**

**Abstract:**

* Design and implement appropriate solutions in one or more application domains using software engineering approaches
* Understand and apply software verification, testing, and validation for quality assurance in software development
* Gain practical experience working as part of a team to develop and deliver quality software artefacts
* Our project is divided into two parts that is desktop application and web application.

**Introduction:**

The goal of zoo management project is create a tool for zoo administrators to store information about their zoo on a computerized database system which is divided into two parts: the desktop application and the admin UI (CMS system). The website provides a system, which stores important data of the zoo such as, the number of animals along with their IDs and names, the number of cages with its size and ID, number of animals inside each cage, and location of cages, among other useful information. This information can only be updated or deleted through the admin UI. Whereas on the desktop application we are planning to have push notification which will help the user to receive notifications about the new update, whereas the other features are similar to admin UI and the main feature we will be having is assigning the person for error occurred. This report is a continued as on the report 1 to show the task completed from the week 1 until the week 12.

**Task Description:**

|  |  |  |
| --- | --- | --- |
| Sr no | Name | Task |
| 1 | James Das | Documentation |
| 2 | Kevin Leslie | Developing a desktop application on C# platform |
| 3 | Nguyen Ha Bao Duy | Developing a desktop application on C# platform |
| 4 | Pritam Soni | Server side coding on using PHP |
| 5 | Rushabh Padalia | Our Troubleshooting expert and doing the implementation of the php to the User interface design |

**Top Ten Feature:**

|  |  |
| --- | --- |
| **Sr. No.** | **Feature** |
| **1** | **Update Function in web application and desktop application which is used to update the information in case of modification** |
| **2** | **Delete function in web application and desktop application which is used to delete the information in case of modification** |
| **3** | **Insert function which is used to add new cage or add new animal to a cage** |
| **4** | **Login function to authenticate the zoo management staff** |
| **5** | **Alert function which will be used in case of error occurrence** |
| **6** |  |
| **7** |  |
| **8** |  |
| **9** |  |
| **10** |  |

**Design Pattern Used:**

Design Patterns offer solutions to common revenant issues. Design patterns can be classified as creational, structural or behavioral .Singleton patterns is creational pattern.

**Singleton Design Pattern**

To constrain the number of instances of any object to one at max at any instant of time. The practical implementation of this design pattern can be seen in applications that we use daily. Take the example of Microsoft word .What happens when you open multiple word documents and navigate between them. There is only one instance of the word application active at a time ,you can verify this by checking the active processes .So all the requests for handling the different word documents is handled by a single application instance.

The Bridge pattern is one of these ways - it is similar to the [Adapter pattern](http://giorgiosironi.blogspot.com/2010/01/practical-php-patterns-adapter.html), but its underlying motivation is different: an Adapter makes two unrelated, existing classes work together, when the two participants were not thought to be aware of each other during design, or they must not be coupled as part of different modules. A Bridge is a structure that separates concerns and which is chosen at the design level, before the creation of the participants classes.  
The Abstraction that a Bridge shows to the Client is a class that composes an Implement or interface. Two trees may develop: the abstraction hierarchy and the implementation one. Without this differentiation, the two hierarchies would be merged and the result would be a growing number of classes needed to cover the combinations of Refined Abstractions and Concrete Implementers.  
This pattern is a powerful example of favoring composition over inheritance, one of the motifs of the [GoFhttp://www.assoc-amazon.com/e/ir?t=invistotheeye-20&l=btl&camp=213689&creative=392969&o=1&a=0201633612](http://www.amazon.com/Design-Patterns-Elements-Reusable-Object-Oriented/dp/0201633612?ie=UTF8&tag=invistotheeye-20&link_code=btl&camp=213689&creative=392969) design patterns book. The link between abstraction and implementation consists in and has-a relationship whose target is set at runtime, and that can be extended without the end user of the Abstraction noticing.

**About the admin UI:**

In the admin UI website we have created 2 main features which adhere cage, animal.

**Cage:**

This function is very important because different animals live in different a cage, which includes size, different materials of cages, and so on. Inside this featurewe have created an Add, Delete, and update option.

* **Add:** This option provides a new cage to add with the cage type such as size. This can be done until and unless a new animal comes inside a cage.
* **Delete:** This option removes the existing cage and animal from the website application by selecting options that you want to delete it.
* **Update:** This option updates add and delete function and saves the data in our database.
* **Location:** This option is very useful, users can identify that, in which location all the animal has been placed.
* **Type:** This option provides the type of the cages, because different animal stays in different kind of cages. **For Example** Birds stays in a net, which is made of very thin kind of net cages, whereas lion stays in a solid iron cage. This option will be very useful

**Animal:**

Inside this feature we have created animal type, add to cage, delete, and update options.

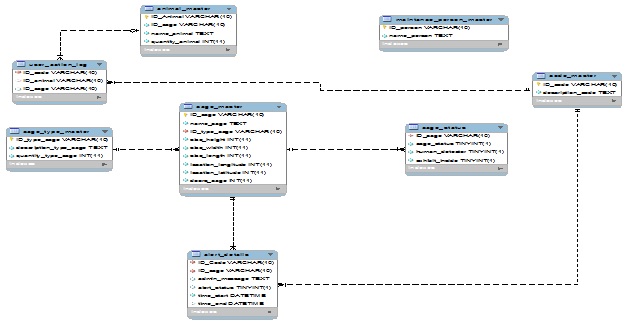
* **Animal type:** In this option it allows a new animal to add, by giving the animal names, age, and quantity. (Description of each animal).
* **Add to cage:** This option allows the animal to add inside the cage. We also have different cage type, because different types of animal stay in different cages.
* **Delete:** This option removes the number of animals from the cages.
* **Update:** this option updates all the data in the database.

In the cage and animal features it is also has the validation function. For example if a user does not select the size of the cage or type, then the error will be pop-up saying that **“Please give the size of the cage”.**

***Please check the prototype, which is attached in this project file***

**Entity Relationship Model:**

*(Please zoom in this chart to get a proper view)*



**Log Book**

**Week 6**

**Tasks completed:**

* Decided the interface design of the website
* Started implementing the interface design of the website by using css
* Started implementing the code for user login.

**Expected tasks completed by the next week:**

* User login part need to be done
* Validation part need to fix by using PHP
* Documentation needed to be done.
* Add cage part need to be done.
* Interface need to be done

**Week 7**

**Tasks completed:**

* Complete the user login part
* Complete the login interface
* Started implementing the code for add cage. (to put the new animal inside the cage )
* Started implementing the code for delete cage (to delete cage
* Started writing the documentation for this project.

**Expected tasks completed by the next week:**

* Interface design for alert details need to be done.
* Update cage and update animal need to be done
* Documentation needed to be done.

**Week 8**

**Tasks completed:**

* Completed the design interface for alert details
* Completed the update part for the animal and cage part
* Started implementing the code to delete animal and cage from the website.

**Expected tasks completed by the next week:**

* delete cage and animal part need to be done
* documentation need to be done.

**Week 8**

**Tasks completed:**

* Completed the design interface for alert details
* Completed the update part for the animal and cage part
* Started implementing the code to delete animal and cage from the website.

**Expected tasks completed by the next week:**

* Need to generate the code for animal table and cage to keep a record data for animal id , cage id and animal record id .
* Need to create database server to the store the record of animal and cage.

**Week 9**

**Tasks completed:**

* Started implementing the code for animal table to keep a record data for animal id , cage id and animal record id
* Started implementing the database server for to store the records of animal and cage .

**Expected tasks completed by the next week:**

* Need to complete the table for animal record table
* Need to complete the database server the storing of records

**Week 10**

**Tasks completed:**

* Completed implementing the code for animal table to keep a record data for animal id , cage id and animal record id
* Database part is created and completed to store the animal records.

**Expected tasks completed by the next week:**

* Testing of database server
* Testing of the code
* Need to find out if any bugs occur.
* **Week11**

**Tasks completed:**

* Database was tested to check records are getting stored in the database or not.

**Expected tasks completed by next week:**

* Test the database.
* Validation part need to fix by using PHP
* Design needed to be finish by using CSS.
* Documentation needs to be done.

**Week 12**

**Tasks completed:**

* Checked all the bugs and fixed it out.
* User testing
* Checked the database.
* Design of the interface has been completed.
* Tested database server
* Documentation of the website is completed.