CS319 Analysis Report

Project Name

Zoo Master

Section 1

Group Members

Ege Berkay Gülcan

Uğurcan Uyumaz

Muhammad Hamza Khan

Kaan Kale

Contents

1. Introduction

Zoomaster is a desktop based application that assists zookeepers. The purposes of the application are to store the information of animals and plants in the zoo and notify the user to feed the animals and water the plants. In order to show the information of the species, the zookeeper has to add the information of the species into the system. New species can be added to the system, and the ones who are already in the system can be deleted or edited.

The reason of choosing this application is that it covers major topics of the course and the application will be implemented with OOP. Instead of developing advanced algorithms, the application is mostly based on the relationship among the classes and object oriented design pattern.

The context of this report is about the initial design of the project. The reports contains an overview of the application which elucidates how the application works, and how to use it. Then the requirements of the application which are separated into non-functional, functional, and pseudo are discussed in detail. To elaborate the functions of the application the report also includes system models of the application which are scenarios, models, and rough sketch of the interface.

2. Proposed System

2.1. Overview

Zoomaster will be implemented in Java, and the aim of the application is that it stores the information about species and their daily work. How to use the application will be as following:

- ➔ The main purpose of the application is to notify the zookeeper for feeding of animals and watering of plants. In order to do that, while adding the information about the species, the watering/feeding time is needed to be added.
- ➔ The application is protected with a password which is asked after starting the Zoomaster. The Zookeeper can change this password freely with a password change feature.
- ➔ There are 2 options available for adding species. These are adding plant or adding animal.
- ➔ The application stores the information about species and it contains the following:
    - ◆ Name
    - ◆ Latin Name
    - ◆ Country of Origin
    - ◆ Gender
    - ◆ Age
    - ◆ Light Time (For plants)
    - ◆ Feeding or Watering time
    - ◆ Picture
- ➔ After adding information it can be edited again or deleted. This is done through the search page which can be opened with a button on the main screen.
- ➔ Main screen will have daily work table for the current day, and there is also a monthly schedule on the main page from which the zookeeper can choose any day from and see the timetable for the chosen day.
- ➔ The search page provides common information about species. The zookeeper can search specific animals in there to find if it exists.
- ➔ The search page also includes detailed information about a specie. The detailed information can be reached after clicking on the specie.

2.2. Functional Requirements

2.2.1. Adding Animal/Plant

The user is prompted to choose between animal/plant types. In the following screen he/she is prompted to enter information based on the animal's/plant's type. For example, in a lion's case, the zoo keeper chooses mammal, then enters its feeding frequency (e.g. every 8 hours), age, gender, name, species, etc. In addition, the user may enter number of animals to add a bulk of animals.

2.2.2. Remove Animal/Plant

The user is able to delete an animal/plant by entering its name. If there is not an animal/plant by that name, he/she receives a warning message.

2.2.3. Reminders to the zookeeper

While adding a plant/animal to the system, the user will specify the times at which the animal needs to be fed or the plant needs to be watered. Then, the application will keep track of that time and on appropriate time will notify the user to take the action required.

2.2.4. Editing Animal/Plant Info

This application will further allow the user to edit info about any animal/plant. For instance, the user can modify the number of monkeys in the zoo or change the feeding time of lions.

2.2.5. Chage Password

The user is able to change the password of the application. But this requires to remember the old password. After confirmation the password changes into the new one.

2.3. Non-functional Requirements

2.3.1. Performance

Performance is maybe the most important non-functional requirement for our application. System will work properly because if application collapses then users can’t track animals or the correct time of their feeding times.

2.3.2. Extendibility

Development is an important issue in the application "Zoomaster". In order to add new properties and functions, implementation will be extendable.

2.3.3. Ease of Use

The application will be easily controlled. The selection from the main menu will be done by clicking with mouse or touchpad.

2.3.4. Open Source

It will be open source project we will share every step on GitHub.

2.3.5. Security

It is important because for every user we have username and password confirmation.

2.4. Pseudo Requirements

2.5. System Models

2.5.1. Scenarios

2.5.1.1. Scenario #1

Add Animal/Plant

When “Zoo Master” application starts, the zookeeper must enter a password in the field to continue. Then the main screen comes up and on that screen there is a calendar and buttons named “main”, “search” and “add”. To add an animal or plant zookeeper presses the “add” button, selects type of animal or plant, then enters name, age, feeding frequency, gender and country.

2.5.1.2. Scenario #2

Remove Animal/Plant

When “Zoo Master” application starts, the zookeeper must enter a password in the field to continue. Then the main screen comes up and on that screen there is a calendar and buttons named “main”, “search”, “add” and “password change”. To remove an animal or plant, the zookeeper presses the “search” button. Then the zookeeper enters the name of the plant or animal in the search field. After that, zookeeper presses the button “X”, next to the name of the animal/plant that the user wants to delete.

2.5.1.3. Scenario #3

Getting Detailed Information of a Species

When “Zoo Master” application starts, the zookeeper must enter a password in the field to continue. Then the main screen comes up and on that screen there is a calendar and buttons named “main”, “search”, “add” and “password change”. To get the detailed information of a specie, the zookeeper presses the “search” button. Then zookeeper enters the name of the plant or animal in the search field. After that zookeeper presses the button with the name of the species. A new page comes up with detailed information of the species.

2.5.1.4. Scenario #4

Edit Animal/Plant

When “Zoo Master” application starts, the zookeeper must enter a password in the field to continue. Then the main screen comes up and on that screen there is a calendar and buttons named “main”, “search”, “add” and “password change”. To get to the edit page of a specie zookeeper, presses the “search” button. Then the zookeeper enters the name of the plant or animal in the search field. After that zookeeper presses the edit button with the name of the specie. A new page with editable information comes up to the screen.

2.5.1.5. Scenario #5

Looking Reminders on Calendar

When “Zoo Master” application starts, the zookeeper must enter a password in the field to continue. Then the main screen comes up and on that screen there is a calendar and buttons named “main”, “search”, “add” and “password change”. After that zookeeper presses the day that he/she wanted to see the reminders of.

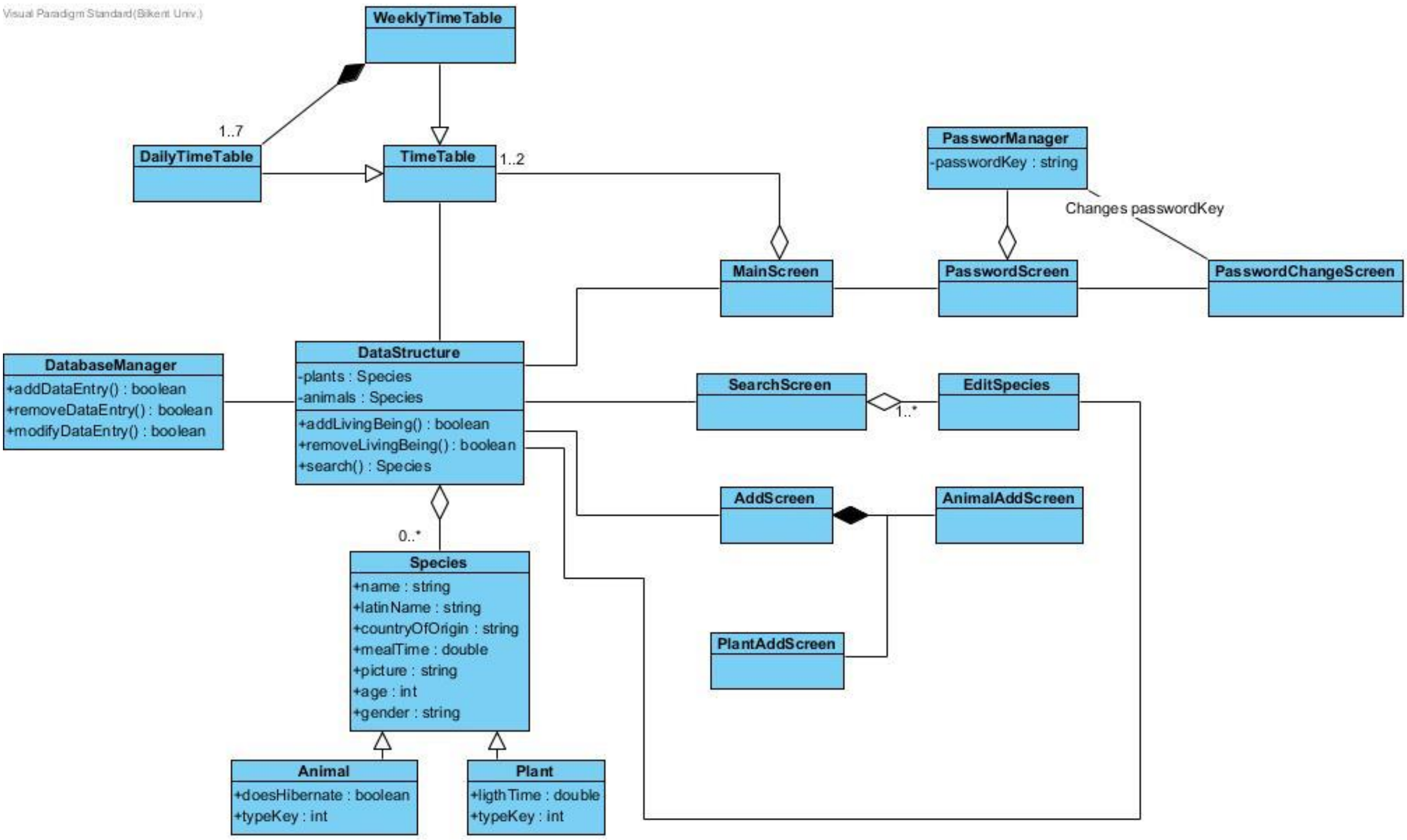
2.5.1.6. Scenario #6

Changing Password

When “Zoo Master” application starts, the zookeeper must enter a password in the field to continue. Then the main screen comes up and on that screen there is a calendar and buttons named “main”, “search”, “add” and “password change”. The user presses the “password change” button. After that a new page opens, and the user has to write his/her old password and new password, and the password changes to the new one.

2.5.2. Use-Case Model

2.5.3. Object and Class Model



The application uses a data structure to effectively organize the data of animals and plants.The data structure class also provides the data needed for other classes. Since this is a data driven application, this data structure is the core class of the system.

To store this data, through the DatabaseManager, the system uses a SQL database. The database manager queries the database to add, delete and edit the entries.

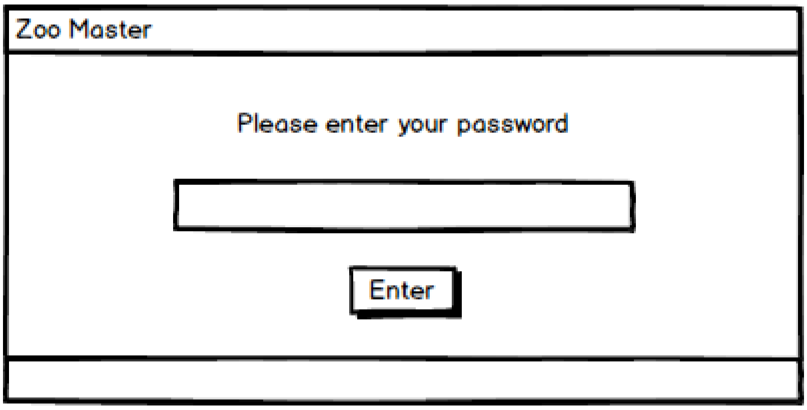
Through the data obtained from the data structure class, time table classes creates a schedule for the zookepers such as feed the lions at 1 pm, water flowers at 10 am, etc.

Password manager stores the password and manages the authentication process. It also manages the password change process.

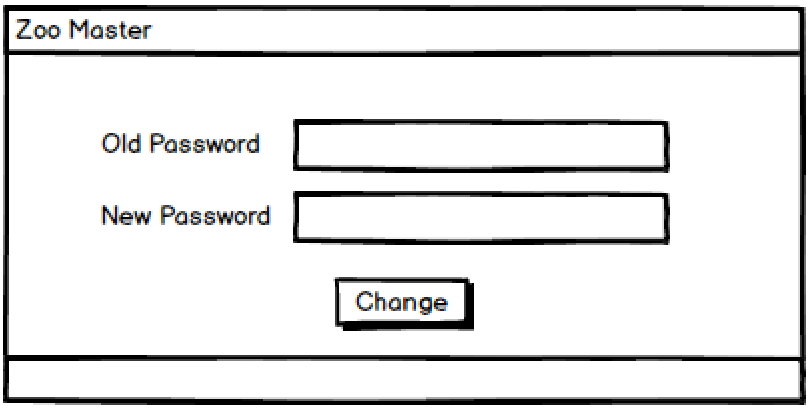
2.5.4. Dynamic Model

2.5.5. User Interface

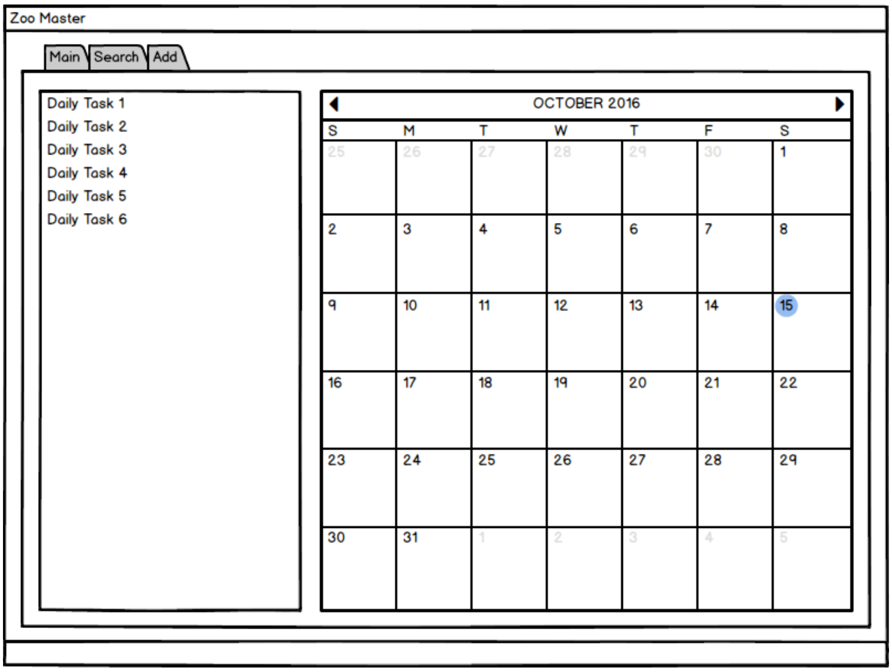
The application has a GUI for user interaction. The first screen of the application is the password page. The page consists of a text field for password input, a button to submit the password and password change button.



In the password change screen, there are 2 textboxes and a button. One of the textboxes is for inputting the old password and the other one is to input the new password. Button is for submitting the new password.



After entering the password, if it is correct the user is directed to the main page. The page contains a calender and a daily task bar. When the user clicks the desired day, task bar shows the tasks that are needed to be completed that day, e.g., watering plants, feeding animals.



Daily Task 1
Daily Task 2
Daily Task 3
Daily Task 4
Daily Task 5
Daily Task 6

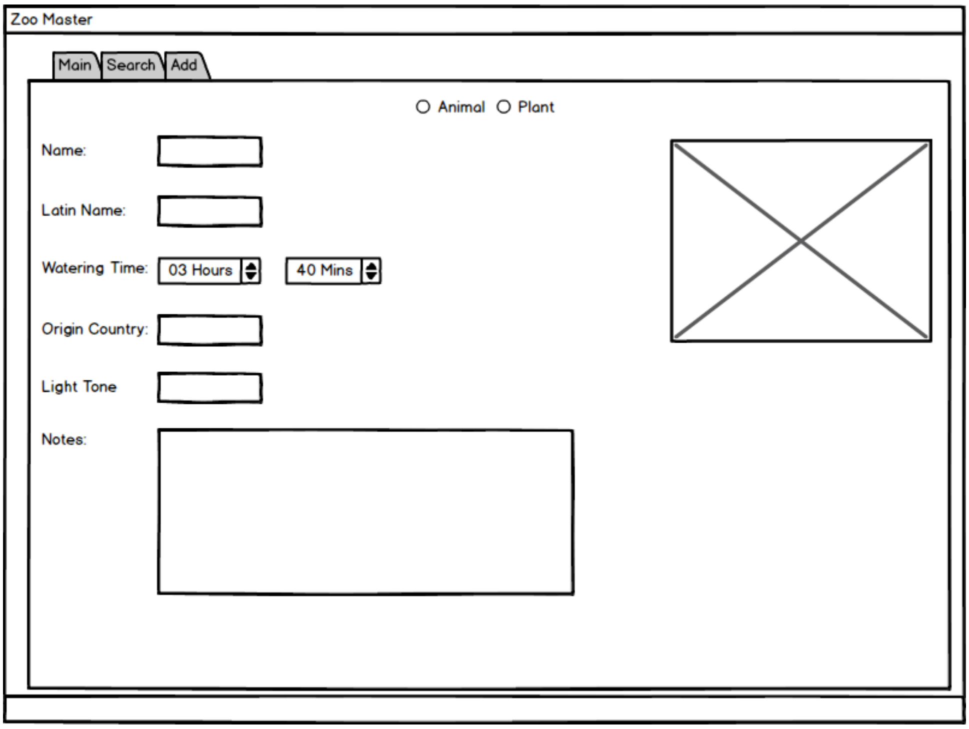
OCTOBER 2016

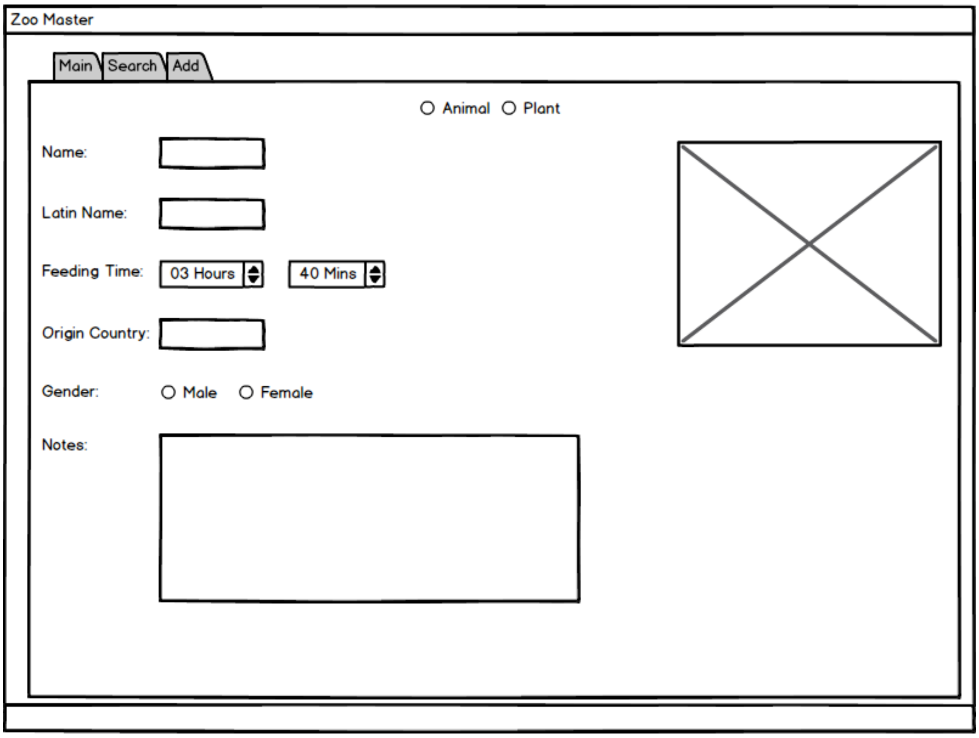
S	M	T	W	T	F	S
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

It also has 3 tabs to change between main, add and search screens.

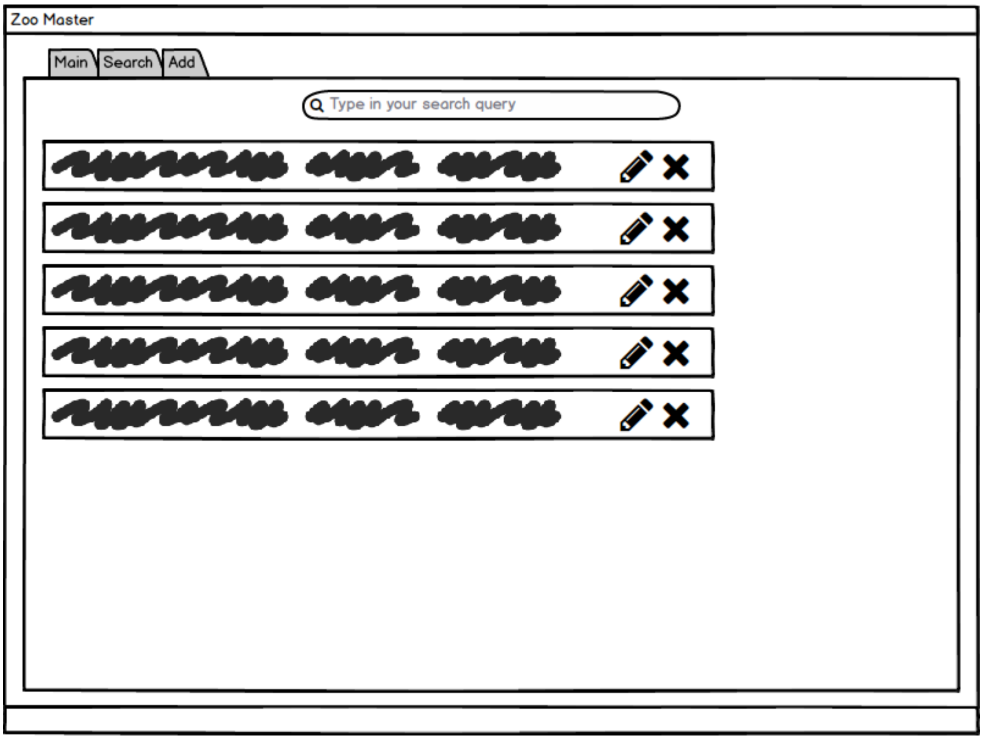
.

In add screen, there are 2 radio buttons to change between plant and animal add screens. The screens has common text fields for information such as name, latin name and country of origin. Another common property of the two pages is adding image. However, since plants and animals have different properties along with their common ones, the pages have different text fields such as light tone and time for plants,and if it hibernates or not for animals.





The last screen is the search screen. The search screen only contains a search bar to search the animal and plant database. When the user searches the animal or plant, the screen displays bars to represent the searched animal or plant. The bars also has buttons on it to edit the animal or plant and delete it.



3. Glossary

4. References