



CS319 Object Oriented Software Engineering

Revised Analysis Report

ZooMaster

Instructor: Bora Güngören

Group Members

Ege Berkay Gülcan	21400461
Uğur Can Uyumaz	21301417
Muhammad Hamza Khan	21402884
Kaan Kale	21000912

Table of Contents

1. INTRODUCTION	3
2. OVERVIEW	3
3. FUNCTIONAL REQUIREMENTS	4
4. NON-FUNCTIONAL REQUIREMENTS.....	6
4.1. PSEUDO REQUIREMENTS.....	7
5. SYSTEM MODELS.....	7
5.1. SCENARIOS	7
5.1.1. Scenario #1.....	7
5.1.2. Scenario #2.....	7
5.1.3. Scenario #3.....	8
5.1.4. Scenario #4.....	8
5.1.5. Scenario #5.....	9
5.1.6. Scenario #6.....	9
5.2. USE-CASE MODEL	10
5.3. OBJECT AND CLASS MODEL	11
5.4. DYNAMIC MODEL	12
5.4.1. Sequence Diagrams	12
5.4.2. Activity Diagram.....	18
5.5. USER INTERFACE	20

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 contain 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. 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 species. The detailed information can be reached after clicking on the species.

3. Functional Requirements

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.

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.

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.

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.

Change Password

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

4. Non-Functional Requirements

Performance

- Very elegant and efficient algorithms will be used in the development of the application to make it fast in speed and to make it consume less memory.
- The amount of time taken for the searches through the databases will be kept at as much less as much possible.

Robustness

- Rigorous testing will be done to ensure that there are no bugs in the application.
- All boundary line and unusual cases will be covered to make sure that the application doesn't crash under any circumstance.

Extendibility

- The code will be properly documented and general programming conventions will be followed to make it easier to build on top of the existing code, in the future.

Ease of Use

- The learning curve of the application will not at all be steep.
- Labels and notifications will be appropriately used to make it easier for the user to use the application.

Open Source

- The source code of the application, along with all of the development models, will be shared on github under MIT Open Source License

Security

- A password will be required to access the application. This is to ensure that no unauthorized person can mingle with the database of species or the timetable of the zookeeper.

4.1. Pseudo Requirements

- The application should run in the background, monitoring the time, and give the notifications to the user, even though the user might have closed the application.
- Learning curve shouldn't be steep because the zookeepers might have limited knowledge in computers.

5. System Models

5.1. Scenarios

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

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.

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 species, 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.

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 species 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 species. A new page with editable information comes up to the screen.

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.

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.

5.2. Use-Case Model

Visual Paradigm Standard (Bilkent Univ.)

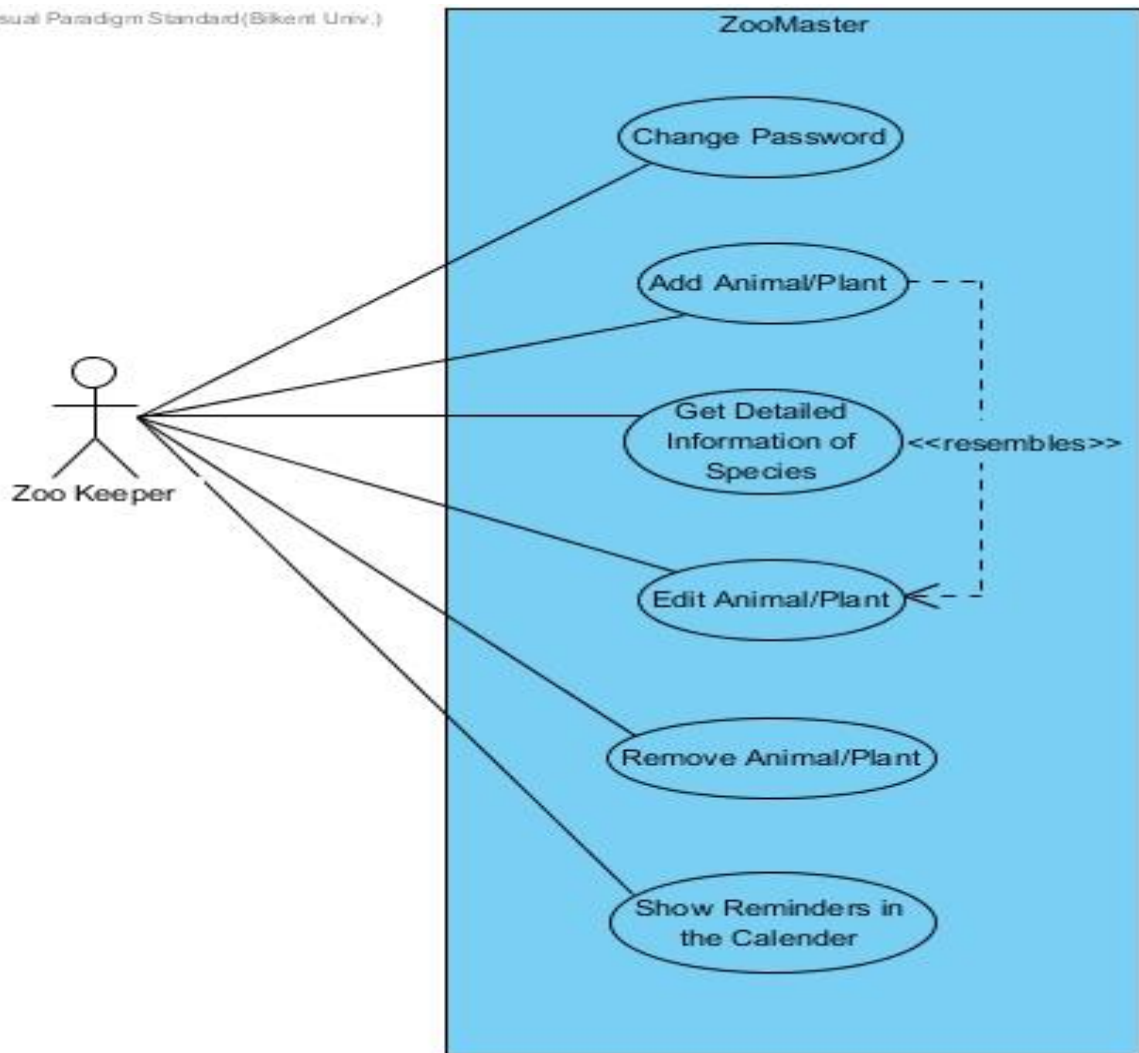


Figure 1: Use-Case Diagram

The only actor of the application is the zoo keepers. There are 6 use cases for the application which are adding animal/plant, changing password, getting detailed information of species, editing animal/plant, removing animal/plant and showing reminders in the calendar.

5.3. Object and Class Model

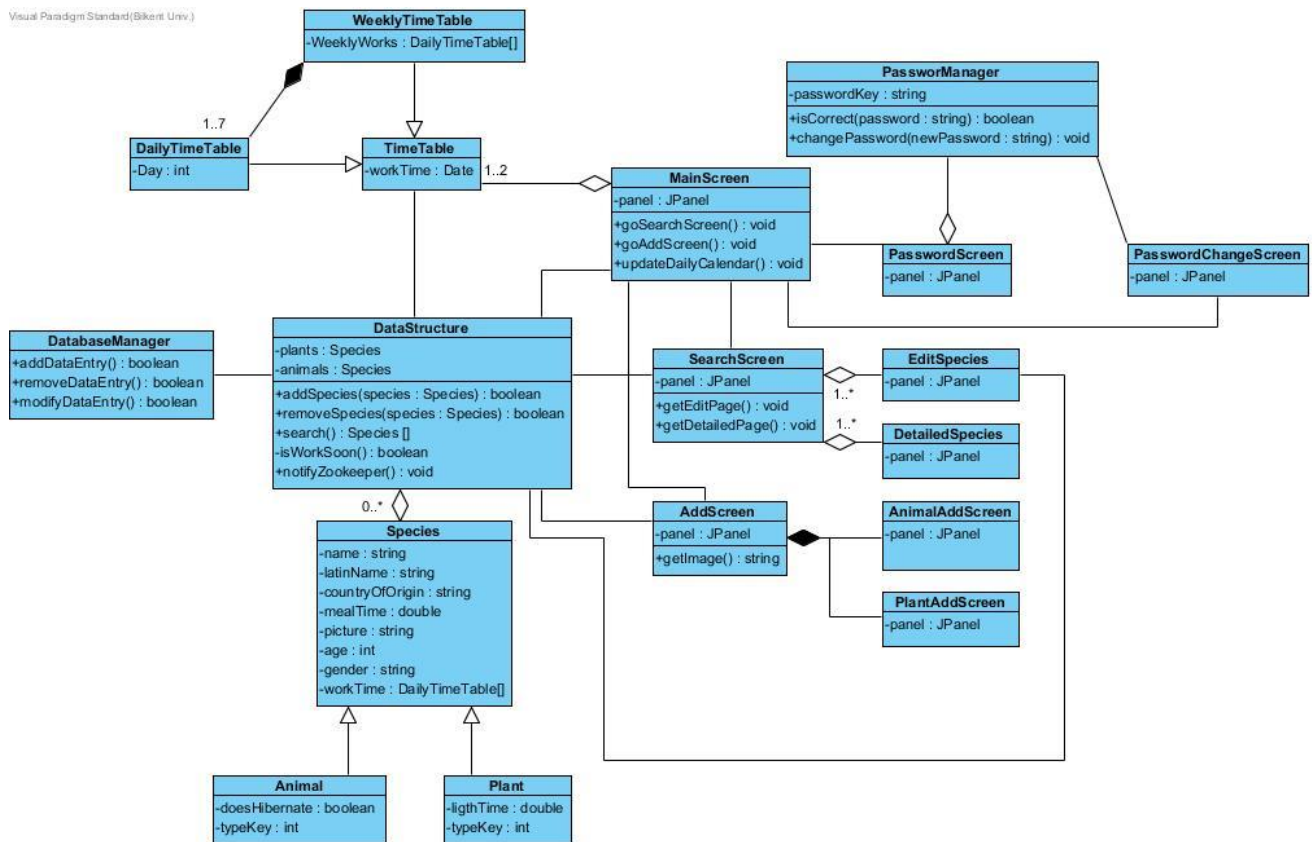


Figure 2: Class Model

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 zookeepers 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.

5.4. Dynamic Model

5.4.1. Sequence Diagrams

5.4.1.1. Add Animal/Plant

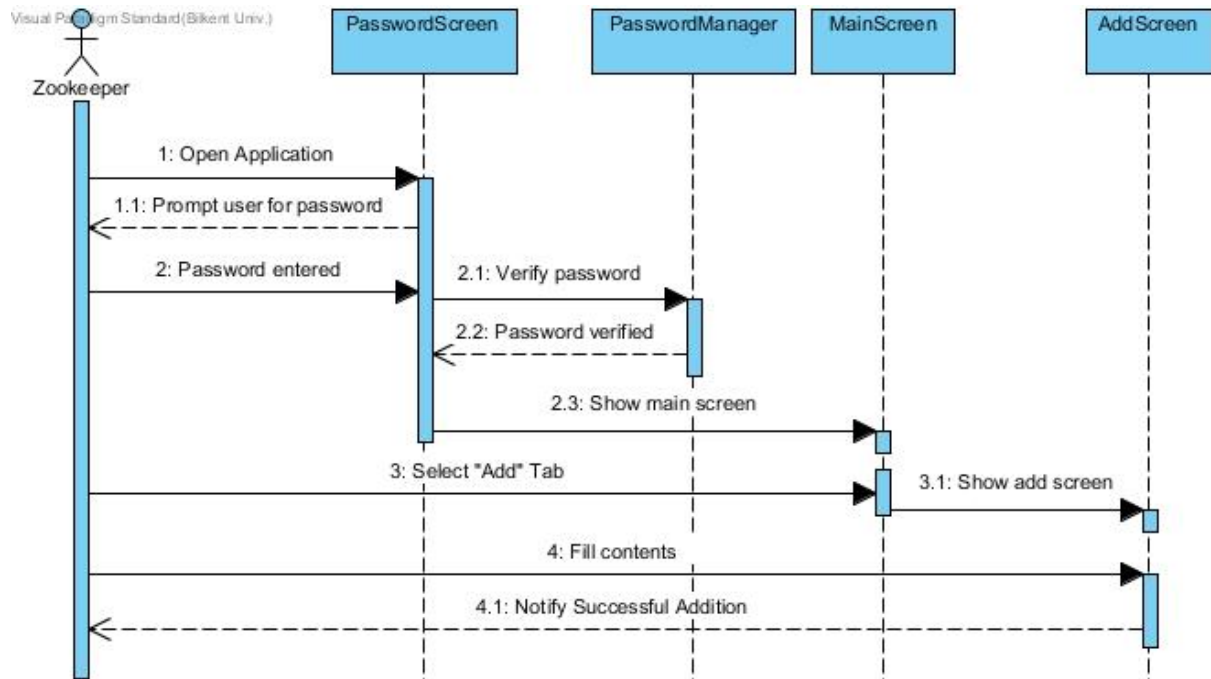


Figure 3: Sequence Diagram of Adding Animal/Plant

Figure 3 explains a scenario in which the user adds a new animal or plant to the system. The user opens the application and the application prompts the user for password for authentication. If the password is correct, the user is directed to the main screen. Then the user selects the add tab and the system shows the add screen. After that, the user proceeds to enter information about the animal/plant he/she is adding. Following it, the user is notified about the successful addition of the animal/plant.

5.4.1.2. Change Password

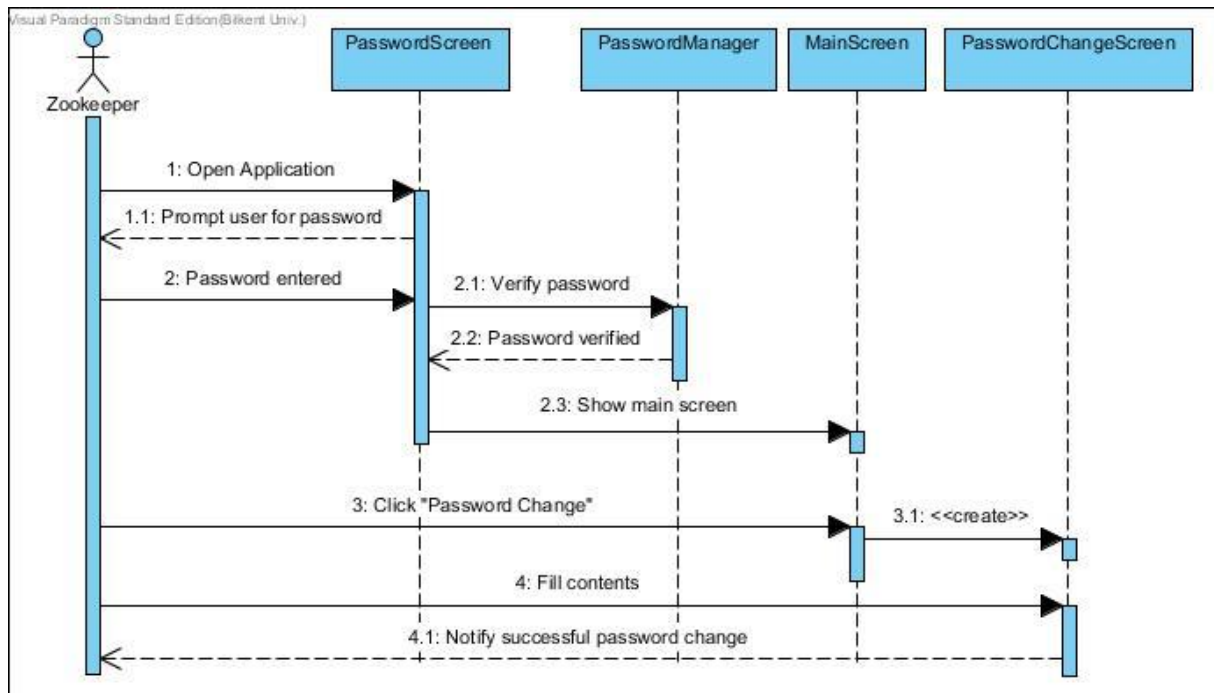


Figure 4: Sequence Diagram of Changing Password

The diagram on figure 4 describes a scenario in which the user changes the password of the system. To do it, firstly, the user opens the application and enters the password to receive authentication. If the password is correct, the user is directed to main screen. From there user clicks the button for password change. Then the system creates the password change screen and directs the user to it. After that, the user enters the old and new password and then the system notifies the user about the successful change.

5.4.1.3. Edit Animal/Plant

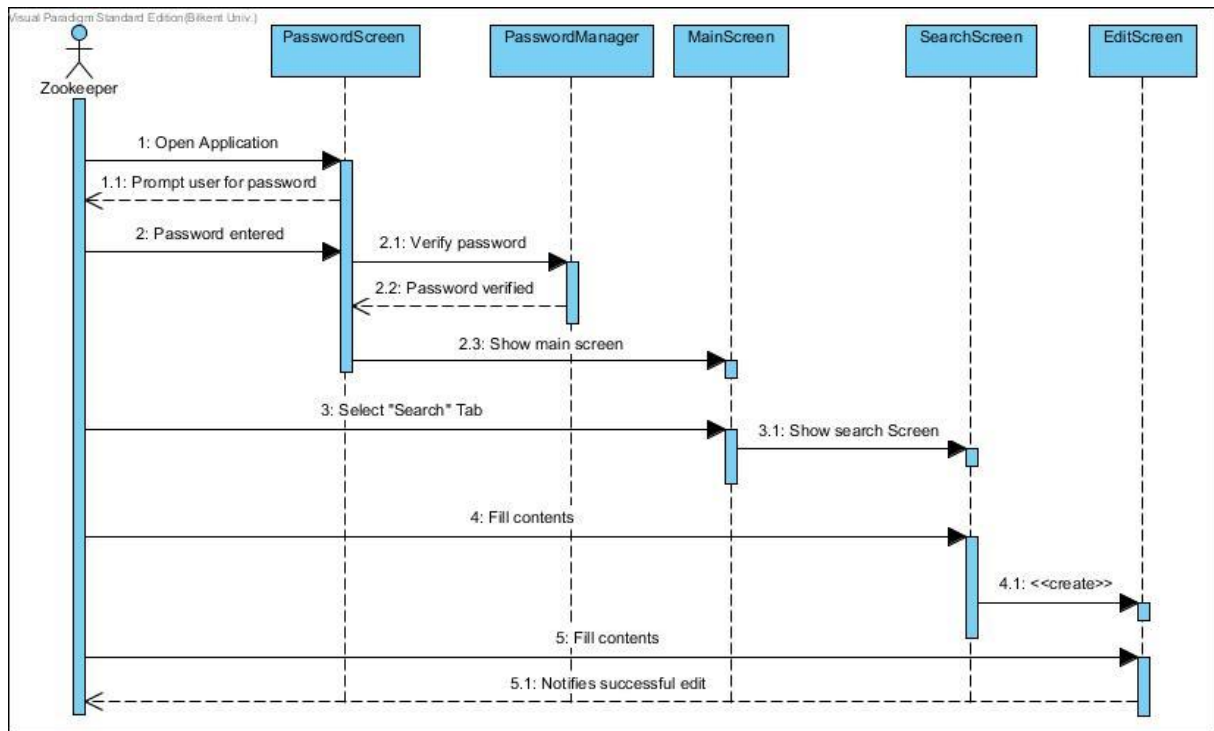


Figure 5: Sequence Diagram of Editing Animal/Plant

In figure 5, the diagram explains a scenario in which the user edits an existing animal or plant. Firstly, the user opens the application and enters prompted password. If the password is correct, the application shows the main screen to the user. From the main screen, user selects the 'Search' tab. In the search tab, the user enters name of the animal/plant that he/she wants to edit. The system creates fields for the species. Then the user clicks the edit button and makes changes. Following that, the system notifies the user for successful edit.

5.4.1.4. Get Detailed Information of a Species

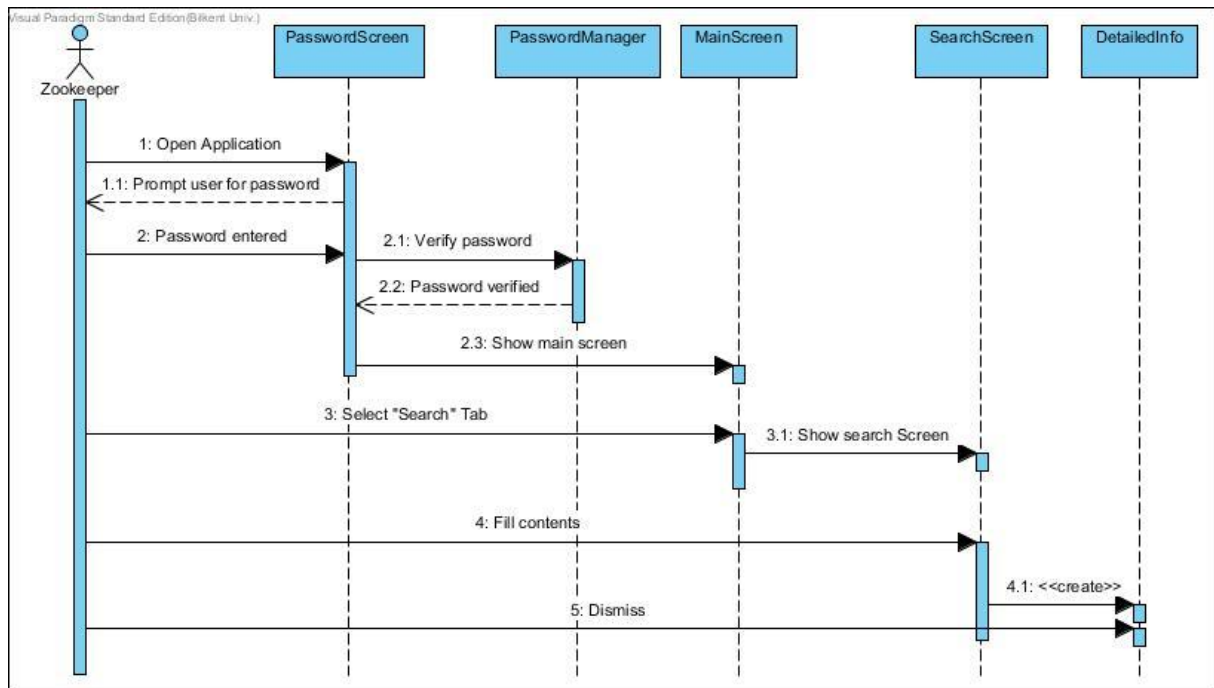


Figure 6: Sequence diagram of Getting Detailed Information of a Species

The diagram in figure 6 represents a scenario in which the user gets detailed information of a species. After opening the application, the system prompts the user for a password. After successful password entry, the user is directed to the main screen. In the main screen, the user selects the tab for searching animals and plants. In the search screen, the user fills the search bar and the system creates a field for the animal/plant. After clicking the field, a new page is created that contains detailed information about that animal/plant.

5.4.1.5. Looking for Reminders on Calendar

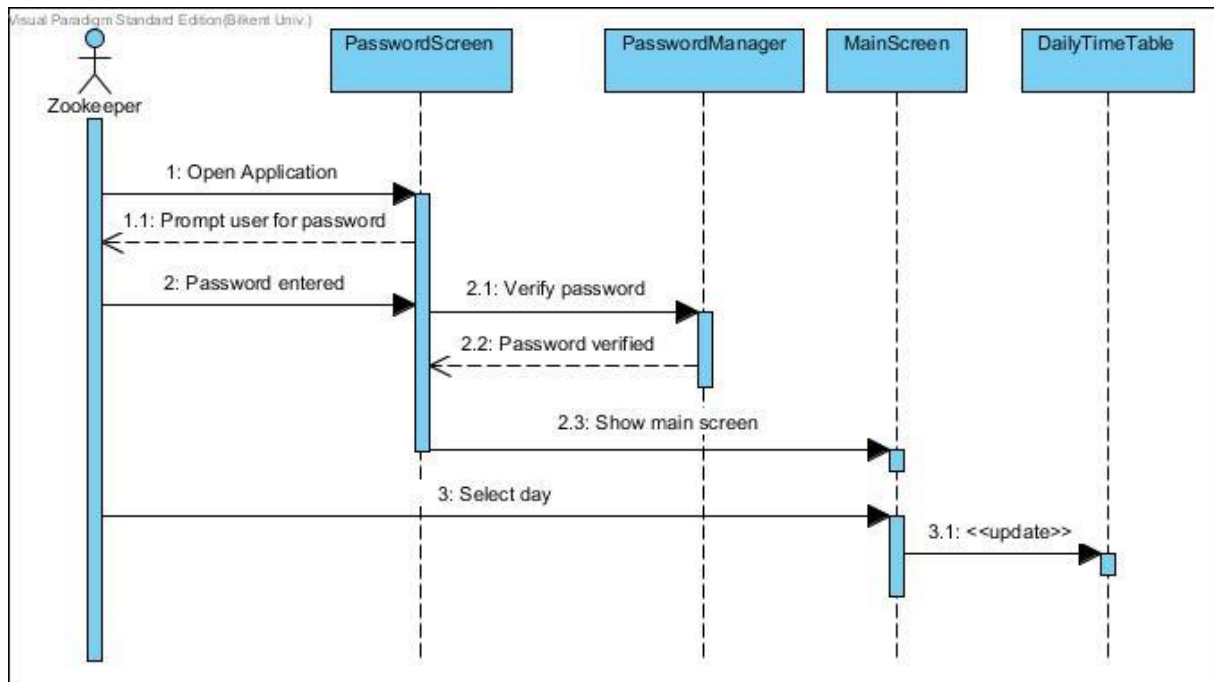


Figure 7: Sequence Diagram of Looking for Reminders on Calendar

In figure 7, the diagram describes a scenario in which the user looks for reminders in the calendar. To do this, the user first enters the password and if it is correct the system directs the user to the main screen. In the main screen, the user selects a day. Following that the system updates the screen to show the reminders for that day.

5.4.1.6. Removing Animal/Plant

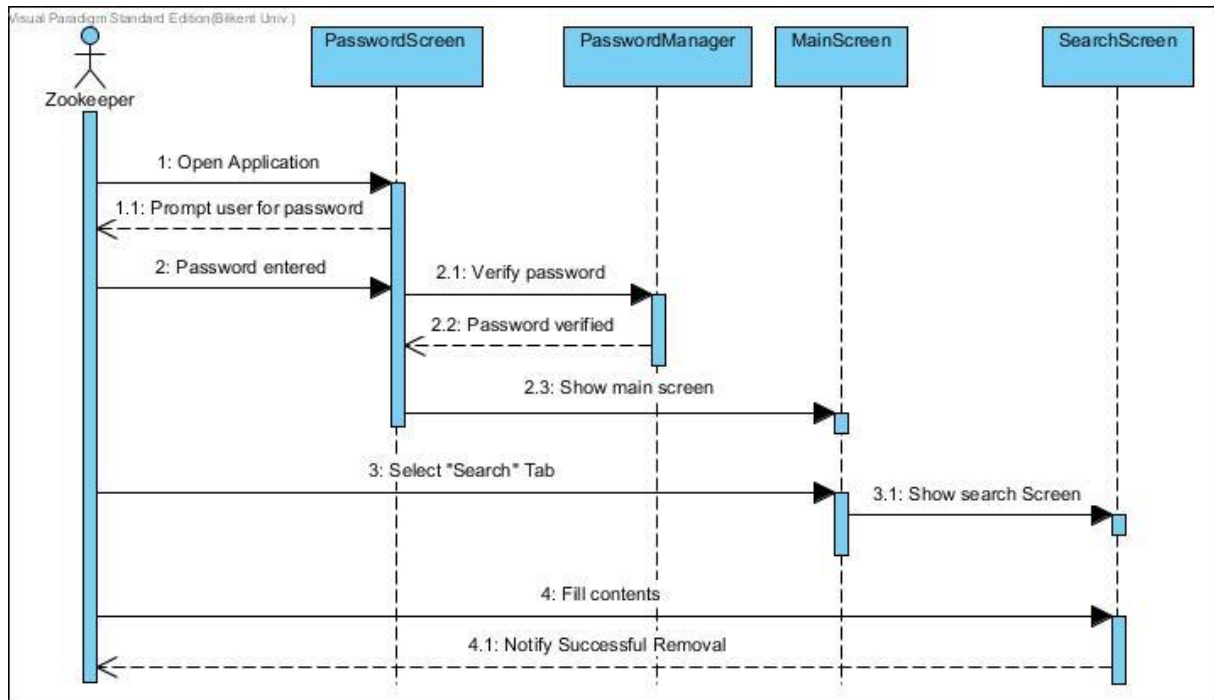


Figure 8: Sequence Diagram of Removing Animal/Plant

The diagram in figure 8 explains a scenario in which the user removes an animal/plant. After opening the application, the system prompts the user for a password. If the entered password is correct, the user is directed to the main screen. In the main screen, the user selects the 'Search' tab. Then the user enters the name of the animal/plant in the search bar and the system creates a field for that animal/plant. The user clicks the remove button and the system notifies the user about the successful removal.

5.4.2. Activity Diagram

Visual Paradigm Standard Edition (BitKent Univ.)

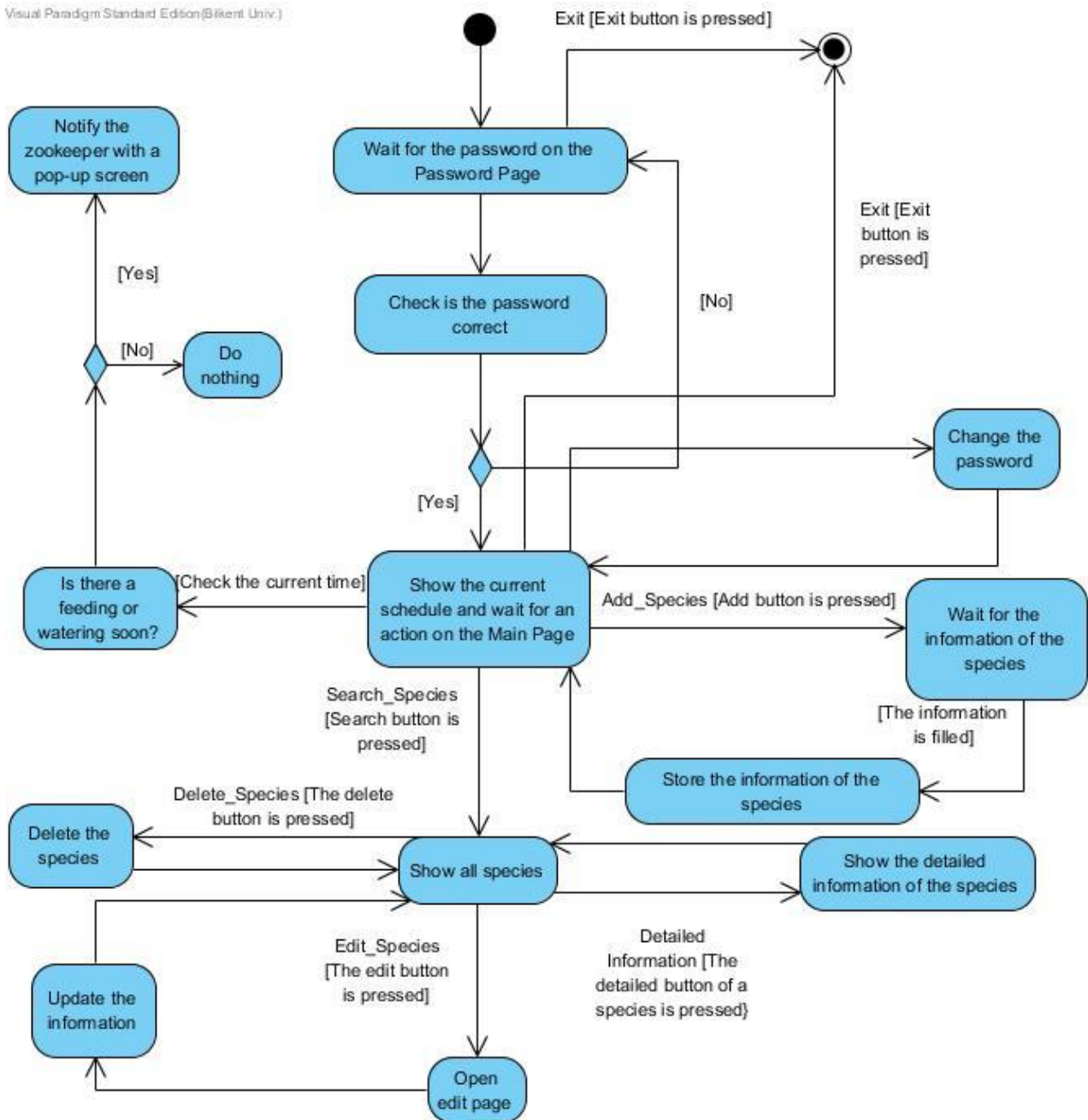


Figure 9: Activity Diagram

The diagram in figure 9 shows the general activity flow of the application. At application start the user is prompted for a password. If the password is correct, the system directs the user to the main screen which shows the schedule. In the main screen the user could change the password, add new animals/plants and search

them. Also the in the main screen, the application checks the current time and if there is a scheduled event, notifies the user about it. If the user wants to change the password, he/she is directed to the password change screen and in there he/she changes the password. If the user wants to add animals/plants, he/she fills information about the species then the species is added. If the user wants to delete a species he/she goes to the search screen and searches for the species and then deletes the species by pressing the delete button. If the user wants to edit a species, he goes to the search screen and by pressing the edit button edits the information of the species. Finally, if the user wants to see detailed information about the species, he/she searches the species in the search screen and the system shows detailed information about the species.

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.

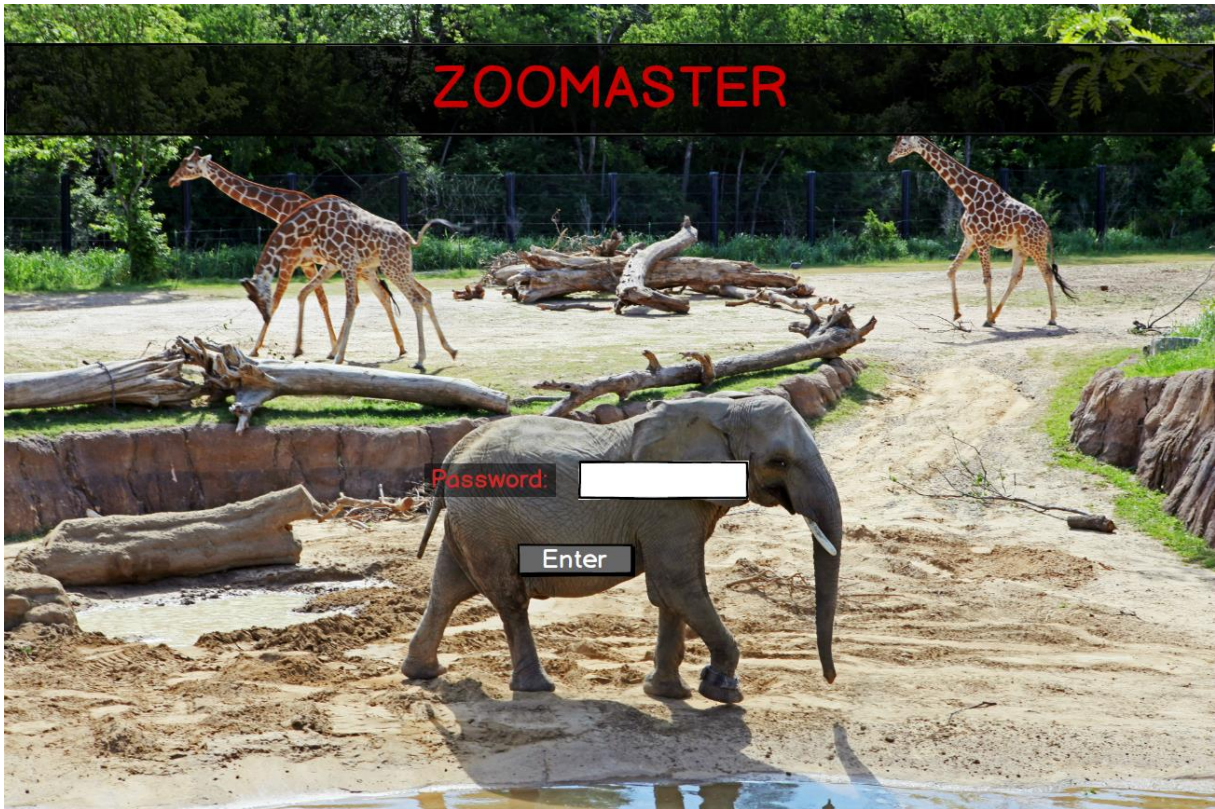


Figure 10: Password Enter Screen

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.

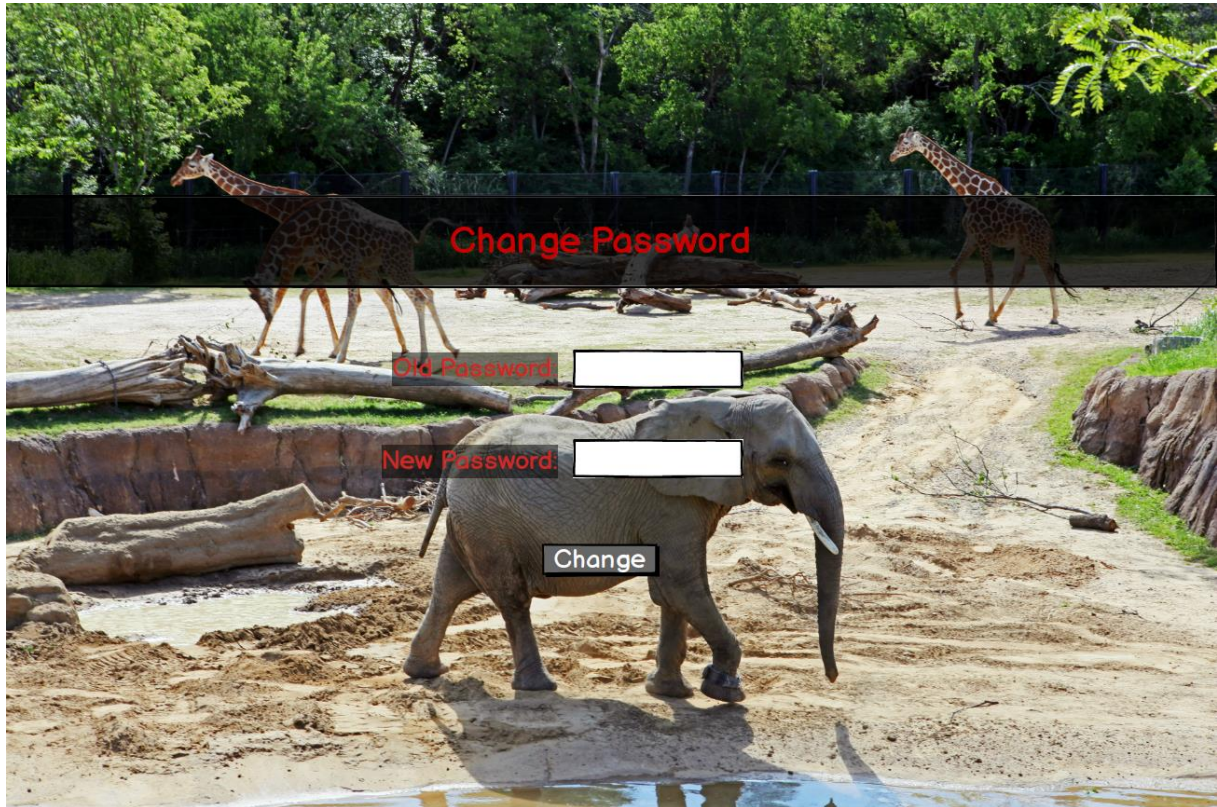


Figure 11: Password Change Screen

After entering the password, if it is correct the user is directed to the main page. The page contains a calendar 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. It also has 3 tabs to change between main, add and search screens.

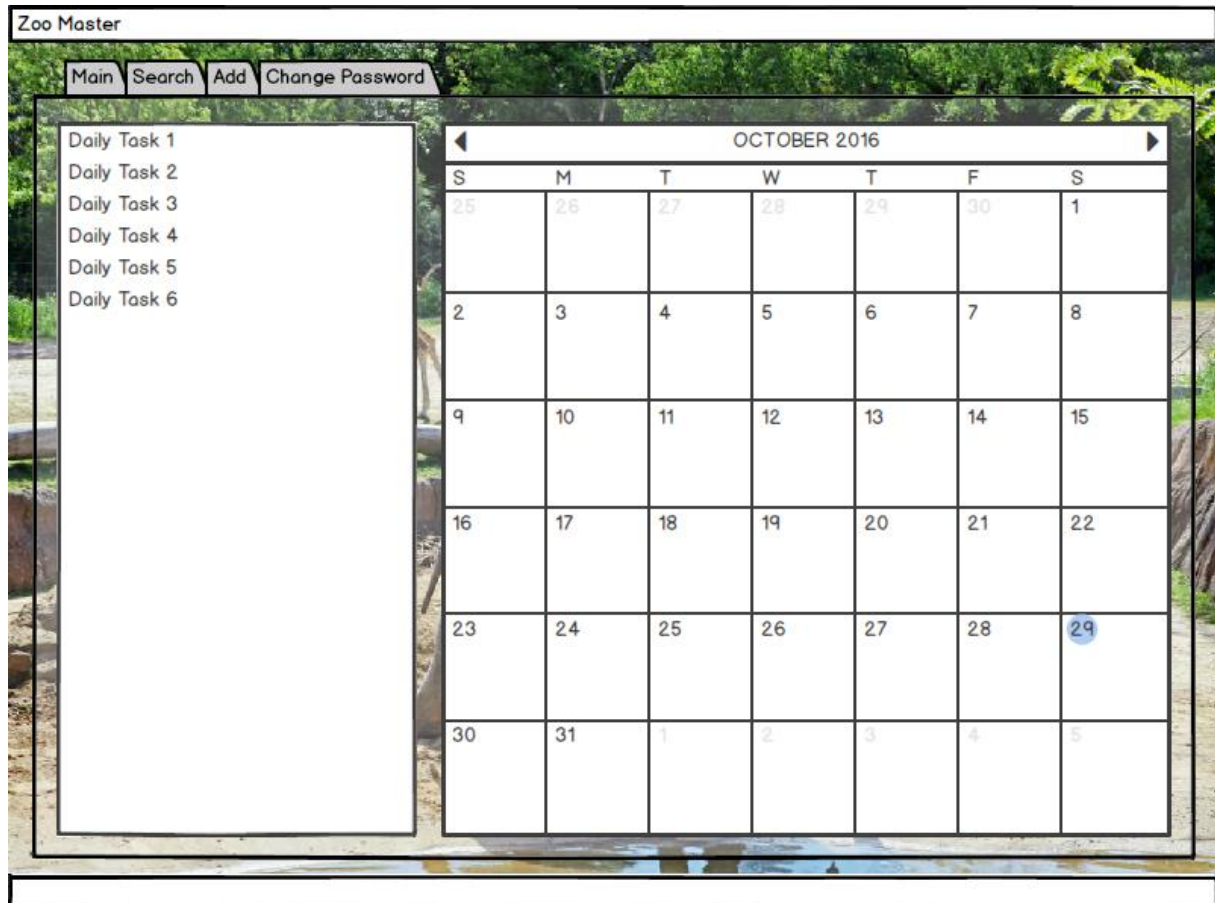


Figure 12: Main Screen

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.

Zoo Master

Main Search Add

☐ Animal ☐ Plant

Name:

Latin Name:

Feeding Time: 03 Hours 40 Mins

Origin Country:

Gender: ☐ Male ☐ Female

No. of animals: 3

Notes:

Select Image

Figure 13: Animal Add Screen

Zoo Master

Main Search Add

☐ Animal ☐ Plant

Name:

Latin Name:

Watering Time:

Origin Country:

Light Tone:

No. of plants:

Notes:

Select Image

Figure 13: Plant Add Screen

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 have buttons on it to edit the animal or plant and delete it.

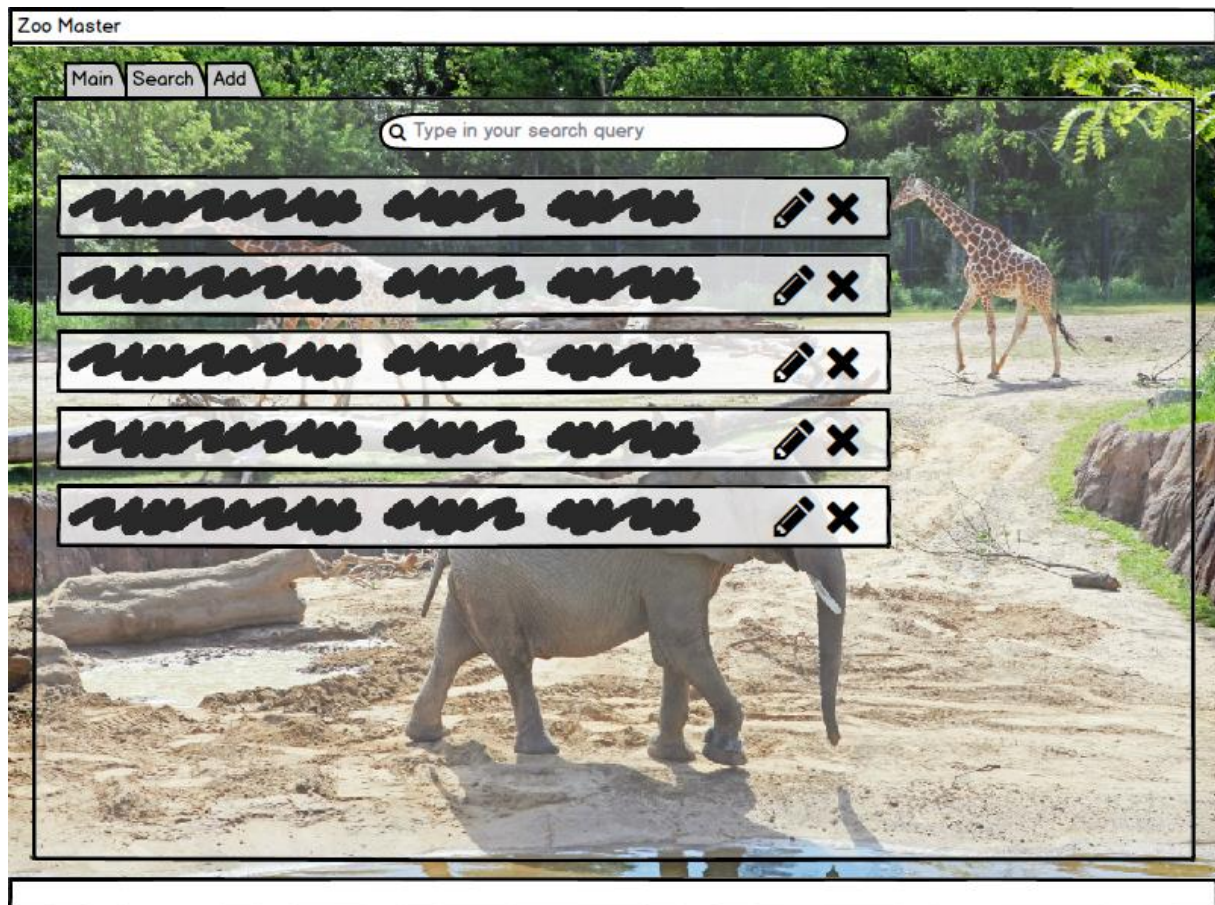


Figure 14: Search Screen