CS353 PROJECT PROPOSAL

ZOO DATABASE MANAGEMENT SYSTEM

Group Members

Ramazan Melih DİKSU - 21802361
Aleyna SÜTBAŞ - 21803174
Berk TAKIT - 21803147
Sarp TEKIN - 21600813

Contents

1. Introduction

This report is about the proposal of Zoo Database Management System which will be used by visitors and employees of the zoo. The report will contain the purpose of the project, basic functional requirements and limitations of the project.

The next chapter will introduce the project's properties, why this project is needed and how it will be implemented.

In the requirements chapter, requirements will be divided into non-functional and functional requirements. Functional requirements specify what the project will allow users to do and non-functional requirements will specify back-end requirements like security & privacy, usability, performance and reliability of the project.

In the limitations chapter, the limitations of different user types in the project will be discussed.

In the last chapter the design of the project will be shown with an E/R diagram to show the relation of entities in the project and illustrate the logical structure of the database.

2. Project Definition

This project is a web-based database management application that will be used by both the employees and visitors of the zoo. The aim of the project is to increase the efficiency of communication and planning between employees and visitors.

There will be 4 different types of users which all have their unique features. Employees will be divided into 3 sub groups keeper, veterinarian and coordinator which will all have their unique features in the app that they can use. Coordinators will be able to create events like group tours, educational programs or a conservation organization. Furthermore, they will be able to invite veterinarians to educational programs and assign keepers to the cages and respond to the complaint forms. Keepers will be able to request treatment for an animal by the veterinarians and schedule training for animals. Veterinarians will be able to accept or reject invitations to educational programs by the coordinators and treatment request by the keepers. Visitors will be able to attend group tours by paying the price of the tour they desire and make donations to the conservation organization. Furthermore, they will be able to write complaint forms and comment on group tours as well.

Since there are many different users and many different request combinations, a well structured database will be needed for the project. This

database will allow the data to be organised and retrieved more efficiently then a simple file system.

3. Functional Requirements

This section will include the functional requirements of the Zoo Database System with respect to its end-users given below.

3.1. Visitor

- Visitors should be able to create memberships.
- Visitors should be able to upgrade their memberships (gold, silver etc.)
- Visitors should be able to view the dates and schedules of group tours.
- Visitors should be able to create reservations for group tours.
- Visitors should be able to make reservation payments.
- Visitors should be able to comment about their group tour experience.
- Visitors should be able to file complaint forms.
- Visitors should be able to browse through the animals residing in the zoo.
- Visitors should be able to make donations to conservation organizations.
- Visitors should be able to view the shops in the zoo.

3.2. Keeper

- Keepers should be able to regularize the food served in animal meals.
- Keepers should be able to browse through the animals residing in the zoo.
- Keepers should be able to schedule training for the animals.
- Keepers should be able to request treatment for animals.

3.3. Veterinarian

- Veterinarians should be able to view invitations.
- Veterinarians should be able to view the dates and schedules of educational programs.

- Veterinarians should be able to browse through the animals residing in the zoo.
- Veterinarians should be able to accept treatment requests.

3.4. Coordinator

- Coordinators should be able to create events.
- Coordinators should be able to invite veterinarians to educational programs.
- Coordinator should be able to browse through the animals residing in the zoo.
- Coordinators should be able to view event participant lists.
- Coordinators should be able to view and respond to complaint forms.
- Coordinators should be able to assign cages to keepers.

4. Non-Functional Requirements

4.1 Security & Privacy

- Users will choose passwords that have at least one capital letter and special character. Also, passwords must consist of at least eight characters.
- Users will be able to use two-factor authentication to protect their personal and credit card information.
- User data will not be available to third parties.
- Visitors will be able to comment on group tours anonymously.

4.2 Usability

- The website will be user-friendly. The flow of information about the events will be meaningful enough to follow.
- The placement and the design of the buttons will ensure a fast and easy transaction for the customers.

4.3 Performance

- Considering the possibility of having a large amount of data, the database system should be able to process this data as quickly and efficiently as possible.

4.4 Reliability

- The website will go through maintenance once a week in the night time to be able to provide a smooth experience.
- The database system will be able to hold a large amount of data to keep up with a possible increase in requests.

5. Limitations

- Keepers can only request treatment for animals they are looking after.

- A cage or a shop can only be in one area.

- Visitors can only comment on group tours that they have attended.

- Visitors can only have one type of membership active at a time.

- A coordinator may assign a cage to more than one keeper.

- Keepers can regulate food only for the cage/s they are assigned to.

- Keepers can only schedule training for animals they are looking after.

- An animal may not be assigned to more than one cage.

6.ER Model
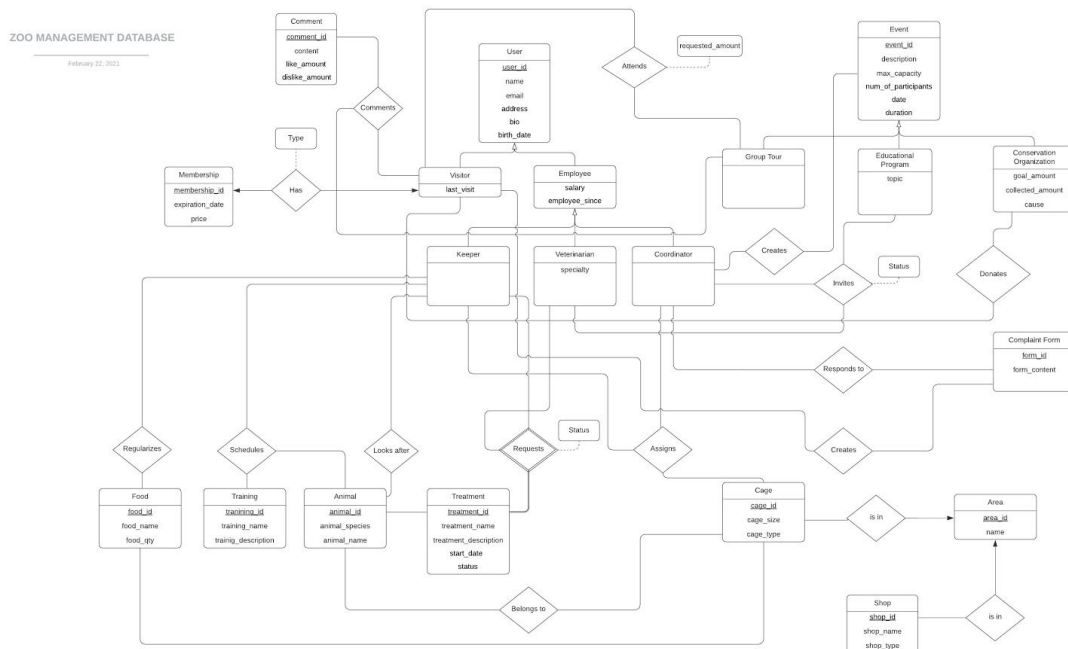
7.Website

Our website ,which includes a full scale file of the ER Model, can be found at berk-t.github.io.