BILKENT UNIVERSITY

SPRING 2021

CS353 TERM PROJECT

PROPOSAL

GROUP 31

Zoo Database Management System

Mehmet Yaylacı - 21802347 - Section 3

Yiğit Erkal - 21601521 - Section 2

Selin Kırmacı - 21802177 - Section 3

Selcen Kaya - 21801731 - Section 3

Table of Contents

Introduction **2**

Application System **2**

Description **2**

Why is a database needed? **3**

Requirements **3**

Functional Requirements **3**

Visitors 4

Employees 4

Keepers 4

Veterinarians 4

Coordinators 5

Non-Functional Requirements **5**

Constraints **6**

Limitations **6**

Conceptual Design **7**

Webpage **9**

References **9**

1.0 Introduction

This report is a project proposal for the CS 353 Database Systems term project. In this project, we are going to design and implement a zoo database management system. With this platform, the visitors of the zoo will be able to gain information about the zoo and the hosted events, and the employees of the zoo will be able to keep track of animals and events, and be able to interact with the visitors with comments on event pages. Further description of the project's application system and clarifications on system's requirements and limitations are provided below. Finally, this report gives a clear picture of the conceptual design of the proposed project with an ER diagram.

2.0 Application System

The description of the application system is presented below as well as the reasons for using a database for the project.

2.1 Description

The aim of our project is to create a database management system for a zoo. The users of the system will be visitors and different types of employees. We will also implement events and the complaints of the visitors if they have any. Our system will also include the cages and foods of the animals and different actions performed by different employees. Currently, there are three different types of employees in the system: keepers, veterinarians and coordinators. The keepers are the employees responsible for the animals of the zoo. In the system, they will be assigned to cages and animals to look after, food options that they can regulate and request treatment for the animals from veterinarians. The veterinarians are responsible for the health and general well-being of animals they are treating. They also will be able to participate in educational events as a speaker if they are invited. Every animal will be kept in a cage. The coordinators are responsible for events and keep track of keepers. They can create three types of events: group tours, educational programs or conservation organizations. The coordinators will invite veterinarians to educational programs. The coordinators are able to respond to comments or complaint forms created by visitor users. The visitors can see the available events, attend group tours where they pay the required amount of money for educational programs and they can make donations to conservation organizations. They are able to post comments or create complaint

forms about group tours. The visitors will also have the chance to see the comments made by other visitors.

2.2 Why is a database needed?

Our proposed system consists of different entities and the entities have numerous rows. Examples of these entities can be given as: keepers, cages, visitors and these entities can have many rows. Databases are used to collect, retrieve, sort and manipulate data. We need to store aforementioned entities and we need to sort them in a structure. When a user opens our system, they should be able to 'login'. Our system should be able to connect the user with the stored username and password and to achieve this we need a database. When a visitor writes a complaint we should be able to collect the text and we should be able to connect the complaint to an event. When a user wants to see the events and when a user wants to see the comments written for a tour, our system needs a database that will retrieve and show the necessary data. Finally, it can be said that a database is necessary for the proposed system to function.

3.0 Requirements

The requirements of the system are explained in three sections below in subheadings named functional requirements, non-functional requirements and constraints.

3.1 Functional Requirements

The common functional requirements for any user of the system are as follows.

- A registered user can be a visitor or an employee, and an employee must be either a keeper, veterinarian or coordinator.
- A registered user can change their password.
- A registered user can edit their own information such as displayed name.
- Any user can look at available events of the zoo.

The divergent functional requirements for each type of user are described below with specified user types as headers.

3.1.1 Visitors

- Attend group tours by paying the requested amount of money for it.
- Request refund for any group tour payments made before the event started.
- Make donations to conservative organization events.
- Create complaint forms about group tours they attended.
- See the complaint forms created by themselves.
- See the responses to their complaint forms from coordinators.
- Write comments on group tours.
- See other visitors' comments on any group tour.
- Save credit card information after first payment for any future payment.

3.1.2 Employees

There are three different types of employees as mentioned before; keepers, veterinarians and coordinators. Different functionalities for each type of employee are explained below.

3.1.2.1 Keepers

- Request treatment for an animal belonging to a cage they are responsible for.
- Schedule training for animals.
- Regularize food for cages.
- Check assigned cages.

3.1.2.2 Veterinarians

- Accept or reject invitations to educational programs from coordinators.
- Accept or reject animal treatment requests from keepers.

3.1.2.3 Coordinators

- Create 3 types of events: group tour, educational program and conservation organization.
- Edit or cancel events.
- Invite veterinarian employees to educational program events.
- Assign keeper employees to animal cages.
- Respond to complaint forms created by visitors.

3.2 Non-Functional Requirements

The non-functional requirements of the project are listed below, in no particular order.

- **Scalability**: System should be open to improvement and be able to incorporate newer entities or more advanced entity relationships. The size, popularity, visitors or event types of the zoo might increase in the future and the system should be able to support the necessary improvements.
- **Security**: System should be secure from malicious third-parties since some visitors may choose to save sensitive information such as credit card numbers on the database. System should also procure safe login features.
- **Maintainability**: System should be quickly fixable and easily navigable when there are problems. When the environment changes the system should be easy to keep problem-free. Simple, readable and understandable code is necessary to achieve maintainability.
- **Reliability**: System should have preferably zero, or realistically minimal, errors. Power loss or a server crash should not result in data loss. Since the database stores information concerning the animals' health and food, reliability is essential.
- **Performance**: RAM usage should be less than 3 GBs and data retrieval should be lower than 2 seconds for user satisfaction. Slow waiting times could potentially result in user loss because of annoyance and impatience.
- **Usability**: User-interface of the system should be easy to understand and should be easy to use since the target user-base, which is visitors of a zoo, includes a diverse

ensemble of people. A simple interface that enhances accessibility and intelligibility are significant.

- **Efficiency**: System should use as little time as possible to execute database operations like insertion and retrieval.

3.3 Constraints

The constraints, or the pseudo-requirements, of the project are as follows.

- React with HTML, JavaScript, CSS for the front-end.
- MySQL to implement the database.
- Java for the back-end.
- Material-UI and React-Bootstrap frameworks with React for the front-end.

4.0 Limitations

The limitations of the application system are as follows.

- All users should provide an unique email address and a password when signing up to the system.
- Personal data (such as credit card information, email address, salary, etc.) of all users will not be seen by anyone else using the system.
- Anyone visiting the website will be able to look at available events, however in order to make a payment to attend an event, they need to be logged in as a visitor.
- A visitor will be allowed to join a group tour event if the event's capacity is not filled yet.
- A visitor will be allowed to request a refund for a payment they made for a group tour event that hasn't happened yet, and not be allowed to attend the event if the refund is accepted.
- A visitor will not be able to request a refund for a donation made to a conservation fund.
- A visitor will only be allowed to send one complaint form for a group tour they attended.
- The employees will not be allowed to delete visitor comments or complaints on group tours.
- A coordinator can only edit events that they created, and only do this before the event has happened.

- A coordinator can cancel an event that they created if it hasn't happened yet. A group tour event can only be cancelled if all payments can be refunded to all payers.
- A coordinator can only create an educational program event if a veterinarian has accepted the invitation to take part in the event. The veterinarian of an educational program can only be changed if another veterinarian has accepted the invitation.
- A keeper can only regularize food or request treatment for an animal in a cage they are responsible for.
- A keeper can only assign non-expired food to cages while regularizing food.
- A keeper can schedule training for an animal in a cage they are responsible for.
- All cages must be assigned to a keeper at all times.
- A keeper can request treatment for the same animal again if/when the response from the inquired veterinarian is negative.
- A veterinarian can respond to invitations to events if the event hasn't happened yet.
- A veterinarian can respond to treatment requests for 2 days, otherwise it will be assumed that the response is negative.

5.0 Conceptual Design

The conceptual design of the database is visualized with the Entity-Relationship diagram in Figure 1, created using an online diagram-drawing tool [1].

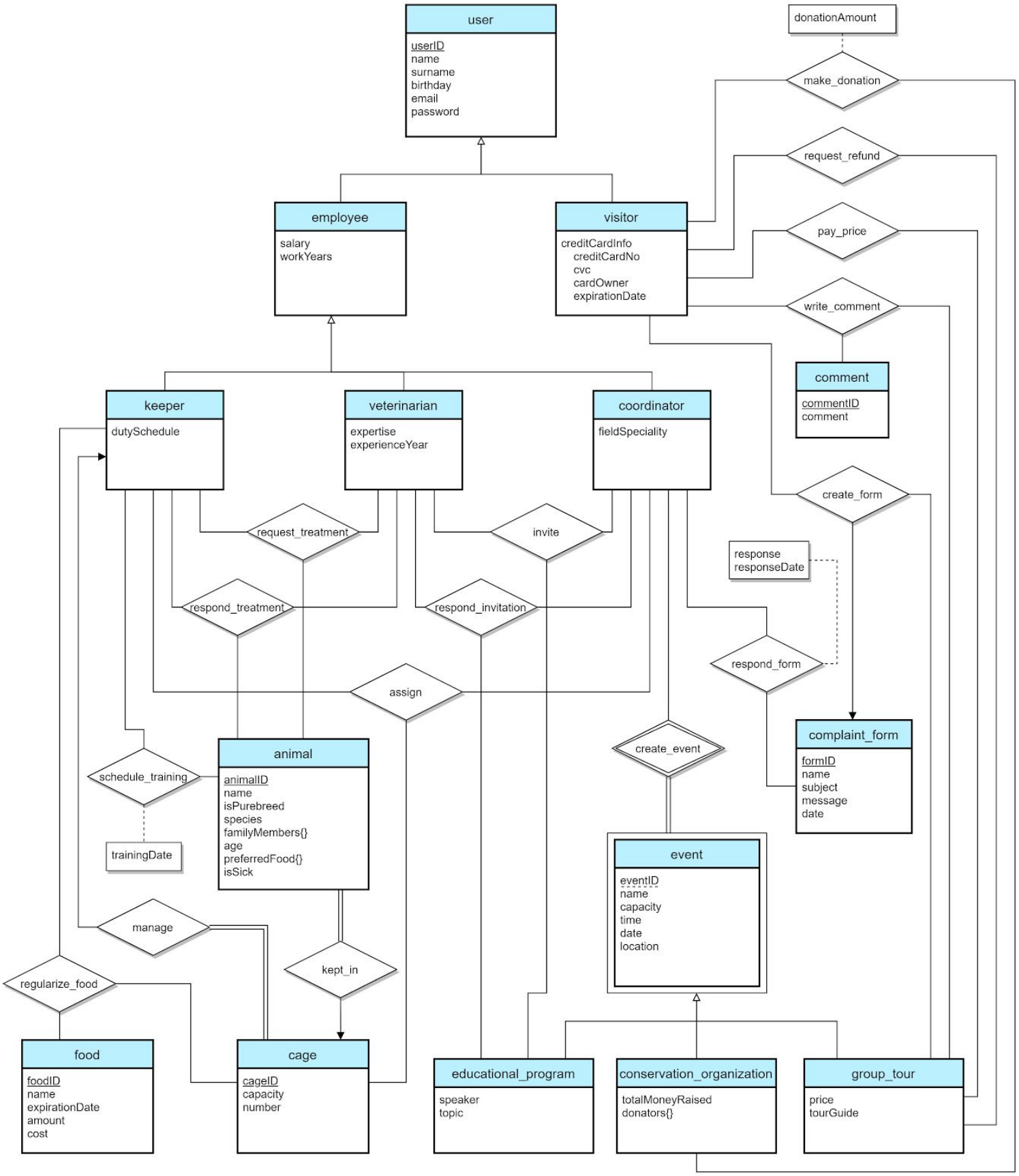

Figure 1: E/R diagram of the database.

6.0 Webpage

This section provides the webpage for the term project of our group, which as of now includes this proposal report.

Webpage: https://mehmetyaylacci.github.io/index.html

7.0 References

[1] “Security-first diagramming for teams.” https://www.diagrams.net/ (accessed 17 February 2021).