Bilkent University

Department of Computer Engineering

CS 353 Term Project

Travel Agency Data Management System

Project Specifications Document

Project Group Members:

Zübeyir Bodur	21702382
Abdul Razak Daher Khatib	21801340
Emine Ezgi Saygılı	21802871
Funda Tan	21801861

Instructor: Prof. Özgür Ulusoy

TA: Mustafa Can Çavdar

Table of Contents

Introduction **3**
Description 3
Usage of Databases 4

Requirements & Constraints **4**
Functional Requirements 4
Register and Login to System 4
Reserve a Place in a Tour 4
Reserve a Hotel Room 4
Review and Rate Tours and Guide 4
Update Reservation Details 5
Assign a Guide to Tours 5
Accept or Decline the Assigned Tours 5
Give Feedback About Tours 5
Apply Discount 5
Receive Discount 5
Preview Activities and Sightseeing Places included in Tours 5
Non-Functional Requirements 5
Privacy & Security 5
Performance 5
Usability 6
Scalability 6
Backup & Recovery 6
Robustness 6
Implementation Constraints 7
Limitations 7

Conceptual Design **8**

References **9**

1. Introduction

In this report, we will propose a travel agency system, used for organizing and managing tours. The system depends heavily on databases, therefore, we are planning to implement a relational database and integrate it into the system to be the core and the memory. We will describe the system, then explain the reason for the need for a database afterwards we will walk through the requirements, both functional and non-functional, and finally, we will discuss the constraints and limitations of the system.

1.1. Description

Our proposal is regarding a travel agency system. The system will provide the users the opportunity to plan their trips. The system enables the user to reserve a tour by herself or through an agent of the company, later they can edit and update the reservations. The reservations can be made for the hotel rooms or the tours. The tour will include several sightseeing places that are usually visited by tourists. Along with that, they will also have the ability to participate in activities like festivals and concerts. After the tour is over the user will be able to review the tour or the guide and rate them. The guide will also have the right to add feedback about the tour. To make things easier and beneficial for the agency we added multiple features to make the system more convenient and easier to use. For example, a discount can be applied to any reservation made by a customer, which can help the agency promote its tours. In addition to that, because employees can be responsible for reservations and, the others will be guides, we distinguish between these different employees for easier classification. In summary, we provide in our system what most travel agencies would need, from reservations to activities and sightseeing places, to discounts for the reservations.

1.2. Usage of Databases

Databases will be essential for our system since as mentioned above we will have a wide variety of components in our systems each with its attributes. For example in order for us to maintain data regarding each user, her reservations, and reviews we need to keep that in a data entity. Additionally, tours, in addition to its basic attributes, have a large amount of data that needs to be reserved, namely, reviews from customers, feedback from guides, sightseeing places that are visited, activities performed, reservations made by customers, and finally, the customers and guide assigned to the tour. So as we see, our system cannot be implemented without a database.

2. Requirements & Constraints

2.1. Functional Requirements

2.1.1. Register and Login to System

There are three types of users in the system. These types of users are Customer, Employee, and Guide. Users can register and log in to the system with the required information.

2.1.2. Reserve a Place in a Tour

Customers can reserve a tour in the city they choose either by themselves or with the help of an agent.

2.1.3. Reserve a Hotel Room

Customers can reserve a hotel room in the hotel they choose either by themselves or with the help of an agent.

2.1.4. Review and Rate Tours and Guide

Customers can review both their past tours and guides and rate them.

2.1.5. Update Reservation Details

Employees and customers can update the reservation details.

2.1.6. Assign a Guide to Tours

Agents can assign guides to tours. Tours have one guide and guides can be assigned to multiple tours.

2.1.7. Accept or Decline the Assigned Tours

Guides can accept or decline the tours assigned by agents.

2.1.8. Give Feedback About Tours

Guides can give feedback about the tours they guided.

2.1.9. Apply Discount

Agents can apply discounts to customers for their tours or hotel reservations.

2.1.10. Receive Discount

Reservations can receive discounts.

2.1.11. Preview Activities and Sightseeing Places included in Tours

All users will be able to preview activities and sightseeing places that will be a part of a tour.

2.2. Non-Functional Requirements

2.2.1. Privacy & Security

Data collection will comply with the GDPR so that sensitive data will not be shared with third parties and user data will not be collected without their consent, which will be stated in “Terms and Conditions” as users register. Moreover, Bearer authentication will be used in the web application to increase security. Furthermore, users’ financial information used for payments will be encrypted and secured to prevent any harm. Finally, all data will be encrypted.

2.2.2. Performance

Since the large number of users and tours will increase the space complexity of the queries, the framework that generates those queries up-to-date to ensure the latest performance improvements, and algorithms used to send and receive HTTP requests should be as efficient as possible. To be more precise, any operation should be completed in at most 2 seconds in the web application.

2.2.3. Usability

Since one of the rationales of the project is to make it easy to use by users, especially customers, without any assistance, the functionalities of the system should be intuitive and easily reachable to be applied and used. A new user should be comfortable using the travel agency management system in at most 2 minutes. Moreover, proper documentation, namely a help page that describes the functionality of the website, both for guides, agents, and customers, would be provided.

2.2.4. Scalability

The product should be scalable in terms of not only features but also capacity. Should there be new features to add to our system, it should be done by only changing necessary parts of the design. Moreover, the system should be able to be usable by at least 100,000 users.

2.2.5. Backup & Recovery

The database of the web application and the system files should be backed up on a regular basis so that in case of malicious attacks, data corruption, or any unexpected situations, the data stored should be recovered with minimum loss.

In this context, a full backup each week, and a backup of each day’s transaction should be created automatically. In addition to that, all payments will be saved immediately after being made.

2.2.6. Robustness

The web application should reject all bad requests that are sent to the API and the listener of the API should be resistant to a flood of requests. In case of bad requests, proper error pages should be opened without crashing the website.

Moreover, most of the wrong inputs entered by users should be validated before sending the request to the APIs, warning the user.

2.3. Implementation Constraints

- For collaboration and version control, a GitHub repository will be used.
- For the database, we will use MySQL.
- For the back-end framework, which generates queries for the database to execute as API receives requests, we will use Node.js.
- For the front-end framework, we will use Vue.js.
- Bootstrap will be used as a CSS library.
- Since we are using Node.js in the back-end framework, JavaScript programming language, as well as HTML/CSS will be used throughout the project.
- The documentation of this project will be accessible via this website [1]: https://cs353-travel-agency-system.github.io/.

2.4. Limitations

- Users of different types have different interfaces and interact with the system in different ways.
-
- A user will be able to store a single email address as an attribute. Therefore, users will not be able to associate multiple email addresses with their accounts in the system. Also, the user should provide a valid email address.
- If not specified, the username of the user will be the concatenation of his/her first name and last name. Therefore, a user's first name and last name cannot be null. Also, for security, a user's password cannot be null.
- Each phone number must have exactly 10 characters, which are integers ranging from 0 to 9. Also, the user should provide a valid phone number.
- Customers will be reviewing the guide and tour separately, and will not be rating individual qualities of them. For example, for the tour guide, customers will give an overall review for all qualities of the guide such as his/her knowledge of the tour area, ability to communicate effectively, strong empathy and understanding et cetera. Also, for the tour, customers will give an overall review for all qualities of the tour such as its price, duration, sightseeing places visited et cetera. This is done in order to simplify the reviewing process.

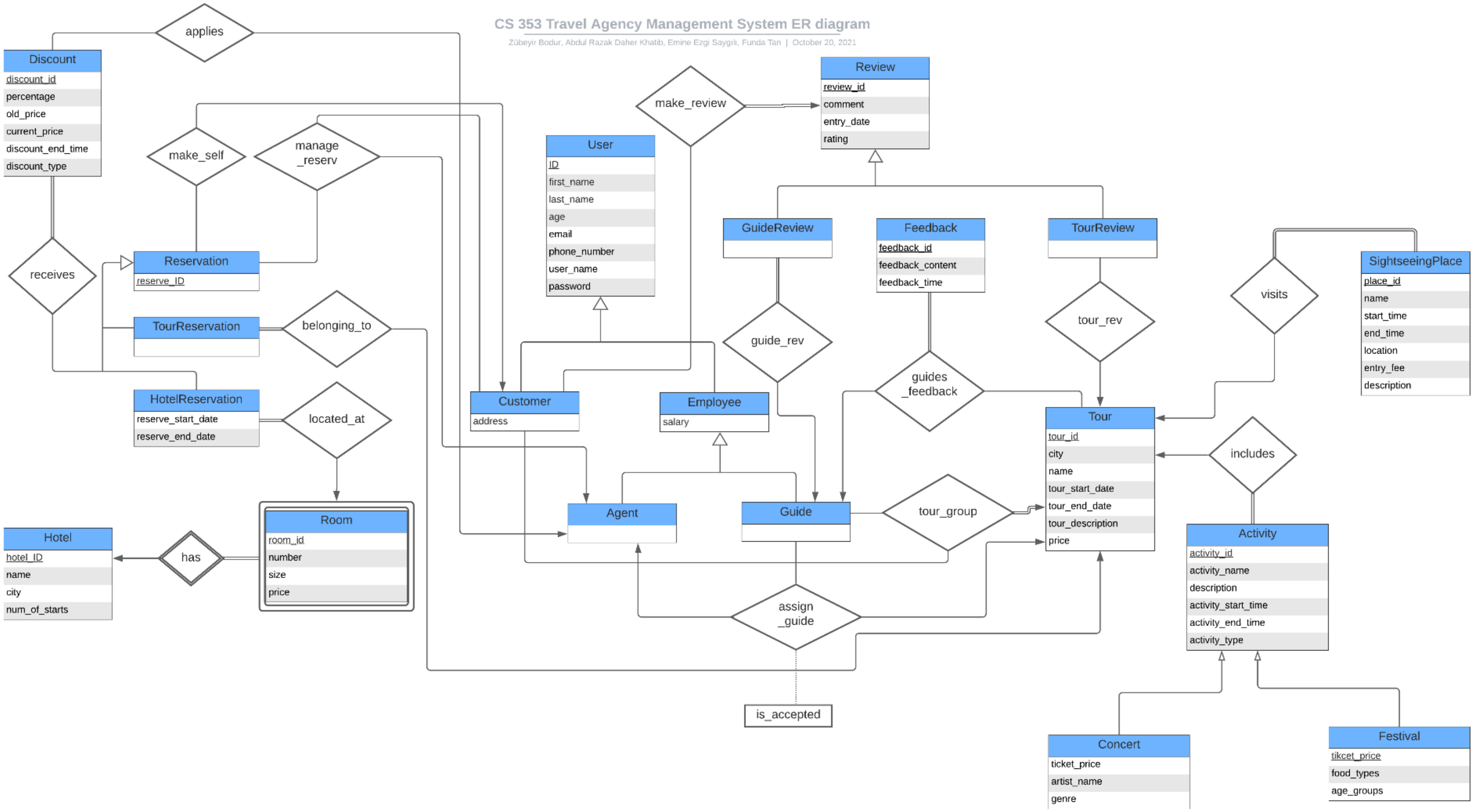
- A tour will only be within a single city, as a tour will be storing a single city as an attribute, so tours will not cover different cities.
- A tour will be guided by a single guide in the system. In real life, for a very large tour group sometimes more than one guide might be necessary for engaging with the tour group. However,it is assumed that every tour group will be small enough not to require more than one guide.
- A guide will be allow associated with a single language as an attribute. Therefore, users will not be able to associate multiple email addresses with their accounts in the system. Also, the user should provide a valid email address.

3. Conceptual Design

The figure on the next page shows the initial conceptual design of our project. We have drawn the diagram below by using an online tool [2].

CS 353 Travel Agency Management System ER diagram

Zübeyir Bodur, Abdul Razak Daher Khatib, Emine Ezgi Saygılı, Funda Tan | October 20, 2021



4. References

[1] “Travel Agency Management System | https://cs353-travel-agency-system.github.io,” *GitHub Pages,* [Online]. Available : https://cs353-travel-agency-system.github.io/. [Accessed Oct. 20, 2021].

[2] "Online Diagram Software & Visual Solution | Lucidchart", Lucidchart, 2021. [Online]. Available: https://www.lucidchart.com . [Accessed Oct. 20, 2021].