VILNIUS UNIVERSITY
FACULTY OF MATHEMATICS AND INFORMATICS
INSTITUTE OF COMPUTER SCIENCE
INFORMATION TECHNOLOGIES STUDY PROGRAM

Course work, Mid-term

A travel accommodation website

Done by:
Deividas Bendaravičius

Supervisor:
Linas Būtėnas

Vilnius
2023

Contents

1 Summary

The core objective of the system is to provide users with a comprehensive and easy platform to search, compare, and book accommodations that meet their preferences and budget, thereby enhancing their travel experience. In addition to just being a booking accommodation, payments will be done via Blockchain transactions, with the possibility to "stream" money. Hence, the project is a Web3 integration into the existing eco-system today.

2 Introduction

This section is reserved for the introduction of the document, and will be filled in later

3 Algorithms

Work in progress...

4 Web3

Work in progress...

5 Technologies

The technologies to create the system will be used as follows.

The MERN stack:

- Frontend: React.js + Typescript
- Backend: Node.js + Express.js + Typescript
- Database: MySQL
- Design: CSS + MUI and styled-components libraries

React.js (Javascript library) is one of the most widely used libraries for developing smooth and reactive web applications worldwide. React uses a declarative approach to building UI components, making it easier to manage the complexity of applications enabling easy scalability. React also offers high performance, as it uses a virtual DOM that minimizes the number of updates required to render changes to the UI.

Node.js allows to use of JavaScript in the backend and is also widely used. It provides an event-driven, non-blocking I/O model that makes it ideal for building scalable and high-performance applications. Node.js also has a vast ecosystem of packages and libraries that make it easier to develop and deploy web applications.

Express.js is a popular web application framework built on top of Node.js. It provides developers with a minimal and flexible set of tools for building web applications and APIs. Express.js also supports middleware, which allows adding functionality to the system, such as authentication, error handling, and logging easier.

MUI is going to be used to save time by using the provided components for building a reactive, safe, and bug-free UI, while also using best HCI practices. In addition to these components, styled-components will be used to take the design even further and adjust it according to the needs.

And MySQL is a popular, open-source database management system that enables efficient storage and retrieval of data.

To make it short, the MERN stack provides a powerful set of tools for building web applications that are scalable, efficient, and easy to maintain. React.js provides a robust and performant UI framework, Node.js and Express.js provide a flexible and scalable server-side architecture, and MySQL provides a reliable and scalable database solution. In addition, MERN main components use Javascript (to avoid errors and bugs, will be using Typescript instead) which I am familiar with.

6 Non-functional requirements

SECURITY The app will be using HTTPS requests between the backend and frontend to ensure data encryption. The system will also handle sensitive user data such as login credentials, to combat the risk of a leak, all data will be hashed and nothing is going to be stored in plain-text. The user accounts will be protected by requiring the user to provide credentials upon registration and login processes. Accessing routes to the backend will require JWT authorization tokens.

COMPATIBILITY The website should be compatible with any popular browser such as, but not limited to - Firefox, Chrome, Microsoft Edge, and Opera GX. The website should be accessed by a machine running minimal hardware requirements.

USABILITY The website will be applying best practices and other HCI principals to make the user experience easy and pleasant to use. The components that will be used to build the UI are going to be easily recognizable, because of the popular MUI library that many websites already use.

PERFORMANCE The website will not do any client-side calculations, and in addition to this, the website is going to be a single-page website using React.js that optimizes rendering based on the user's actions. This allows the application to be performant on any device.

RELIABILITY The system must be reliable and available for use at all times, with minimal downtime or system failures that would otherwise prevent users from creating, searching, and accessing their reservations.

7 System architecture

7.1 Use cases

The diagram below showcases the use cases of the system. There are three actors at play. User is a simple user that is able to use the majority of the functionalities that the system brings. User (HOST) is an advanced user, that inherits all of the simple users' use cases, in addition, has the ability to host his properties as available accommodation places. Finally, the User (ADMIN) is a user that can delete properties, ban users and view reports of the regular users.

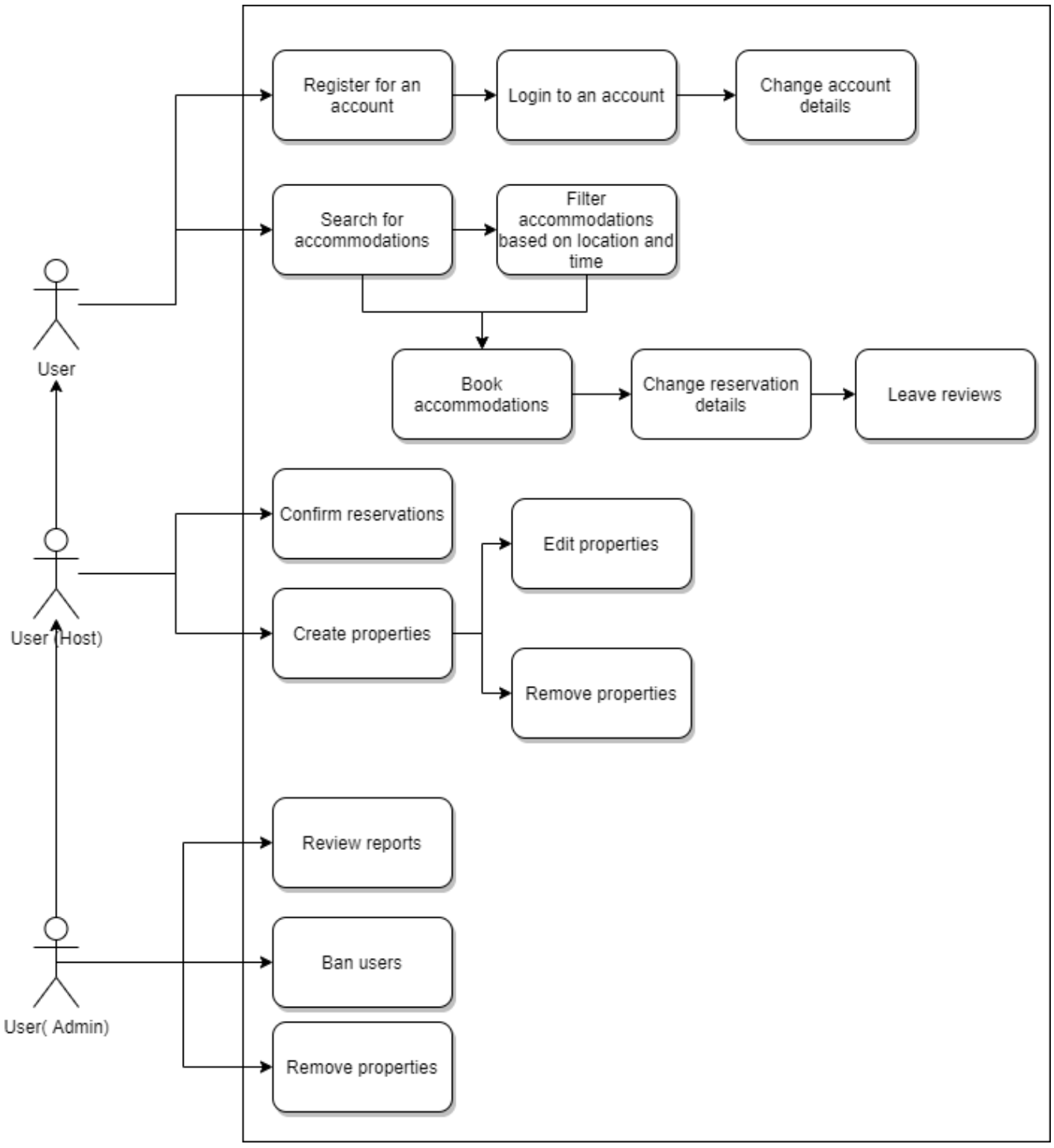

Figure 1. UML Use Case diagram

User:

- Register for an account - before using the website, a user has to create an account.
- Login to an account - in order to access listings, the user has to login on the website.
- Change account details - the user can change his account details, such as name, age, profile picture etc.
- Search for accommodations - the user is able to search for provided accommodations.
- Filter accommodations - the user is able to filter accommodations by location and availability, as well as price.
- Book accommodations - the user is able to book an accommodation.
- Change reservation details - the user is able to request a change of reservation details.
- Leave reviews - the user is able to leave reviews of the accommodation by the end of the stay.

User (Host). All of the above plus:

- Confirm reservations - the host user can confirm or decline reservations
- Create properties - the host user can create a property to list
- Edit properties - the host user can edit properties
- Remove properties - the user is able to delete owned properties from being listed.

7.2 Deployment

The deployment of the entire system is displayed in the diagram below. The system will consists of three parts

- Web browser running the React.js website
- Cloud platform hosting two servers. Backend server to handle communication between the database and the frontend, running Node.js. Database server storing data and listening to queries from the backend, running MySQL.

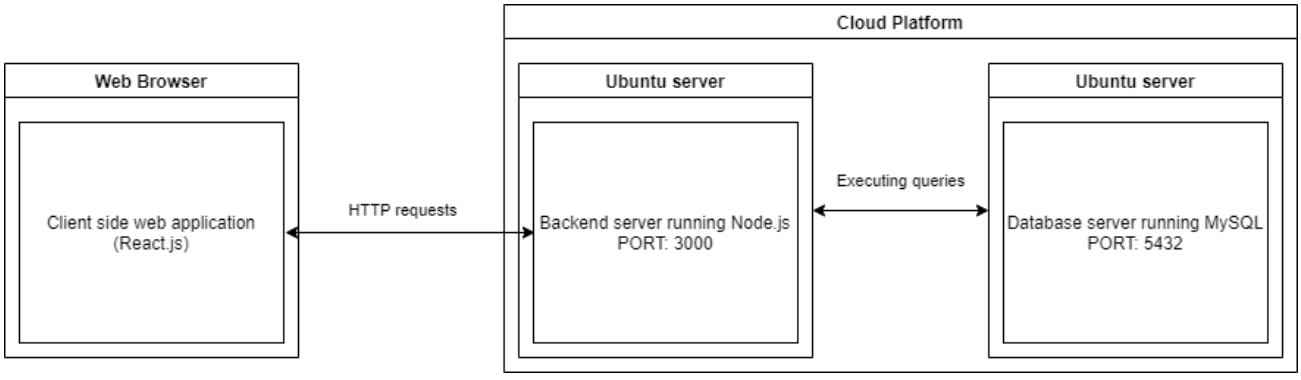

Figure 2. UML Deployment diagram

7.3 Relational Model

In order to better understand how data will be stored in the MySQL database server, the following relational model, as well as a short description is provided. Note, this is the initial design and will change as the system gets developed.

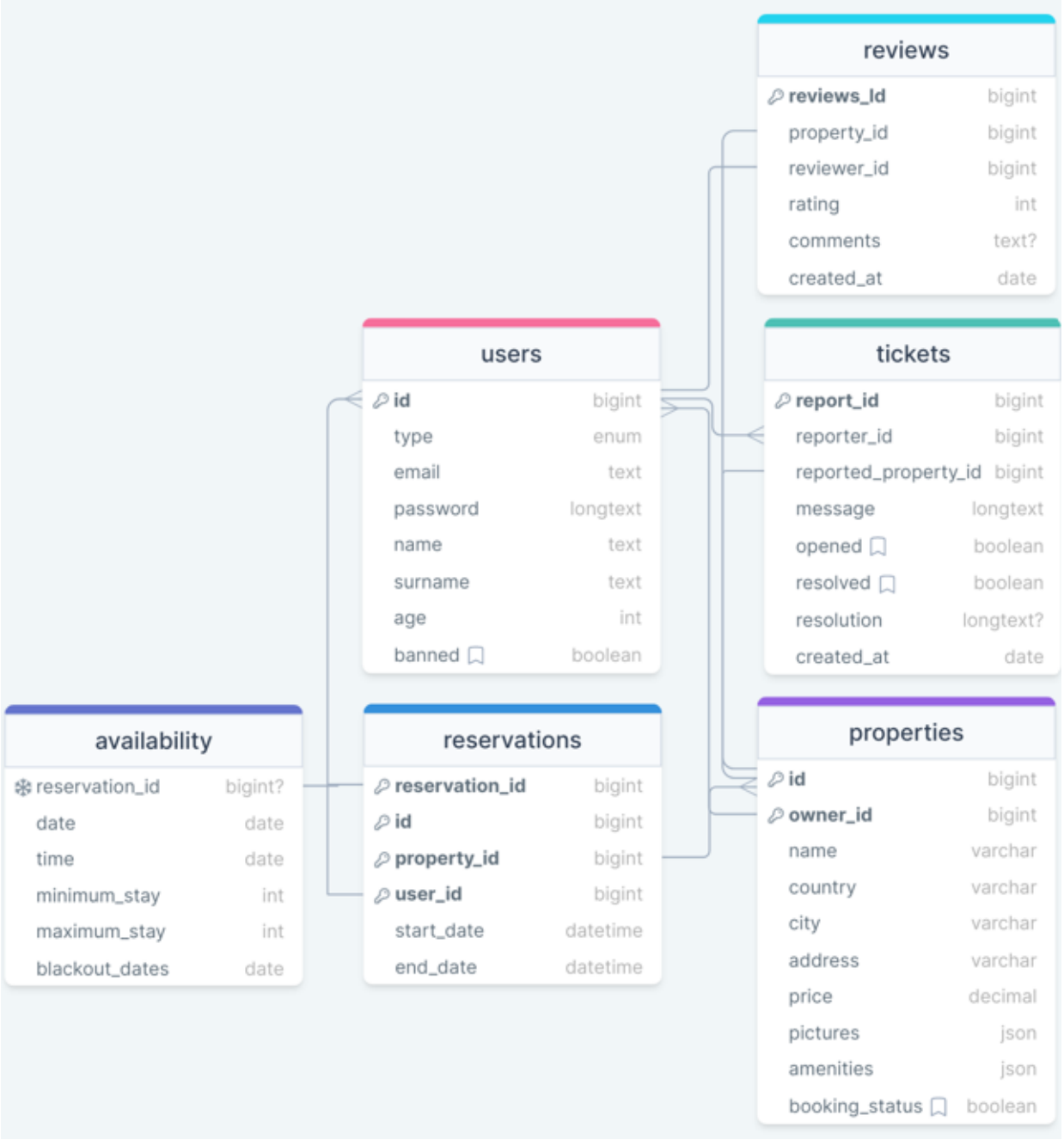

Figure 3. Database Relational model

Tables:

- users
  - id - a unique identifier
  - email - hashed email address of the user

  - type - a type of the user (regular/host)
  - name - the name of the user
  - surname - the surname of the user
  - age - the age of the user
  - password - hashed user password
  - banned - boolean type if the user was banned
- properties
  - id - property id
  - owner_id - property owner id
  - name - property name
  - location - location of the property
  - price - price of the property per night
  - pictures - property pictures
- reservations
  - reservation_id - reservation id
  - property_id - property id
  - user_id - user that reserved the property id
  - start_date - start date of reservation
  - end_date - end date of reservation
- availability
  - reservation_id - reservation id
  - date - beginning date
  - time - duration of stay
- tickets
  - report_id - unique id of a report
  - reported_id - id of the user that reported
  - reported_property_id - property that was reported
  - message - report message
  - opened - boolean value if the ticket was viewed by an admin
  - resolved - boolean value if the ticket was resolved
  - resolution - admin message about the resolution