Stay Cation: A Comprehensive Backend Platform for Vacation Property Management

Abstract

Stay Cation is a backend platform designed for comprehensive management of vacation properties. This poster presents the development and implementation of the system, including its conceptual model, software architecture, and computational techniques used to solve real-life problems. The platform includes features such as user registration, property management, bookings, comments, and billing, integrating the Google Maps API to enhance the user experience.

Methods and Materials

Project Description: Stay Cation centralizes the management of vacation properties, serving property owners, managers, and guests. The business model is based on a monthly subscription for owners and managers and a small percentage of each transaction.

Conceptual Model: UML diagrams were used to represent the system architecture, including deployment, activity, sequence, and state diagrams, along with CRC cards and class diagrams to define the backend architecture.

Design Patterns: The MVC pattern separates the application's concerns, ensuring modularity. The Singleton pattern ensures a single instance of the database connection manager.

Class Structure:

- **User Class:** Manages user registration, login, and profile management.
- **Property Class:** Represents properties, handles CRUD operations, and integrates with the Google Maps API.
- **PropertyAddon Class:** Manages additional features or services for properties.
- **PropertyType Class:** Categorizes properties into different types.
- **Booking Class:** Manages the booking process.
- **Comment Class:** Handles user comments on properties.
- **Bill Class:** Manages billing and invoice generation.
- **Card Class:** Represents payment cards and processes payments.

Tools Used:

- **MySQL and DBeaver:** For database management.
- **XAMPP:** For the server environment.
- **Python:** For backend development.
- **Google Maps API:** For property location visualization.

Introduction

The Stay Cation project aims to provide a complete solution for managing vacation properties.property visualization on maps, comment administration, and billing. Technologies used include MySQL for the database, Python for business logic, and DBeaver along with XAMPP for database management and visualization.

Methodology

The development of the Stay Cation platform followed a structured methodology to ensure the successful implementation of its features and functionalities. The key steps in the methodology are outlined below:

1. Requirement Analysis:

- Identify the needs of property owners, managers, and guests.
- Define user stories to capture the specific requirements and goals of each user type.

2. System Design:

- Develop UML diagrams, including deployment, activity, sequence, and state diagrams, to visualize the system architecture.
- Choose appropriate design patterns (MVC, Singleton, Factory) to ensure modularity, scalability, and maintainability.

3. Implementation:

- Set up the development environment using MySQL, DBeaver, XAMPP, and Python.
- Implement the core classes (User, Property, PropertyAddon, PropertyType, Booking, Comment, Bill, Card) based on the design specifications.

4. Testing:

- Conduct unit tests to verify the functionality of individual components.
- Perform integration tests to ensure functionality.

CRC Cards

CRC (Class-Responsibility-Collaborator) cards were used to design the core classes of the Stay Cation platform. Each card represents a class and outlines its responsibilities and collaborators.

User Class:

- **Responsibilities:** Register new users, authenticate users during login, manage user profiles.
- **Collaborators:** Booking Class, Comment Class, Card Class.

Property Class:

- **Responsibilities:** Add new properties, update property details, delete properties, generate property location URLs using the Google Maps API.
- **Collaborators:** User Class, PropertyAddon Class, PropertyType Class, Booking Class, Comment Class.

Booking Class:

- **Responsibilities:** Create new bookings, update booking details, cancel bookings.
- **Collaborators:** User Class, Property Class, Bill Class.

Comment Class:

- **Responsibilities:** Add new comments, display comments on properties, edit and delete comments.
- **Collaborators:** User Class, Property Class.

Bill Class:

- **Responsibilities:** Generate invoices for bookings, update billing records.
- **Collaborators:** Booking Class, User Class.

Card Class:

- **Responsibilities:** Add and update card information, delete card information, process payments for bookings.
- **Collaborators:** User Class, Booking Class.

User Stories

User stories were created to define the project scope and ensure that the system meets the needs of its users.

- **As an owner**, I want to register my properties so that they can be viewed and booked by guests.
- **As a manager**, I want to manage property bookings to ensure there are no date conflicts.
- **As a guest**, I want to search and book available properties for my vacation.
- **As an owner**, I want to see comments and ratings for my properties to improve the guest experience.
- **As a manager**, I want to generate automatic invoices for bookings to facilitate the payment process.
- **As a guest**, I want to add my payment card details to make booking payments easier.
- **As an owner**, I want to add additional features or services to my properties to attract more guests.

Conclusion

Stay Cation provides a robust solution for vacation property management, leveraging modern technologies and design patterns to address real-life problems. The platform's architecture ensures scalability and maintainability, making it a valuable tool for property owners, managers, and guests. Future work will focus on optimizing performance, enhancing security, and incorporating additional features based on user feedback.

Acknowledgments

The development of Stay Cation was supported by the Universidad Distrital Francisco José de Caldas. Special thanks to Eng. Carlos Andrés Sierra, M.Sc., for his guidance and support throughout the project.

References

- Donald E. Knuth, The TeX Book, Addison-Wesley Professional, 1986.
- Leslie Lamport, LaTeX: a Document Preparation System, Addison Wesley, Massachusetts, 2nd edition, 1994.

Josue Urrego Lopez

Josue Urrego Lopez

Universidad Distrital Francisco José de Caldas