**StayEasy**

A logo of a person in a circle

Description automatically generated

*“Find the Perfect Place”*

*Prepared by:*

Pajares, Josemar  
Te, Jushua Peter

Medallo, Christian Jay

StayEasy is a modern platform that redefines the way people find, book, and experience accommodations. Our system offers a user-friendly interface, advanced search features, secure booking processes, and a robust review system to facilitate seamless transactions and build trust within the community. Whether you're a host looking to monetize your property or a guest seeking a memorable stay, StayEasy provides a convenient and reliable solution. With features like instant booking, virtual tours, and personalized recommendations, we strive to deliver an exceptional experience for both hosts and guests. Join our community and discover the future of accommodation booking.

A diagram of a computer

Description automatically generated with medium confidence**ERD**

## **Functions**

1. **User Registration and Login:**

* Requirement: The system must allow users to register and log in securely.
* Rationale: This provides a way for users to create an account and access their information, bookings, and other features.
* MVP Implementation:
  + User registration form with basic fields (name, email, password, etc.)
  + Email verification for new users
  + Secure login functionality with password hashing and salting
  + Basic user profile management

1. **Review and Rating System:**

* Requirement: The system must allow guests to leave reviews and ratings for their hosts and properties.
* Rationale: This helps build trust and credibility within the platform, allowing guests to make informed decisions when booking a property.
* MVP Implementation:
  + Review form with rating system like 1-5
  + Review moderation system to ensure accuracy and fairness
  + Host response system to allow hosts to respond to reviews
  + Review and rating display on property detail pages

1. **Search and Booking:**

* Requirement: The system must allow guests to search for properties based on location, dates, and desired amenities.
* Rationale: This provides the core functionality for guests to find suitable accommodations and complete bookings.
* MVP Implementation:
  + Simple search function with basic filters (location, dates, price range, etc.)
  + Property detail pages with essential information (description, photos, amenities, etc.)
  + Booking calendar to display availability

1. **Wishlist:**

* Requirement: The system must allow guests to save properties to a wishlist for future reference.
* Rationale: This allows guests to keep track of properties they're interested in and easily access them later.
* MVP Implementation:
  + Wishlist button on property detail pages and options
  + Guest wishlist page to display saved properties
  + Option to remove properties from wishlist

1. **Payment Processing:**

* Requirement: The system securely processes payments for bookings, allowing guests to pay securely using various payment methods.
* Rationale: This enables smooth transaction processing and ensures financial security for both hosts and guests.
* MVP Implementation:
  + Integration with a reputable payment gateway like gcash
  + Support for multiple payment methods like credit card, bank ewallet, or cash

1. **Booking Confirmation and Management:**

* Requirement: The system should generate booking confirmations for guests and provide hosts with a dashboard to manage bookings.
* Rationale: This ensures clear communication and provides a centralized platform for managing bookings.
* MVP Implementation:
  + Automated booking confirmation emails to guests
  + Host dashboard with a list of upcoming and past bookings
  + Basic booking management features (e.g., update booking status, view guest information)

I. Identification (10 items, 2 point each)

1. Who developed Python and when was it first released?

2. What is the philosophy of Python often summarized in?

3. What Python keyword is used to define a function?

4. How does Python handle indentation in code?

5. What type of loop repeats actions for a specific sequence or range?

6. What is an example of a derived attribute in an ERD?

7. Name the syntax used in Python for single-line comments.

8. Which Python function converts a string to an integer?

9. Define an entity in the context of an ERD.

10. What does OOP stand for in Python?

Il. Enumeration (5 items, 2 points each)

1. List 3 features that make Python popular.

2. Enumerate the 3 phases of data modeling.

3. Name 3 types of control structures in Python.

4. List the 3 components of an ERD.

5. Provide 4 examples of standard Python libraries.

III. Reasoning (5 items, 3 points each)

1. Explain why Python's dynamic typing is both an advantage and a potential risk.

2. Describe the importance of ERDs in database design.

3. Why is Python's readability important for program maintenance?

4. Discuss the significance of using indentation in Python syntax.

5. How do inheritance and polymorphism improve code reusability in Python OOP?

IV. Coding (5 items, 5 points each)

1. Write a Python function that checks if a number is even or odd.

2. Create a Python class 'Car' with attributes 'make", 'model', and year'.