PROPERTY FINDER

FUNCTIONAL REQUIRMENTS

1 - CREATE ACCOUNT.

- **Description:** User shall be able to create account on the system.
- **Inputs:**
  - Username.
  - User Type
  - User email.
  - Password.
  - phone number
  - User Type.
- **Source of inputs:** User (Traveler, Host).
- **Pre-conditions:**
  - Valid user email (Must Contains @domain.com).
  - Username must be unique.
  - Password must meet the minimum password requirements. (at least 4 chars).
  - Phone number must be exactly 11 digits.
- **Post-conditions:**
  - Account added to database.
  - User redirected to login page.
- **Output:**
  - A notification *"Account Created Successfully, Welcome username"*.

2 - LOGIN

- **Description:** User shall be able to login into his account on the system.
- **Inputs**:
  - Email.
  - Password
- **Source of inputs:** User (Traveler, Host).
- **Pre-conditions:**
  - The user must enter his *Email* and *password.*
  - The user has a valid *username* and *password*.
  - The account exists in the database.
- **Post-conditions:**
  - User Login Successfully.
  - The user status is *"logged in"*.
  - The user is redirected to the home page.
- **Output:** A notification *"Logged in Successfully, Welcome! (username)"*.

FUNCTIONAL REQUIRMENTS

3 - SEARCH FOR PROPERTY

- **Description:** Traveler shall be able to search for appropriate properties according to some filter options.
- **Inputs:**
  - num of rooms.
  - location.
  - price.
  - property type.
  - Contract type.
- **Source of inputs:** Traveler.
- **Pre-conditions:**
  - Traveler must be logged in.
  - The user must provide the search criteria mentioned above.
- **Post-conditions:** Searching for properties that meet user input and property statutes is *"Available"*.
- **Output:** Display list of available properties.

4 - MAKE RESERVATION

- **Description:** The traveler shall be able to rent/buy any property and will pay using his *E-wallet*.
- **Inputs:**
  - Select Property.
  - Traveler information.
  - Start date & end date. (*In case of rental*)
  - Pay
- **Source of inputs:** Traveler.
- **Pre-conditions:** Traveler must be logged in.
- **Post-conditions:** Reservation sent to the host.
- **Output:** Notification *"Your request is sent to the host!"*.

FUNCTIONAL REQUIRMENTS

5 - COMPLETE THE RESERVATION PROCESS

- **Description:** Perform Renting/ buying process.
- **Inputs:** Make a Reservation.
- **Source of inputs:** Traveler.
- **Pre-conditions:**
  - Host acceptance.
- **Post-conditions:**
  - Update database. Property status became *"In Market"*.
  - Generate contract.
- **Output:**
  - A notification *“Process Done Successfully“*.
  - Process info.

6 - PAY MONEY

- **Description:** Traveler shall be able to pay money to host.
- **Inputs:** Host acceptance.
- **Source of inputs:** Traveler.
- **Pre-conditions:**
  - Credit card information is correct.
  - Money in *credit card* >= *Cost.*
- **Post-condition:** Transform Money from traveler wallet to host wallet.
- **Output:** Notification of transformation of money.

FUNCTIONAL REQUIRMENTS

7 - ADD PROPRIETY

- **Description:** Host can add list of properties providing specification for each one.
- **Inputs:**
  - num of rooms.
  - place.
  - renting or buying.
  - cost.
  - location.
- **Source of inputs:** Host.
- **Pre-conditions:**
  - Host must be logged in.
  - Host must provide all specifications of property.
- **Post-conditions:**
  - The property added in database with statues available.
  - Host redirected to his home page.
- **Output:** Notification *“Property added Successfully“*.

8 - ACCEPT / RECEJCT

- **Description:** Host *accept* or *reject* traveler reservation.
- **Inputs:** Reservation.
- **Source of inputs:** Traveler.
- **Pre-conditions:**
  - Host is logged in.
  - Traveler made reservation.
- **Post-condition:** Host accept or reject user reservation.
- **Output:** Notification of acceptance.

9 - GENERATE CONTRACT

- **Description:** System should generate contract to user.
- **Inputs:** Complete the Reservation Process.
- **Source of inputs:** Complete Reservation Process.
- **Pre-conditions:** Complete Process info.
- **Post-condition:** Send contract to user.
- **Output:** Generated Contract.

FUNCTIONAL REQUIRMENTS

10 - RETURN PROPERTY

- **Description:** change property statues to available.
- **Inputs:** End Date.
- **Source of inputs:** Traveler.
- **Pre-conditions:** Property type is in market and date exceed current date.
- **Post-conditions:** Change type to available.
- **Output:** Property statues became available.

NON-FUNCTIONAL REQUIRMENTS

1 - SECURITY:

- **Description:** The system must use encryption techniques on password to apply protection during storage and transmission.
- **Measure:** Vulnerability Assessment, Penetration Testing.
- **Type:** Product requirement.

2 - PERFORMANCE:

- **Description:** The system should respond quickly to user requests.
- **Measure:** less than 10 seconds per operation.
- **Type:** Product requirement.

3 - PROGRAMMING LANGUAGES:

- **Description:** The system will be developed using.
  - C# programming language.
  - .NET framework.
  - Oracle database.
- **Type:** Organizational requirement.

4 - USABILITY:

- **Description:** The system should be easy to use and navigate with clear and intuitive interfaces that allow users to find and filter properties quickly and easily.
- **Measure:** Users should ask for guide max 2 times per session.
- **Type:** Product requirement.

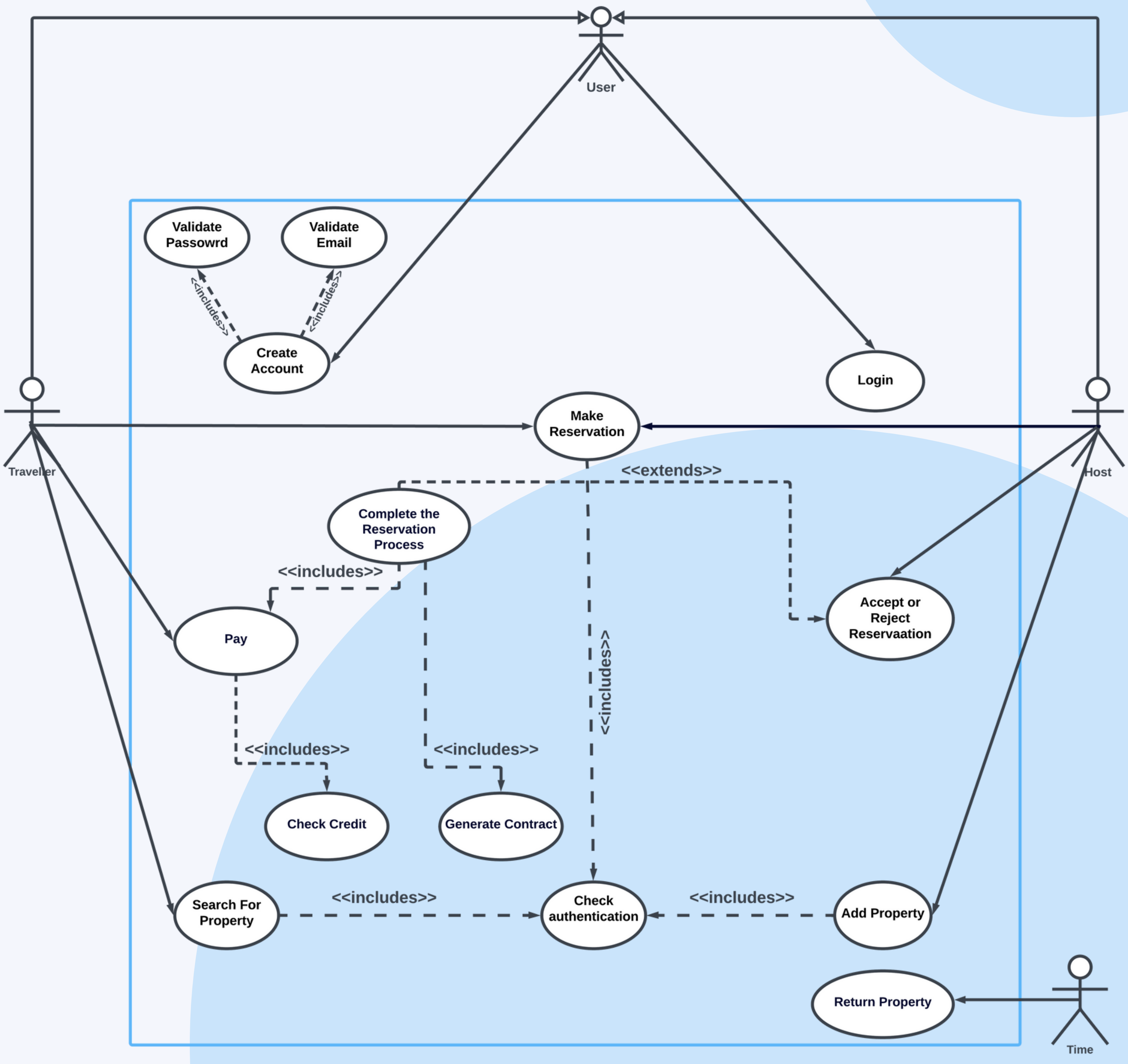
5 - AVAILABILITY:

- **Description:** The system should be available to users 24/7.
- **Measure:** Max number of fails 2 per year.
- **Type:** Product requirement.

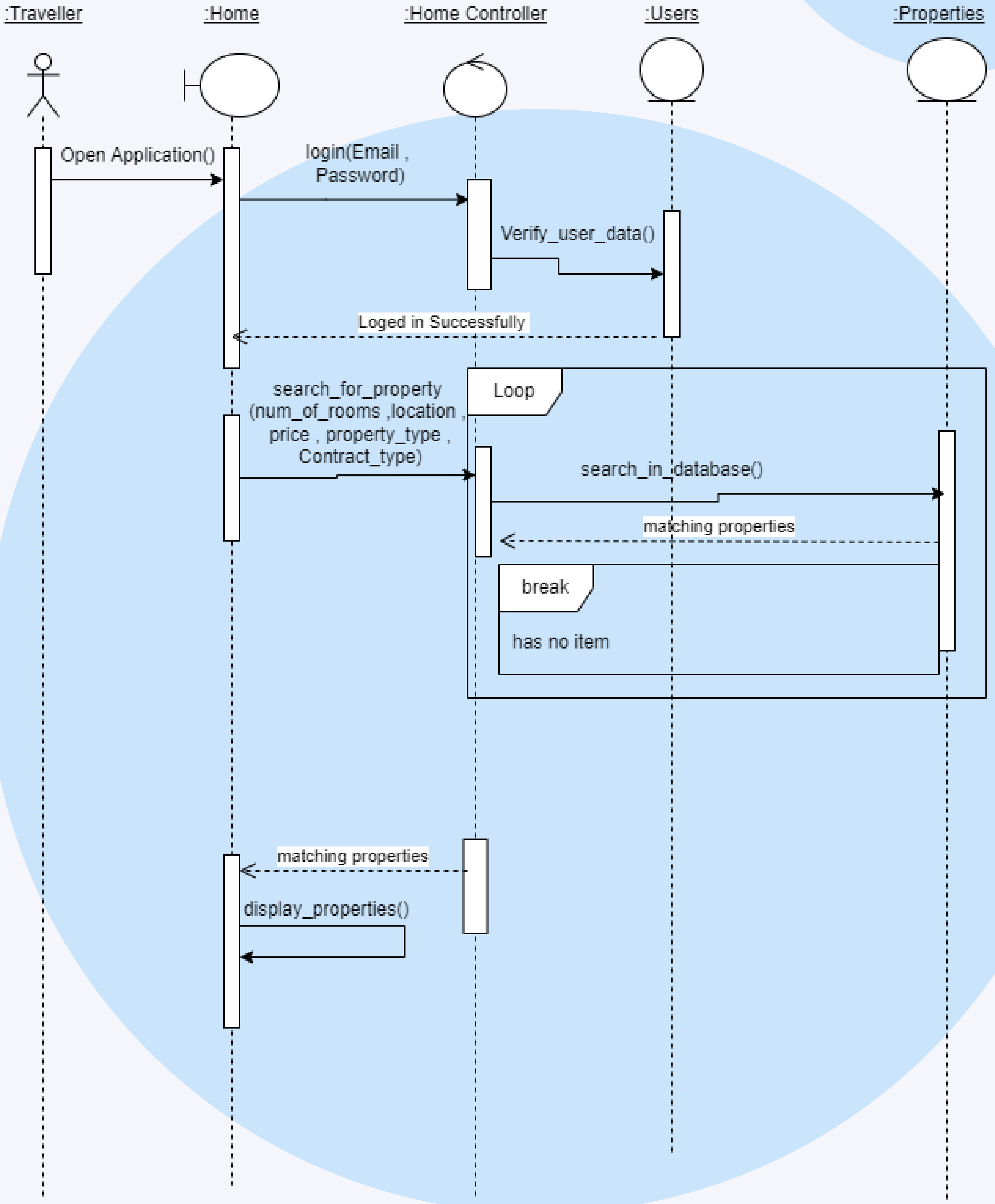
6 - SCALABILITY:

- **Description:** The system must be able to store up to 1000 property per Host.
- **Measure:** System should be stable when connected users are 1000.
- **Type:** Product requirement.

USE CASE DIAGRAM:



SEQUANCE DIAGRAM SEARCH FOR PROPERTY


:Traveller
:Home
:Home Controller
:Users
:Properties
Open Application()
login(Email , Password)
Verify_user_data()
Loged in Successfully
Loop
search_for_property (num_of_rooms ,location , price , property_type , Contract_type)
search_in_database()
matching properties
break
has no item
matching properties
display_properties()