

# **Don't Wreck My House Project Plan**

*Each Layer except the UI will include testing and will be tested alongside. I'm expecting testing to take the same as each section labeled here or up to 1.5x that time.*

## **Models:**

- Guest
  - Fields:
    - Guest\_id [int]
    - First\_name [String]
    - Last\_name [String]
    - Email [String]
    - phone [String]
    - State [String]
  - Constructor
  - Methods
    - Getters
    - Setters
    - Equals
    - hashCode
- Host
  - Fields
    - Id [String]
    - Last\_name [String]
    - Email [String]
    - Phone [String]
    - Address [String]
    - City [String]
    - State [String]
    - Postal\_code [Int]
    - Standard rate [Big Decimal]
    - Weekend rate [Big Decimal]
  - Constructor
  - Methods
    - Getters
    - Setters
- Reservation
  - Fields
    - id [int]
    - start\_date [LocalDate]
    - end\_date [LocalDate]

- guest\_id [int]
- guest [Guest]
- host [Host]
- total [Big Decimal]
- Constructor
- Methods
  - Getters
  - Setters

**Data Access Layer + Models** (*This whole section should take ~ 4 hours to complete, meaning testing should also take ~ 4 hours additionally*):

- Data Exception Class: Extends Exception

```

- public DataException(String message) {
-     super(message);
- }
-
- public DataException(Throwable cause) {
-     super(cause);
- }
-
- public DataException(String message, Throwable cause) {
-     super(message, cause);
- }

```

- GuestRepository Interface
  - READ
    - List<String> findAll
    - List<String> findByEmail
    - List<String> findByLastName
- GuestFileRepository Class
  - List<String> findAll
  - List<String> findByEmail
  - List<String> findByLastName
- HostRepository Interface
  - READ
    - List<String> findAll
    - List<String> findByEmail
    - List<String> findByLastName
- HostFileRepository Class
  - READ
    - List<String> findAll
    - List<String> findByEmail

- `List<String> findByLastName`
- ReservationRepository Interface
  - CREATE
    - `Reservation addReservation (Reservation reservation)`
      - Calculates the total for the reservation using LocalDate based on Standard rate (Sunday - Thursday) and Weekend Rate (Friday - Saturday)
      - Adds a reservation as long as the dates don't overlap with other requirements
        - The start date must come before the end date.
        - The start date must be in the future.
        - Guest, host, and start and end dates are required.
        - The guest and host must already exist in the "database".
    - Guests and hosts cannot be created
  - READ
    - `List<Host> findByEmail ( String Email )`
    - `List<Host> findById ( int id )`
    - `List<Host> findByLastName ( String lastName )`
  - UPDATE
    - `Reservation updateReservation ( Reservation reservation )`
      - Looks for a reservation using one of the above READ methods
      - Start and end date can be edited. No other data can be edited.
        - This means the total will need to be recalculated (Can make the calculation its own method to separate concerns)
  - DELETE
    - `Reservation deleteReservation (Reservation reservation)`
      - Looks for a reservation using one of the above READ methods
        - Will only display future reservations
- ReservationFileRepository Class
  - CREATE
    - `Reservation addReservation (Reservation reservation)`
      - Calculates the total for the reservation using LocalDate based on Standard rate (Sunday - Thursday) and Weekend Rate (Friday - Saturday)
      - Adds a reservation as long as the dates don't overlap with other requirements
        - The start date must come before the end date.

- The start date must be in the future.
- Guest, host, and start and end dates are required.
- The guest and host must already exist in the "database".

Guests and hosts cannot be created

- READ

- List<Host> findByEmail ( String Email )
- List<Host> findByGuestId ( int id )
- List<Host> findByLastName ( String lastName )

- UPDATE

- Reservation updateReservation ( Reservation reservation )
  - Looks for a reservation using one of the above READ methods
  - Start and end date can be edited. No other data can be edited.
    - This means the total will need to be recalculated (Can make the calculation its own method to separate concerns)

- DELETE

- Reservation deleteReservation (Reservation reservation)
  - Looks for a reservation using one of the above READ methods
    - Will only display future reservations

***Domain/Service Layer:***

- ReservationResult (This + GuestService + HostService will take about 4 hours, meaning testing will take an additional 4 hours)

```

- private final ArrayList<Reservation> reservations = new
  ArrayList<>();
- private Reservation reservation;

- public Panel getReservation(){
-     return reservation;
- }

- public void setReservation(Reservation reservation){
-     this.reservation = reservation;
- }

- public boolean isSuccess() {
-     return reservations.size() == 0;
- }

```

```

public List<Reservation> getReservation() {
    return reservations;
}

public void addMessage(Reservation reservation) {
    reservations.add(reservation);
}

```

- GuestService
  - Include all view methods used
- HostService
  - Include all view methods used
- ReservationService (Expecting this Service Class will be broken up into two sections (View + Add + Testing & Update + Delete + Testing) and each take about 4 hours with their corresponding testing)
  - Include all view methods used
  - Add + validations
  - Update + validations checked
  - Delete + validations checked

#### ***UI Layer:***

- MainMenuOption (ENUM)
- ConsoleIO (If I wanna make a separate class or integrate directly into View)
- View: UI Validation (Expecting the view layer will be broken up into three sections (I/O methods, View + Add, and Update + Delete) will each take about 4 hours with their corresponding testing)
- Controller: (Will take another 4 hours to complete)
  - Menu
  - View Reservations
  - Add Reservations
  - Update Reservations
  - Delete Reservations

#### ***App.Java (1-2 hours)***

- Dependency injections + setting up the application to run with instance method call (run).

***FINAL DEBUGGING + RESEARCH*** to solve any issues (~8 Hours)