+FlightUI() + main(String[] args): void <u>+ printWelcomingPage(): void</u> + printLogin(): void <u>+ printActionsPage(RegisteredUser currentUser): void</u> + printFlightSearchNotLoggedIn(): void + printFourMatchesNotLoggedIn(Strin departLocation, String destination, ArrayList<String> <u>airline): void</u> FlightsList + printSingleflightOnlyNotLoggedIn(String departLocation, String destination, ArrayList<String> <u>airline): void</u> -flightLoader: FlightDatabaseLoader + printConnectingFlightsOnlyNotLoggedIn(String departLocation, String destination, -flight: ArrayList<Flight> <u>ArrayList<String> airline): void</u> -flightList: FlightsList + printFlightSearch(RegisteredUser currentUser): void + printFourMatchesPage(RegisteredUser currentUser, String departLocation, String destination, ArrayList<String> airline): void + getInstance(): FlightsList + printBookFlight(RegisteredUser currentUser, ArrayList<Flight> flightSearch): void + getFlightByUUID(UUID id): Flight + printSingleFlightsPage(RegisteredUser currentUser, String departLocation, String + getAllFlights(): ArrayList<Flight> destination, ArrayList<String> airline): void + getFourMatches(String departLocation, String destination, ArrayList<String> + printSingleFlights(String departLocation, String destination, ArrayList<String> airline): airline): String <u>ArrayList<Flight></u> + getFirstMatch(String departLocation, String destination, ArrayList<String> + printConnectingFlightsPage(RegisteredUser currentUser, String departLocation, String airline): ArrayList<Flight> destination, ArrayList<String> airline): void + checkValidityOfAirline(String airline): bool + printAddSeat(FlightBooking flightBooking, int num): void + checkValidityOfClass(String prefClass): bool + printConnectingFlights(String departLocation, String destination, ArrayList<String> airline): + getSecondMatch(String departLocation, String destination, ArrayList<String> <u>ArrayList<ArrayList<Flight>></u> airline): ArrayList<Flight> <u>+ printAddFriendsToFlightBooking(RegisteredUser currentUser, FlightBooking flightBooking):</u> + getThirdMatch(String departLocation, String destination, ArrayList<String> airline): ArrayList<Flight> + checkRatingValidity(double rating): boolean + getFourthMatch(String departLocation, String destination, ArrayList<String> + addRatingPref(): double airline): ArrayList<Flight> + printHotelSearch(RegisteredUser currentUser): void -checkAirline(ArrayList<String> airline): bool  $\underline{\ +\ printFourMatchesByLocation(RegisteredUser\ currentUser,\ String\ location):\ void}$ + getSingles(String departLocation, String destination, ArrayList<String> + printAllMatches(RegisteredUser currentUser, String location, ArrayList<Amenities> amenities, airline): ArrayList<Flight> ArrayList<Accessibility> accessibility, String roomType, int numOfBeds, double rating): void + getConnectingFlights(String departLocation, String destination, + printBookHotel(RegistereedUser currentUser, ArrayList<Hotel> hotelSearch, int response): ArrayList<String> airline): ArrayList<Flight> <u>void</u> + getThreeConnectingFlights(ArrayList<ArrayList<Flight>> returnList, String + printHotelSearchNotLoggedIn(): void departLocation, String destination, ArrayList<String> airline): void + printFourMatchesByLocation(String location): void + getTwoConnectingFlights(ArrayList<ArrayList<Flight>> returnList, String + printAllMatches(String location, ArrayList<Amenities> amenities, ArrayList<Accessibility> departLocation, String destination, ArrayList<String> airline): void accessibility, String roomType, int numOfBeds, double rating): void getFlightsByAirline(ArrayList<Flight> filterList, ArrayList<String airline): void <u>+ addAmenitiesPref(): ArrayList<Amenities></u> + getFlightByDestination(ArrayList<String> airline, String destination): + addAccessibilityPref(): ArrayList<Accessibility> ArrayList<Flight> + printQuit(): void + addTravelers(RegisteredUser currentUser, String first, String last, + printCreateAccount(): void FlightBookign flightBooking): void + createUsername(): String + createFlightBooking(RegisteredUser currentUser, ArrayList<Flight> + createPassword(): String flightSearch): FlightBooking HotelsList HotelDatabaseLoader <u>+ printSetPrefQuestions(RegisteredUser currentUser): void</u> + getFlightByDepartLocation(ArrayList<String> airline, String departLocation): <u>+ printSetPreferences(RegisteredUser currentUser): void</u> ArrayList<Flight> -hotels: ArrayList<Hotel> # HOTEL FILE NAME: String + addAirlinePref(): ArrayList<String> + getFlightsGen(ArrayList<Flight> returnList, String departLocation, String -returnList: ArrayList<Hotel> # HOTEL FILE ID: String + addClassPref(): ArrayList<String> destination): void <u>-hotelList: HotelsList</u> # HOTEL FILE CITY: String + printBookingNotLogin(): void + getAvailableSeats(Flight): void # HOTEL FILE CHECK IN DATE: String + printItineraryPage(RegisteredUser currentUser): void addSeatToFlightBooking(Flight flight, String seatNum, FlightBooking # HOTEL FILE CHECK OUT DATE: String + printltinerary(RegisteredUser currentUser): void <u>-getInstance(): HotelsList</u> flightBooking): Seat # HOTEL FILE CHECK IN TIME: String + printCancelFlight(RegisteredUser currentUser): void -clearSearch(): void + cancelFlight(UUID): void # HOTEL FILE CHECK OUT TIME: String + printCancelHotel(RegisteredUser currentUser): void getAllHotels(): ArrayList<Hotel> + saveFlights(): void # HOTEL NAME: String + printFriendsList(RegisteredUser currentUser): void getHotelByUUID(UUID id): Hotel # HOTEL ROOM TYPE: String + printAddNonUserFriend(RegisteredUser currentUser): void addTraveler(RegisteredUser currentUser, String first, String last, HotelBooking # HOTEL NUMBER OF BEDS: String + printAddUserFriend(RegisteredUser currentUser): void hotelBooking): void # HOTEL FILE Price Per Night: String + printRemoveFriend(RegisteredUser currentUser): void addRoom(HotelBooking hotelBooking, Hotel hotel, int roomNum): void - getbool(String data): boolean getSearch(String location, ArrayList<Amenities> amenities, Room ArrayList<Accessibility< accessibility, String roomType, int numOfBeds, double - numberOfBeds: String rating): ArrayList<Hotel> <u>+ getHotel(): ArrayList<Hotel></u> roomTypes: String getFourMatchesByLocation(String location): ArrayList<Hotel> + parseDate(String date): Date FlightApp -checkInDay: Date getHotelByLocation(String location): void - checkOutDay: Date checkValidityOfAccessibility(String accessibility): boolean currentUser: User checkInTime: String checkValidityOfAmenity(String amenity): boolean - checkOutTime: String getHotelByRating(double rating): void getHotel(String keyword): Hotel HotelDatabaseWriter getHotelByAmenities(ArrayList<Amenities> amenities): void + Room(String room\_type, String - getAllHotels():ArrayList<Hotel> getHotelByAccessibility(ArrayList<Accessibility> accessibility): void number of beds, Date getFlight(String keyword): Flight # Hotel FILE: String getHotelByRoomType(String roomType): void check in date, Date - getAllFlights(): ArrayList<Flight> # ID: String getHotelByNumberOfBeds(int numOfBeds): void - login(): User # Hotel CITY: String check out date, String printRoomByDateAndTime(Hotel hotel, Date checkInDate, String checkinTime, check\_in\_time, String + addUser(...): User # Hotel\_Name: String UserDatabaseWriter Date checkOutDate, String checkOutTime): void check\_out\_time) # Rooms: String + cancelHotel(UUID hotelID): void # Rooms room type: String # User FILE NAME: String + saveHotels(): void getRoomType(): String # Rooms number of beds: # User\_FILE\_Id: String # User\_FILE\_User\_Id: String getRoomCheckInDate(): Date getRoomCheckOutDate(): Date # User FILE Password: String # Roomes check in date: String getRoomCheckOutTime(): # User FILE First Name: String
# User FILE Last Name: String UserDatabaseLoader # Roomes check out date: <<enumeration>> UserList **HotelRoomTypes** String # User FILE Last Name: String
# User FILE Address: String
# User FILE City: String
# User FILE State: String
# User FILE Zip: String
# User FILE Date Of Birthday: String
# User FILE Email Address: String getRoomNumberOfBeds(): # Roomes check in time: String # User FILE NAME: String String # Roomes\_check\_out\_time: SUITE # User FILE Id; String -userDatabaseLoader: UserDatabaseLoader STANDARD # User\_FILE\_User\_Id: String String -userDatabaseWriter: UserDatabaseWriter # User FILE Password: String ROOM ONLY # Roomes Price Per Night: -users:ArrayList<RegisteredUser> CONNECTING # User\_FILE\_First\_Name: String <u>String</u> -userList: UserList MINIMALIST # User\_FILE\_Last\_Name: String # User\_FILE\_Address: String DELUXE # User\_FILE\_Phone\_Number: String + saveHotels(): void # User\_FILE\_Disability: String STUDIO # User\_FILE\_City: String + getHotelsJSON(Hotel): # User\_FILE\_Visa: String # User FILE State: String JSONObject # User FILE Occupation: String
# User FILE Discount: String # User FILE Zip; String + getRoomsJSON(Room): -HotelRoomTypes(String -UserList() # User\_FILE\_Date\_Of\_Birthday: <u>JSONObject</u> textForm) +getInstance(): UserList() # Friend Date Of Birthday: String
# Friend Date Of Birthday: String +toString():String + getUserByUUID(UUID id): RegisteredUser # User\_FILE\_Email\_Address: + printItinerary(RegisteredUser currentUser): void + writeItineraryToFile(RegisteredUser user, String title): void # User\_FILE\_Phone\_Number: + checkValidityOfUsername(String username): boolean <<enumeration>> getFriendsList(RegisteredUser currentUser): void **Amenities** # Friend Discount: String # User\_FILE\_Disability: String + addNonUserFriend(RegisteredUser currentUser, String Hotel # User\_FILE\_Visa: String # Friend\_Flights: String firstName, String lastName, Date dob, String discount): void **POOL** # User\_FILE\_Occuplation: String # Friend\_Flight\_id: String + bookHotel(RegisteredUser currentUser, HotelBooking GYM -name: String # User\_FILE\_Discount: String # Friend\_Flight\_Seat: String hotel): void **TOILETRIES** -location: String # User\_Friends: String # Friend\_Hotels: String + getBookedFlights(RegisteredUser currentUser): COFFEEKIT rooms: ArrayList<Room> # Friend\_Hotel\_id: String ArrayList<FlightBooking> WIFI # Friend Hotel Check In Day: String + getBookedHotels(RegisteredUser currentUser): **PARKING** - amenities: # Friend Hotel Check Out Day: String ArrayList<HotelBooking> **SMOKE** <u>- getUser():</u> ArrayList<Amenities> + cancelFlight(RegisteredUser currentUser, FlightBooking MEAL <u> ArrayList<RegisteredUser></u> -accessibility: User flight): void + saveUsers(): void ArrayList<Accessibility> <u>+ parseDate(String date): Date</u> + cancelHotel(RegisteredUser currentUser, HotelBooking + getUsersJSON(RegisteredUser user): -Amenities(String textForm) flightList FlightsList hotel): void <u>JSONObject</u> +toString(): String -hotelList HotelsList + Hotel(String name, String + bookFlight(RegisteredUser currentUser, FlightBooking + getFriendsJSON(Friend friend): checkInDay, String flight): void <u>JSONObject</u> checkOutDay, String location, + setPreferences(RegisteredUser currentUser, Preferences + getFlightsJSON(Flight flight): getFourMatches(String int numberOfBeds, String pref): void <u>JSONObject</u> departLocation, String roomType) + removeFriend(RegisteredUser currentUser, String first, + getHotelsJSON(Hotel hotel): destination, +addAmenities(): void String last): void <u>JSONObject</u> removeHotell(): void +removeAmenities(): void + addUserFriend(RegisteredUser currentUser, String +addAccessibility(): void username): void Profile <<enumeration>> +removeAccessibility(): void getUserByUsername(String username): RegisteredUser Accessibility + addUser(RegisteredUser user): void firstName: String SERVICE ANIMAL + removeUser(UUID userID): void - lastName: String HEARING ACCESSABILITY KIT + getFriendsList(): ArrayList<Friend> - address: String VISUAL ACCESSABILITY KIT getUser(String username, String password): - city: String VISUAL\_FIRE\_SMOKE\_DETECTOR RegisteredUser - state: String GRAB\_BARS\_IN\_BATHROOM + getAllUsers(): ArrayList<RegisteredUser> zip: String LEVER\_LOCK\_SYSTEM dateOfBirth: String RegisteredUser ALTERNATIVE\_TEXT\_PLACARD emailAddress: String SHOWER\_SEATS phoneNumber: String - mobileNumber: String -id: UUID -disability: String Accessibility(String textForm) userProfile: Profile - visa: String +toString():String username: String - occupation: String -password: String discounts: ArrayList<String> -pref: Preferences flighBookings: ArrayList<FlightBooking> hotelBookings: ArrayList<HotelBooking> HotelBooking userList: UserList +Profile(String firstName, String lastName, String address, String city, String state, String zip, String # travelers: ArrayList<Profile> dateOfBirth, String emailAddress, String rooms: ArrayList<Room> phoneNumber, String mobileNumber, String + RegisteredUser(UUID id, hotel: Hotel disability, String visa, String occupation) ArrayList<Friend> friendList, Profile +getFirst(): String + HotelBooking(ArrayList<Profile> userpProfile, String username, String +getLast(): String travelers, ArrayList<Room> password) +getVisa(): String rooms, Hotel hotel) + RegisteredUser(Profile userProfile, +getOccupation(): String - addTravelers(Profile): void String username, String password) FlightBooking +accessDiscounts(): ArrayList<String> - addRoom(Room): void + getUsername(): String + getHotel(): Hotel + getPassword(): String # travelers: ArrayList<Profile> Preferences - toString(): String + getFriends(): ArrayList<Friend> - flight: Flight -airline: ArrayList<String> getFriendByFirstAndLast(String first, -flightClass: ArrayList<String> +FlightBooking(ArrayList<Profile> String last): Friend travelers, ArrayList<Seat> seats, + setPreferences(Preferences): void +Preferences(ArrayList<String> airline, ArrayList<Flight> flights) + getPreferences(): Preferences ArrayList<String> flightClass) + addTraveler(Profile): void + bookFlight(FlightBooking): void +getAirline(): ArrayList<String> + cancelFlight(FlightBooking): void + addSeat(Seat): void +getFlightClass(): ArrayList<String> + bookHotel(HotelBooking): void + getFlight(): ArrayList<Flight> + toString(): String + cancelHotel(HotelBooking): void + addFriend(Friend): void + addNonUserFriend(String, String, Date, Friend + addUserFriend(String username): -id: UUID Friend -firstName: String + removeFriend(String, String): void -lastName: String + accessItinerary(): void -dateOfBirthday: Date + getHotel(): ArrayList<HotelBooking> -discount: String + getFlight(): ArrayList<FlightBooking> + getID(): UUID + Friend(UUID Friend\_ID, String Friend\_FirstName, String Friend\_LastName, Date do, String Friend\_Discount, ArrayList<Flight> friend\_flight, ArrayList<Hotel> new\_Hotel) + Friend(UUID uuid2, String first, String last, Date dob, String discount) + getProfile(): Profile + getFirstName(): String Flight Seat getLastName(): String getDateofBirthday(): Date id: UUID occupied: bool getDiscount(): String departLocation: String - seatNumber: String toString(): String destination: String + getID(): UUID departDate: String - getFlights(): ArrayList<Flight> Seat(bool occupied, String departTime: String getHotels(): ArrayList<Hotel> seatNumber) arrivalTime: String + isOccupied():bool flightSeats: ArrayList<Seat> <<enumeration>> +setOccupied(bool occupied): void airline: String AirlineCompany +getSeatNumber():String AMERICAN\_AIRLINE + Flight (UUID id, String departLocation, String DELTA destination, String departDate, String departTime, SOUTHWEST AIRLINES String arrivalTime, ArrayList<Seat> flightSeats, String UNITED\_AIRLINES airline) + getID(): UUID getDepartLocation(): String -AirlineCompany(String textForm) getDestination(): String + toString: String getDepartDate(): String - getDepartTime(): String getArrivalTime(): String + getFlightSeats(): ArrayList<Seat> getAirline(): String toString(): String <<enumeration>> **FlightClass ECONOMY** PREMIUM\_ECONOMY FIRST\_CLASS BUSINESS -FlightClass(String textForm) +toString():String FlightDatabase Loader FlightDatabaseWriter # User FILE NAME: String # User FILE\_Id: String # FLIGHT\_FILE: String # User\_FILE\_User\_Id: String # ID: String # User FILE Password: String # Departs\_FROM: String <u> # User\_FILE\_First\_Name: String</u> # DESTINATION: String # User FILE Last Name: String # DEPARTURE\_DATE: String # User FILE Address: String #DEPARTURE\_TIME: String # User\_FILE\_City: String # ARRIVAL\_TIME: String # User FILE State: String #SEATS: String # User\_FILE\_Zip: String #AIRLINE: String # User\_FILE\_Date\_Of\_Birthday: String #SEAT\_NAME: String # User FILE Email Address: String #SEAT\_AVAILABLE: String # User FILE Phone Number: String # User FILE Disability: String + saveFlights(): void # User FILE Visa: String + getFlightsJSON(Flight flight): # User\_FILE\_Occupation: String <u>JSONObject</u> # User FILE Discount: String + getSeatsJSON(Seat seat): # User Friends: String <u>JSONObject</u> + getUser(): ArrayList<RegisteredUser>

+ parseDate(String date): Date