with initialization of data type

Boundary Object

MainInterface
+ login(): void
+ register(): void

<<calls>>

Boundary Object

<<calls>>

Boundary Object

RegisterPage
- email: String
- password: String
- confirmPassword: String
- errorMessage: String
+ authenticate(): Boolean
+ checkPasswordMatch(): Boolean
+ displayErrorMessage(): void

Boundary Object

LoginPage
- accountType: String
+ handleForgetPassword(): void

<<calls>>

Boundary Object

ForgetPassword
- email: String
- newPassword: String
- confirmPassword: String
- errorMessage: String
+ validateEmail(): Boolean
+ validateNewPassword(): String
+ checkPasswordMatch(): Boolean
+ displayErrorMessage(): void

<<use>>

<<use>>

Control Object

<<use>>

AuthenticationMgr
+validateCredentials(): Boolean
+registerUser(): void
+resetPassword(): void

The User class serves as an abstract base defining common attributes and behaviors, while Guest and AuthenticatedUser inherit from it, with Guest having limited access and AuthenticatedUser having full privileges, ensuring a clear distinction between anonymous and logged-in users.

Entity Object

Boundary Object

User
- name: String
+ redirectToMain(): void
+ getDetails(): void

Boundary Object

SettingsPage
+updateSettings(): void
+getSettings(): Settings

Control Object

SettingsMgr
+updateUserPreferences(): void
+getPreferences(): Settings

Entity Object

Settings
-theme: String
-notificationsEnabled: Boolean
-locationPermissions: Boolean
-gpsTrackingEnabled: Boolean
+applyChanges(): void
+restoreDefaults(): void

Boundary Object

CaptureParkingLocation
+ capturePhoto(): void
+ saveNote(): void
+ viewSavedLocation(): void

<<use>>

Control Object

LocationMgr
+saveLocation(): void
+retrieveLocation(): Location
+updateLocation(): void

<<use>>

Entity Object

ParkingLocation
-id: String
-location: String
-photo: Image
-note: String
-timestamp: DateTime

is a

Entity Object

AutheticatedUser
- email: String
- password: String
- confirmPassword: String
- errorMessage: String
+ validateCredentials(): Boolean
+ checkPasswordMatch(): Boolean
+ displayErrorMessage(): void
+ getDetails(): void

Boundary Object

HistoryPage
- results: <List> Result
+ viewHistory(): List
+ createResult: Result

Control Object

HistoryMgr
+saveParking(): void
+retrieveHistory(): List
+deleteHistory(): void

Entity Object

Result
-location: string
-date: DateTime
-duration: Int
-cost: Float

Boundary Object

UserMapView
- userType: String
- currentLocation: String
- selectedParkingSpot: String
+ displayMaps(): void
+ fetchParkingSpots(): void
+ selectParkingSpot(spotID): void

use

use

use

Control Object

CarparkMgr
- carparkList
- currentLocation
- selectedCarpark
- carparkDetails
- availabilityStatus
+ fetchAvailableCarparks()
+ getCarparkDetails()
+ checkAvailability()
+ getCheapestCarpark()
+ getNearestCarpark()
+ sortCarparkBy(criteria)
+ getShelteredCarpark()
+ getEVCarpark()

Control Object

NavigationMgr
- currentLocation: String
- destination: String
- routeDetails: String
- estimatedTimeOfArrival: DateTime
- distance : Float
+ calculateRoute(): Float
+ getDirections(): void
+ calculateETA(): DateTime
+ calculateDistance(): Float
+ displayRoute(): void
+ startNavigation(): void

Control Object

SearchMgr
- searchQuery: String
- filters: String
- searchResults: String
+ performSearch(): void
+ applyFilters(filters): void
+ sortResults(criteria): void
+ searchByLocation(location): void
+ searchbyPriceRange(min,max): void
+ searchByAvailability(): void
+ searchByEVcharger(): void
+ clearSearch(): void

NEAWeatherAPI

OneMap API