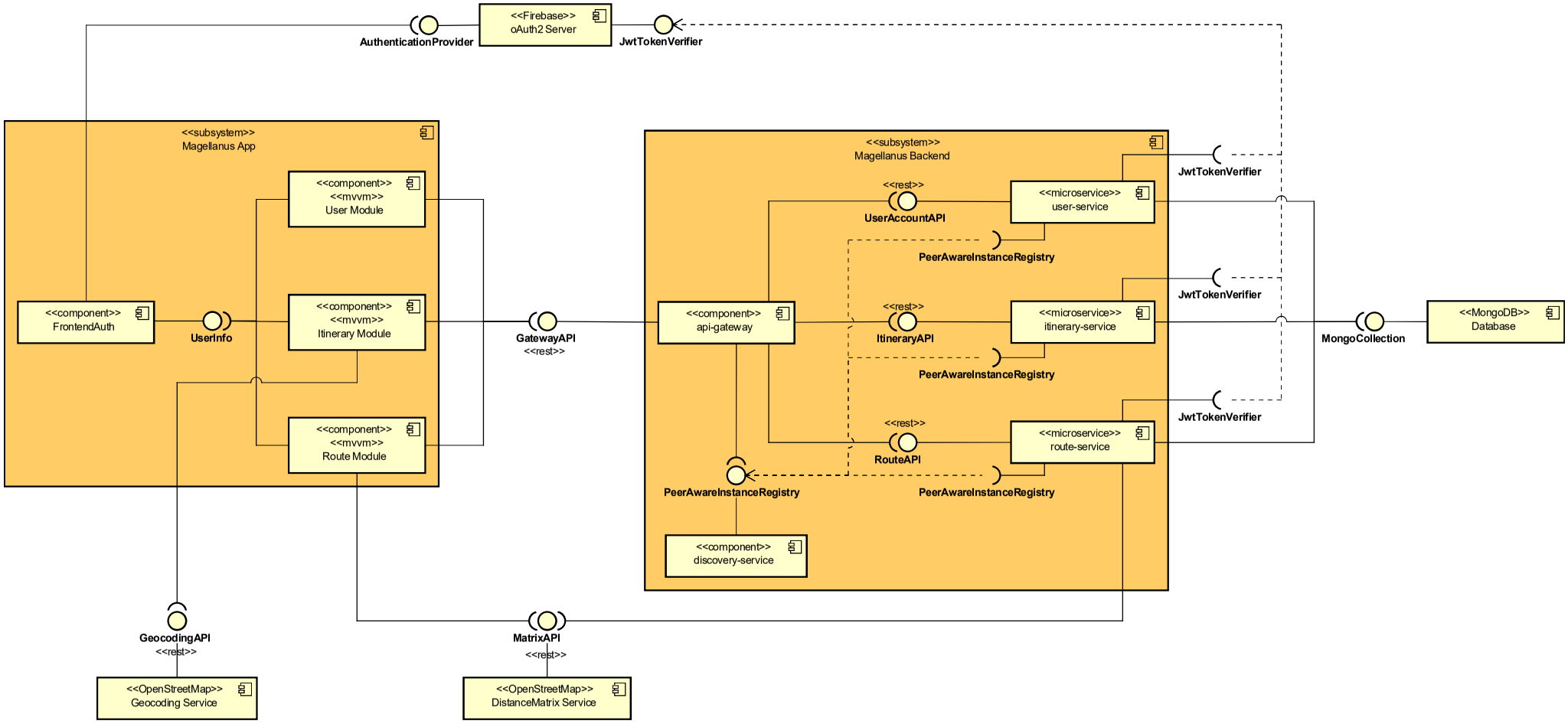
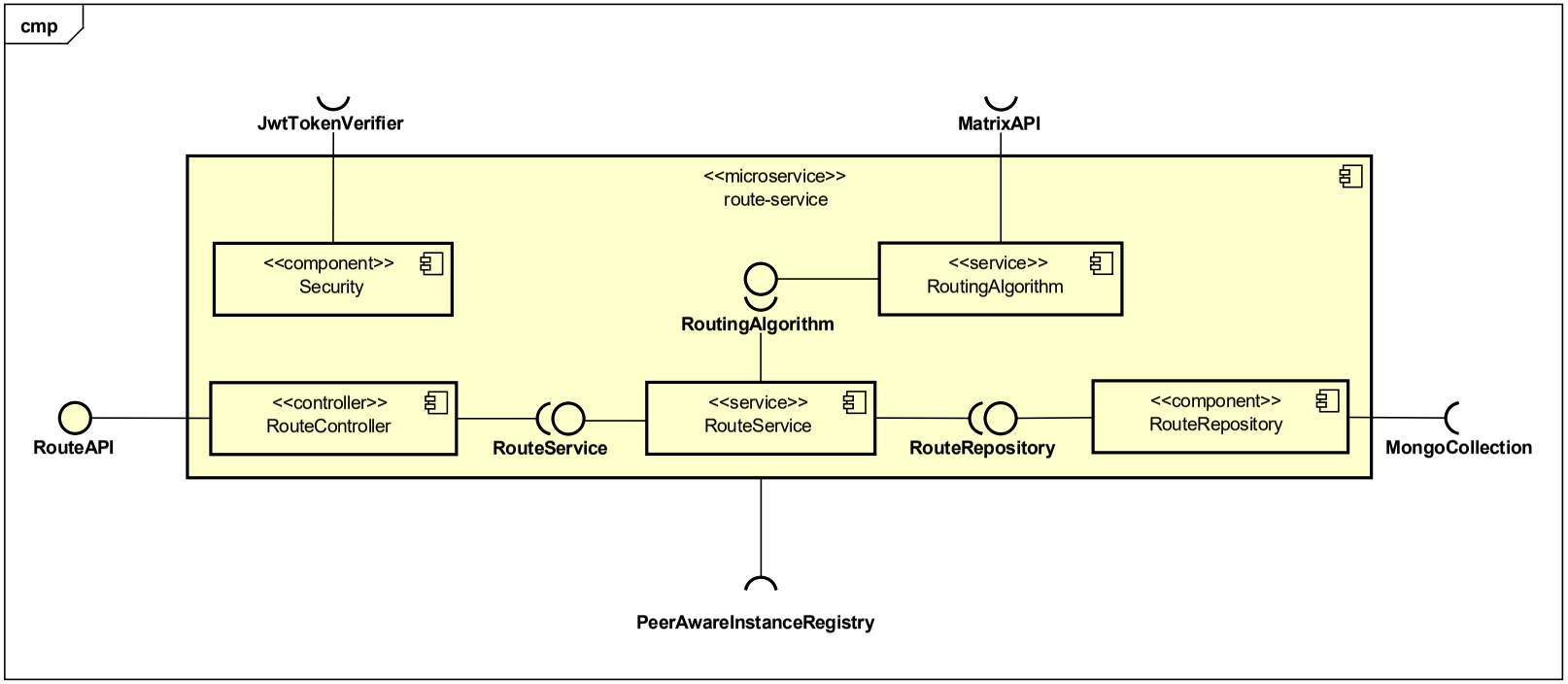
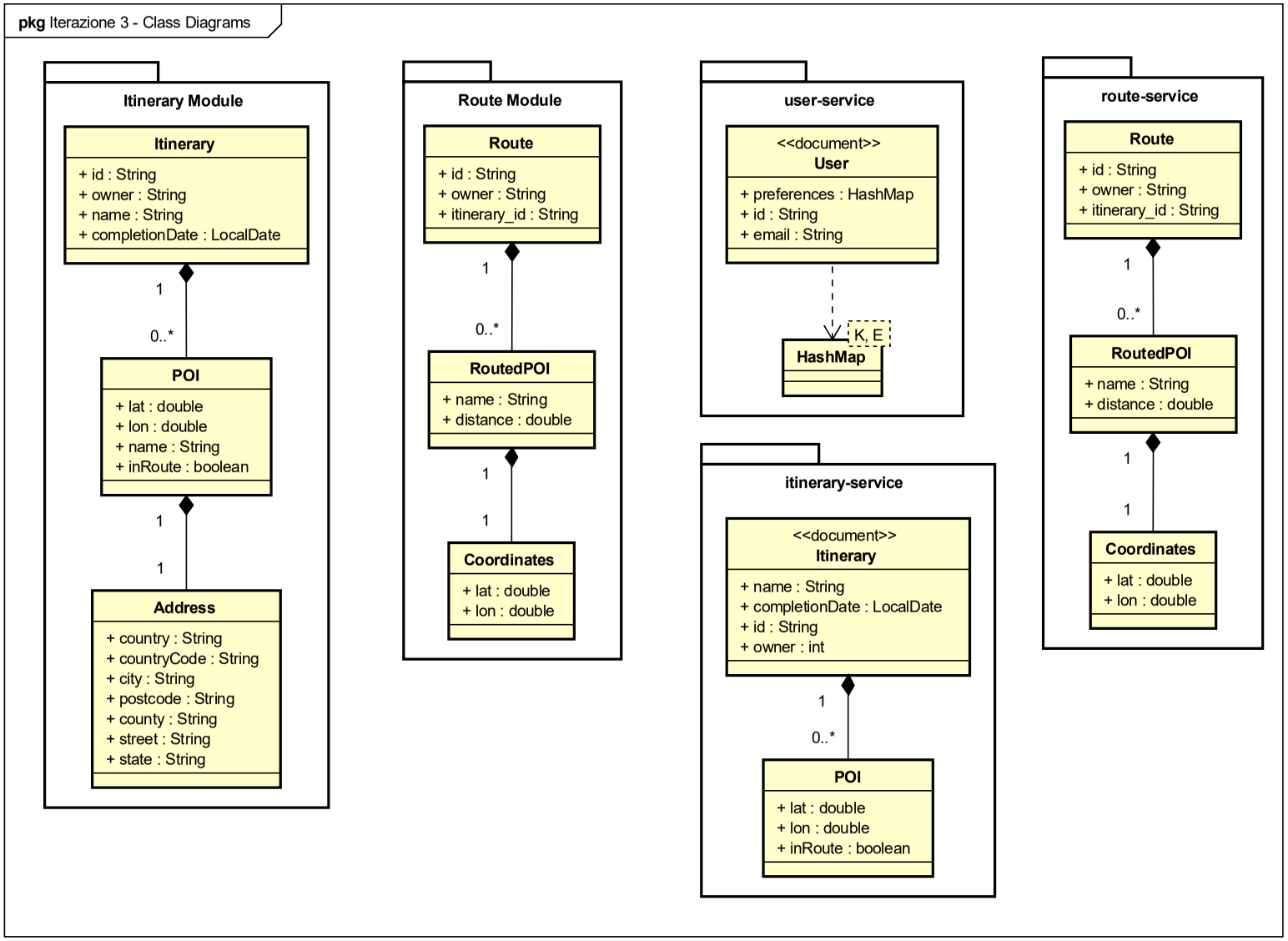
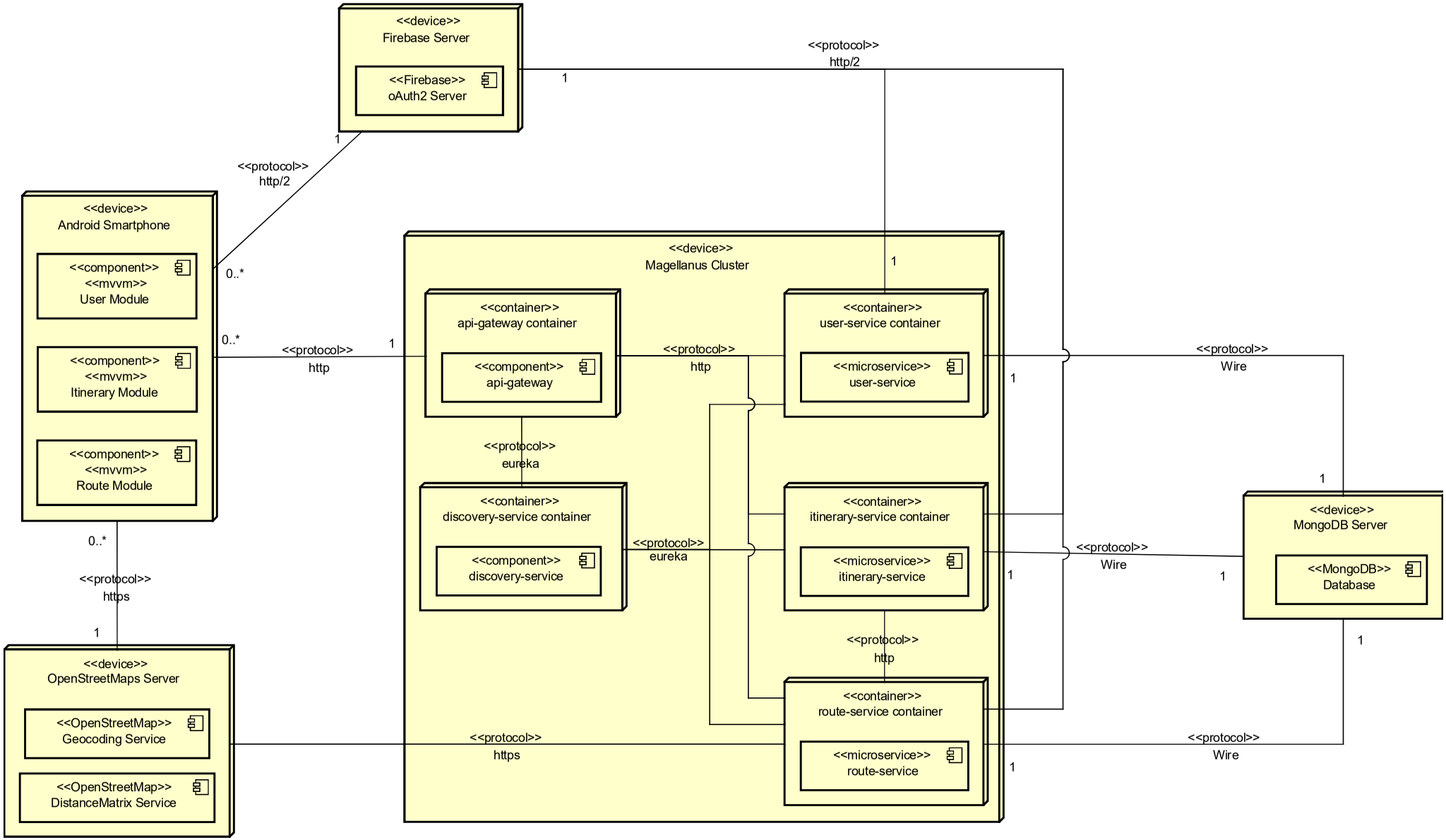
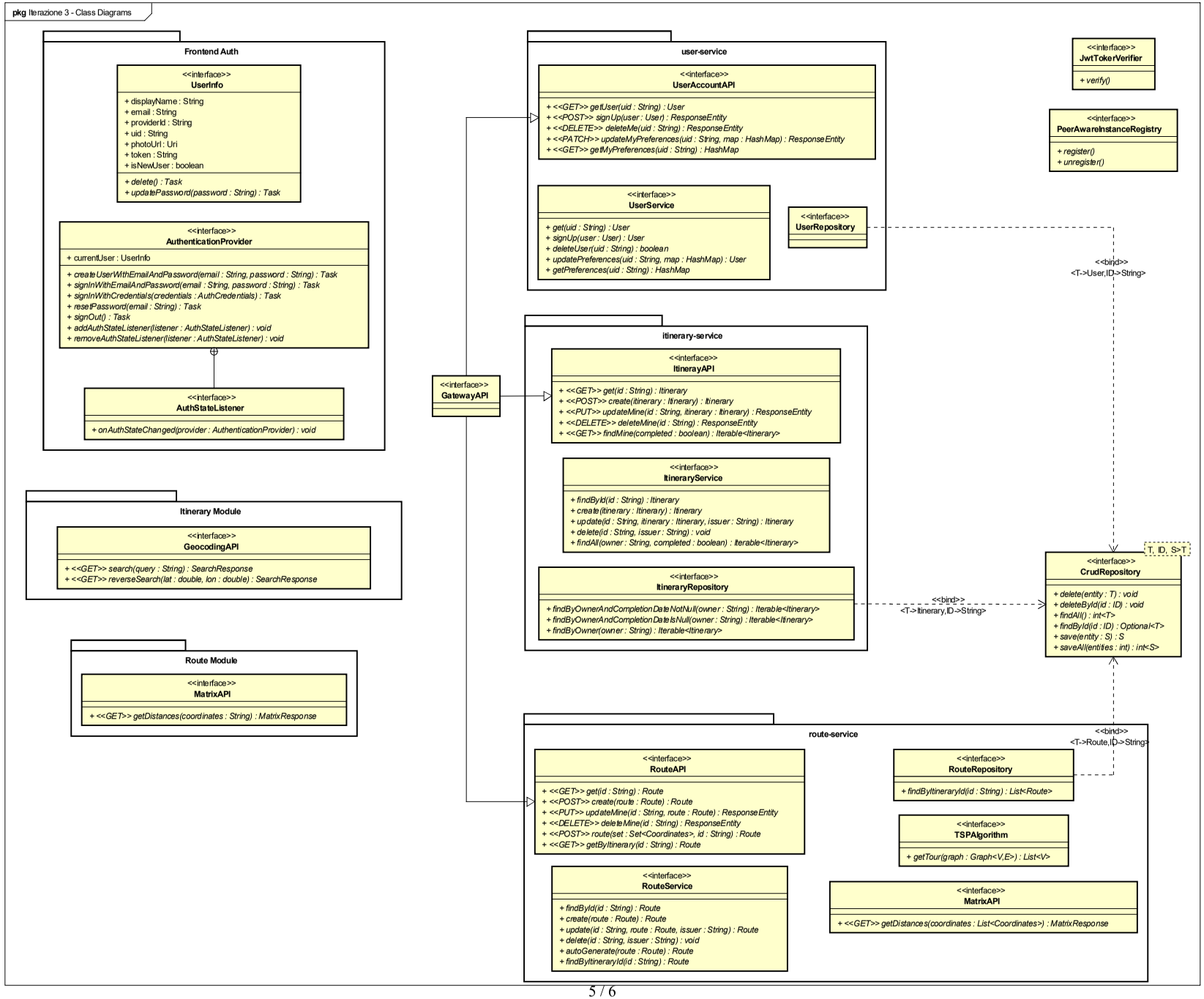
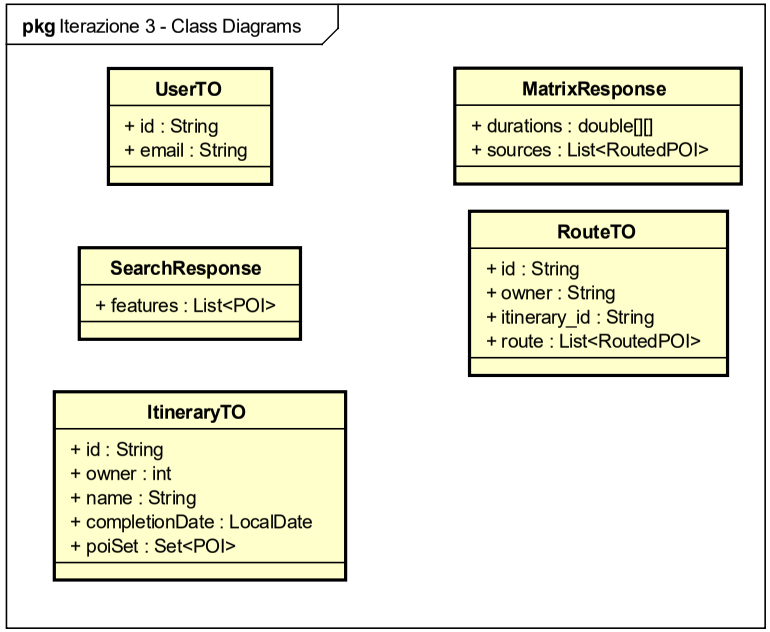

<<Firebase>>
oAuth2 Server
AuthenticationProvider
JwtTokenVerifier
<<subsystem>>
Magellanus App
<<component>>
<<mvvm>>
User Module
<<component>>
FrontendAuth
UserInfo
<<component>>
<<mvvm>>
Itinerary Module
<<component>>
<<mvvm>>
Route Module
GatewayAPI
<<rest>>
<<subsystem>>
Magellanus Backend
<<component>>
api-gateway
<<rest>>
UserAccountAPI
<<microservice>>
user-service
PeerAwareInstanceRegistry
<<rest>>
ItineraryAPI
<<microservice>>
itinerary-service
PeerAwareInstanceRegistry
<<rest>>
RouteAPI
<<microservice>>
route-service
PeerAwareInstanceRegistry
PeerAwareInstanceRegistry
<<component>>
discovery-service
JwtTokenVerifier
JwtTokenVerifier
JwtTokenVerifier
MongoCollection
<<MongoDB>>
Database
GeocodingAPI
<<rest>>
<<OpenStreetMap>>
Geocoding Service
MatrixAPI
<<rest>>
<<OpenStreetMap>>
DistanceMatrix Service


cmp
JwtTokenVerifier
MatrixAPI
<<microservice>>
route-service
<<component>>
Security
<<service>>
RoutingAlgorithm
RoutingAlgorithm
<<controller>>
RouteController
<<service>>
RouteService
<<component>>
RouteRepository
RouteAPI
RouteService
RouteRepository
MongoCollection
PeerAwareInstanceRegistry


pkg Iterazione 3 - Class Diagrams
Itinerary Module
Itinerary
+ id : String
+ owner : String
+ name : String
+ completionDate : LocalDate
1
0..*
POI
+ lat : double
+ lon : double
+ name : String
+ inRoute : boolean
1
1
Address
+ country : String
+ countryCode : String
+ city : String
+ postcode : String
+ county : String
+ street : String
+ state : String
Route Module
Route
+ id : String
+ owner : String
+ itinerary_id : String
1
0..*
RoutedPOI
+ name : String
+ distance : double
1
1
Coordinates
+ lat : double
+ lon : double
user-service
<<document>>
User
+ preferences : HashMap
+ id : String
+ email : String
K, E
HashMap
itinerary-service
<<document>>
Itinerary
+ name : String
+ completionDate : LocalDate
+ id : String
+ owner : int
1
0..*
POI
+ lat : double
+ lon : double
+ inRoute : boolean
route-service
Route
+ id : String
+ owner : String
+ itinerary_id : String
1
0..*
RoutedPOI
+ name : String
+ distance : double
1
1
Coordinates
+ lat : double
+ lon : double


<<device>>
Firebase Server
<<Firebase>>
oAuth2 Server
<<protocol>>
http/2
1
1
<<protocol>>
http/2
<<device>>
Android Smartphone
<<component>>
<<mvvm>>
User Module
<<component>>
<<mvvm>>
Itinerary Module
<<component>>
<<mvvm>>
Route Module
0..*
0..*
<<protocol>>
http
1
<<device>>
Magellanus Cluster
1
<<container>>
api-gateway container
<<component>>
api-gateway
<<protocol>>
http
<<container>>
user-service container
<<microservice>>
user-service
1
<<protocol>>
Wire
<<protocol>>
eureka
<<container>>
discovery-service container
<<component>>
discovery-service
<<protocol>>
eureka
<<container>>
itinerary-service container
<<microservice>>
itinerary-service
1
<<protocol>>
Wire
1
<<device>>
MongoDB Server
<<MongoDB>>
Database
1
1
<<protocol>>
http
<<container>>
route-service container
<<microservice>>
route-service
1
0..*
<<protocol>>
https
1
<<device>>
OpenStreetMaps Server
<<OpenStreetMap>>
Geocoding Service
<<OpenStreetMap>>
DistanceMatrix Service
<<protocol>>
https
<<protocol>>
Wire


pkg Iterazione 3 - Class Diagrams
Frontend Auth
<<interface>> UserInfo
+ displayName : String
+ email : String
+ providerId : String
+ uid : String
+ photoUrl : Uri
+ token : String
+ isNewUser : boolean
+ delete() : Task
+ updatePassword(password : String) : Task
<<interface>> AuthenticationProvider
+ currentUser : UserInfo
+ createUserWithEmailAndPassword(email : String, password : String) : Task
+ signInWithEmailAndPassword(email : String, password : String) : Task
+ signInWithCredentials(credentials : AuthCredentials) : Task
+ resetPassword(email : String) : Task
+ signOut() : Task
+ addAuthStateListener(listener : AuthStateListener) : void
+ removeAuthStateListener(listener : AuthStateListener) : void
<<interface>> AuthStateListener
+ onAuthStateChanged(provider : AuthenticationProvider) : void
Itinerary Module
<<interface>> GeocodingAPI
+ <<GET>> search(query : String) : SearchResponse
+ <<GET>> reverseSearch(lat : double, lon : double) : SearchResponse
Route Module
<<interface>> MatrixAPI
+ <<GET>> getDistances(coordinates : String) : MatrixResponse
<<interface>> GatewayAPI
user-service
<<interface>> UserAccountAPI
+ <<GET>> getUser(uid : String) : User
+ <<POST>> signUp(user : User) : ResponseEntity
+ <<DELETE>> deleteMe(uid : String) : ResponseEntity
+ <<PATCH>> updateMyPreferences(uid : String, map : HashMap) : ResponseEntity
+ <<GET>> getMyPreferences(uid : String) : HashMap
<<interface>> UserService
+ get(uid : String) : User
+ signUp(user : User) : User
+ deleteUser(uid : String) : boolean
+ updatePreferences(uid : String, map : HashMap) : User
+ getPreferences(uid : String) : HashMap
<<interface>> UserRepository
<<interface>> JwtTokerVerifier
+ verify()
<<interface>> PeerAwareInstanceRegistry
+ register()
+ unregister()
<<bind>> <T->User,ID->String>
itinerary-service
<<interface>> ItinerayAPI
+ <<GET>> get(id : String) : Itinerary
+ <<POST>> create(itinerary : Itinerary) : Itinerary
+ <<PUT>> updateMine(id : String, itinerary : Itinerary) : ResponseEntity
+ <<DELETE>> deleteMine(id : String) : ResponseEntity
+ <<GET>> findMine(completed : boolean) : Iterable<Itinerary>
<<interface>> ItineraryService
+ findById(id : String) : Itinerary
+ create(itinerary : Itinerary) : Itinerary
+ update(id : String, itinerary : Itinerary, issuer : String) : Itinerary
+ delete(id : String, issuer : String) : void
+ findAll(owner : String, completed : boolean) : Iterable<Itinerary>
<<interface>> ItineraryRepository
+ findByOwnerAndCompletionDateNotNull(owner : String) : Iterable<Itinerary>
+ findByOwnerAndCompletionDateIsNull(owner : String) : Iterable<Itinerary>
+ findByOwner(owner : String) : Iterable<Itinerary>
<<bind>> <T->Itinerary,ID->String>
T, ID, S>T
<<interface>> CrudRepository
+ delete(entity : T) : void
+ deleteById(id : ID) : void
+ findAll() : int<T>
+ findById(id : ID) : Optional<T>
+ save(entity : S) : S
+ saveAll(entities : int) : int<S>
route-service
<<bind>> <T->Route,ID->String>
<<interface>> RouteAPI
+ <<GET>> get(id : String) : Route
+ <<POST>> create(route : Route) : Route
+ <<PUT>> updateMine(id : String, route : Route) : ResponseEntity
+ <<DELETE>> deleteMine(id : String) : ResponseEntity
+ <<POST>> route(set : Set<Coordinates>, id : String) : Route
+ <<GET>> getByItinerary(id : String) : Route
<<interface>> RouteRepository
+ findByItineraryId(id : String) : List<Route>
<<interface>> TSPAlgorithm
+ getTour(graph : Graph<V,E>) : List<V>
<<interface>> RouteService
+ findById(id : String) : Route
+ create(route : Route) : Route
+ update(id : String, route : Route, issuer : String) : Route
+ delete(id : String, issuer : String) : void
+ autoGenerate(route : Route) : Route
+ findByItineraryId(id : String) : Route
<<interface>> MatrixAPI
+ <<GET>> getDistances(coordinates : List<Coordinates>) : MatrixResponse


pkg Iterazione 3 - Class Diagrams
UserTO
+ id : String
+ email : String
MatrixResponse
+ durations : double[][]
+ sources : List<RoutedPOI>
RouteTO
+ id : String
+ owner : String
+ itinerary_id : String
+ route : List<RoutedPOI>
SearchResponse
+ features : List<POI>
ItineraryTO
+ id : String
+ owner : int
+ name : String
+ completionDate : LocalDate
+ poiSet : Set<POI>