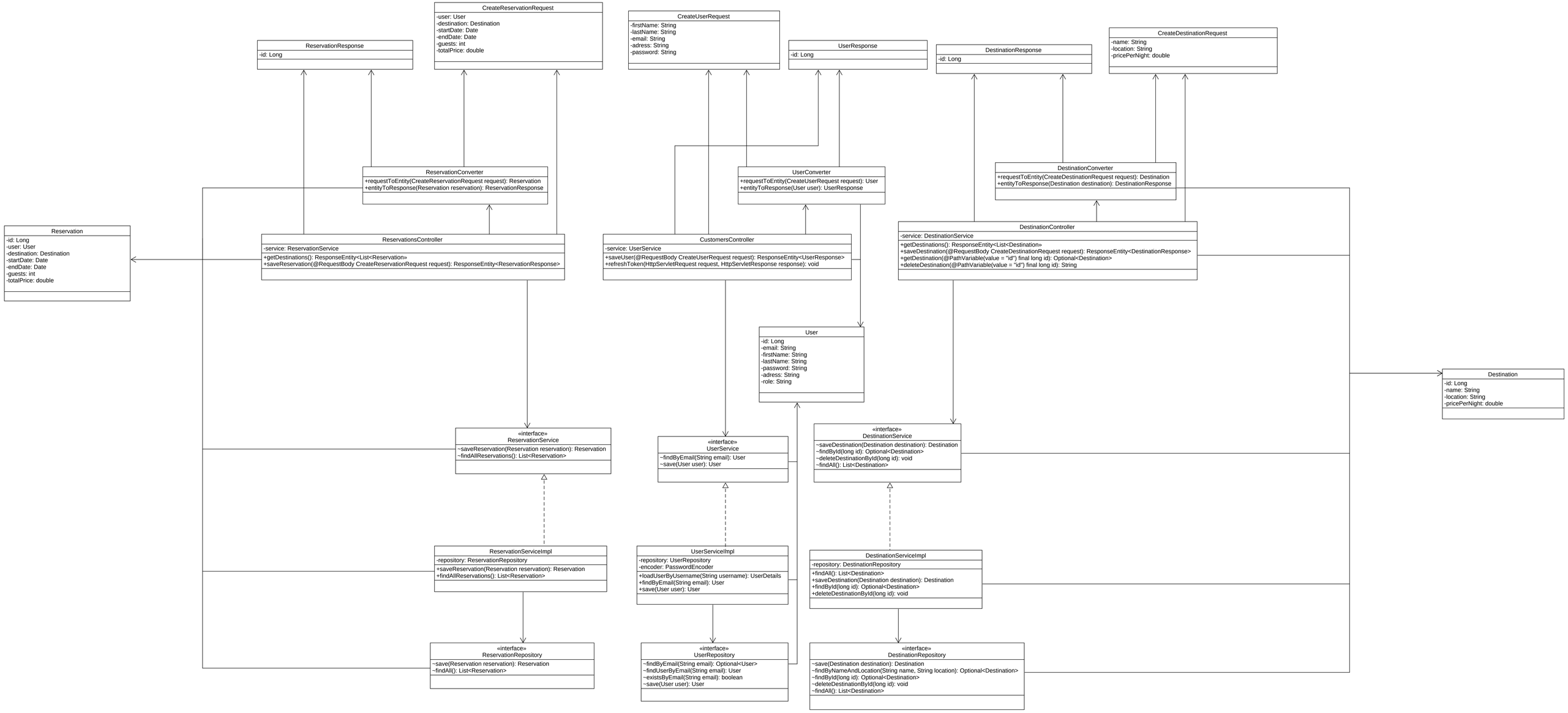
CreateReservationRequest
-user: User
-destination: Destination
-startDate: Date
-endDate: Date
-guests: int
-totalPrice: double
CreateUserRequest
-firstName: String
-lastName: String
-email: String
-adress: String
-password: String
ReservationResponse
-id: Long
UserResponse
-id: Long
DestinationResponse
-id: Long
CreateDestinationRequest
-name: String
-location: String
-pricePerNight: double
ReservationConverter
+requestToEntity(CreateReservationRequest request): Reservation
+entityToResponse(Reservation reservation): ReservationResponse
UserConverter
+requestToEntity(CreateUserRequest request): User
+entityToResponse(User user): UserResponse
DestinationConverter
+requestToEntity(CreateDestinationRequest request): Destination
+entityToResponse(Destination destination): DestinationResponse
Reservation
-id: Long
-user: User
-destination: Destination
-startDate: Date
-endDate: Date
-guests: int
-totalPrice: double
ReservationsController
-service: ReservationService
+getDestinations(): ResponseEntity<List<Reservation>>
+saveReservation(@RequestBody CreateReservationRequest request): ResponseEntity<ReservationResponse>
CustomersController
-service: UserService
+saveUser(@RequestBody CreateUserRequest request): ResponseEntity<UserResponse>
+refreshToken(HttpServletRequest request, HttpServletResponse response): void
DestinationController
-service: DestinationService
+getDestinations(): ResponseEntity<List<Destination>>
+saveDestination(@RequestBody CreateDestinationRequest request): ResponseEntity<DestinationResponse>
+getDestination(@PathVariable(value = "id") final long id): Optional<Destination>
+deleteDestination(@PathVariable(value = "id") final long id): String
User
-id: Long
-email: String
-firstName: String
-lastName: String
-password: String
-adress: String
-role: String
Destination
-id: Long
-name: String
-location: String
-pricePerNight: double
«interface»
ReservationService
~saveReservation(Reservation reservation): Reservation
~findAllReservations(): List<Reservation>
«interface»
UserService
~findByEmail(String email): User
~save(User user): User
«interface»
DestinationService
~saveDestination(Destination destination): Destination
~findById(long id): Optional<Destination>
~deleteDestinationById(long id): void
~findAll(): List<Destination>
ReservationServiceImpl
-repository: ReservationRepository
+saveReservation(Reservation reservation): Reservation
+findAllReservations(): List<Reservation>
UserServiceImpl
-repository: UserRepository
-encoder: PasswordEncoder
+loadUserByUsername(String username): UserDetails
+findByEmail(String email): User
+save(User user): User
DestinationServiceImpl
-repository: DestinationRepository
+findAll(): List<Destination>
+saveDestination(Destination destination): Destination
+findById(long id): Optional<Destination>
+deleteDestinationById(long id): void
«interface»
ReservationRepository
~save(Reservation reservation): Reservation
~findAll(): List<Reservation>
«interface»
UserRepository
~findByEmail(String email): Optional<User>
~findUserByEmail(String email): User
~existsByEmail(String email): boolean
~save(User user): User
«interface»
DestinationRepository
~save(Destination destination): Destination
~findByNameAndLocation(String name, String location): Optional<Destination>
~findById(long id): Optional<Destination>
~deleteDestinationById(long id): void
~findAll(): List<Destination>