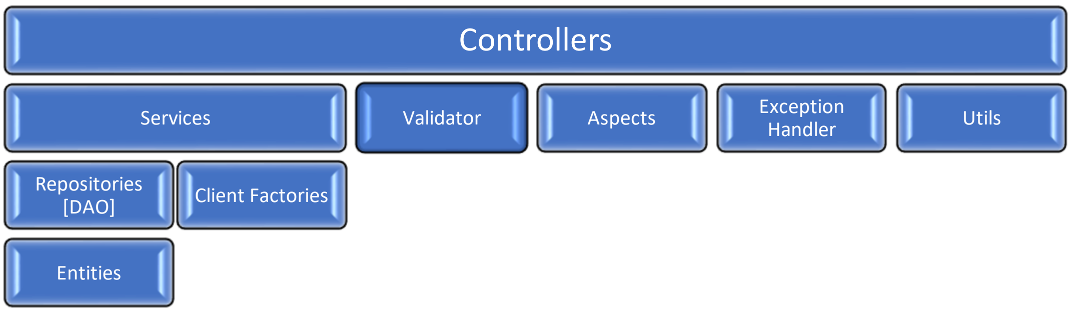
Mobile Info Application High Level Architecture

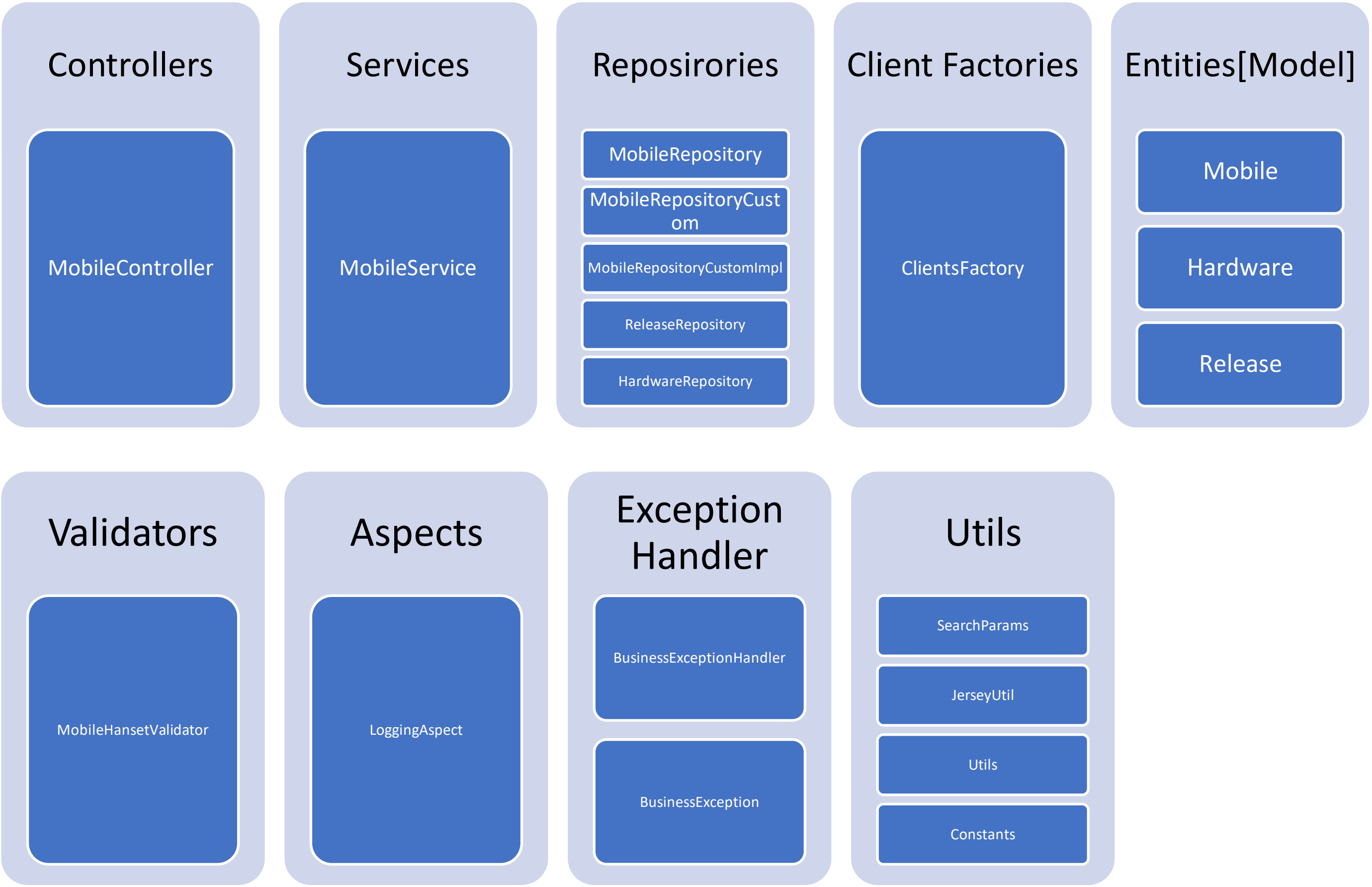
Contents:

System Architecture:

- Controllers
- Services
- Repositories (DAO)
- Aspects
- Exception Handler
- Client Factories
- Validators
- Utils



Architecture Classes:



Controllers

MobileController

Services

MobileService

Reposirories

MobileRepository

MobileRepositoryCustom

MobileRepositoryCustomImpl

ReleaseRepository

HardwareRepository

Client Factories

ClientsFactory

Entities[Model]

Mobile

Hardware

Release

Validators

MobileHansetValidator

Aspects

LoggingAspect

Exception Handler

BusinessExceptionHandler

BusinessException

Utils

SearchParams

JerseyUtil

Utils

Constants

- **Controllers**
    - MobileController:
        - Mobile API Endpoint
        - *load*: loading data from data client API to save it in our application data base
        - *find*: search for mobile data by one, more or no parameters

- **Services:**
    - MobileService
        - Business Layer for implement required business logic on data
        - *LoadAllMobiles*: used to load mobile data in application data base
        - *findMobiles*: used to search for mobile by required parameters

- **Repositories:**
    - MobileRepository
        - Data Access Layer contains default methods for add, edit, delete, search Mobile Data

    - MobileRepositoryCustom
        - Custom repository interface for customize queries for Mobile in DB

    - MobileRepositoryCustomImpl
        - Customized queries implementation for Mobile data in DB

    - HardwareRepository
        - Data Access Layer contains default methods for add, edit, delete, search Hardware Data

    - ReleaseRepository

        - Data Access Layer contains default methods for add, edit, delete, search Release Data

- **Aspects :**
    - LoggingAspect :
        - Generic Logging Aspect for logging methods inputs (showing how to act with aspects)

- **Exception Handlers:**
    - BusinessException:
        - Customized Runtime Exception for logging throwed exception.
    - BusinessExceptionHandler:
        - Custom Exception Handller for controller that return failed message in case of throws exception.

- **Client Factories**
    - ClientsFactory:
        - Factory class that interact with all clients to unify remote clients Handlers.

- **Validators**
    - MobileHansetValidator:
        - Mobile Controller Validator class to validate controller data

- **Utils :**
  Contains util classes and function that is used many times in the application

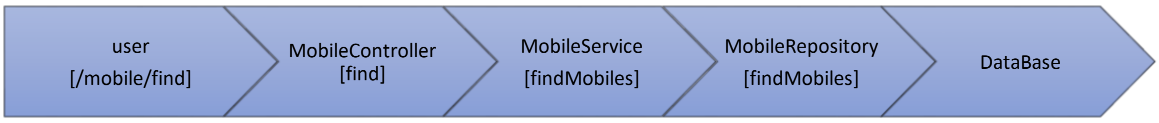
    - Constants:
        - Unify constants Class in case of modify it easy and can be available for all classes.

    - JerseyUtil:
        - Util Class for handle calling remote service endpoint

    - SearchParams:
        - Enum used to validate parameters sent by user in search request

    - Utils:
        - Contains some generic util methods like converting from JSON to String and vise versa

Search Process Flow:


user
[/mobile/find]
MobileController
[find]
MobileService
[findMobiles]
MobileRepository
[findMobiles]
DataBase