Service Layer **TaskService** - boardfacade: BoardFacade + due: string + AddTask(bn, title, description, due, id, email): Response<TaskSL> + MoveTask(bn, column, id, email): Response<> + id: int + UpdateTask(bn, title, description, due, id, email, column): Response<>

BoardService

- boardfacade: BoardFacade
- + CreateBoard(bn, email): Response<BoardSL>
- + DeleteBoard(bn, email): Response<>
- + LimitColumn(bn, column, limit, email): Response<>

UserService

- userfacade: UserFacade
- + Login(un, pass): Response<UserSL>
- + Register(un, pass, email): Response<UserSL>
- + Logout(): Response<>

ServiceFactory

- +,- US: UserService
- +,- BS: BoardService
- +,- TS: TaskService

TaskSL

- + title: string
- + description: string
- + creation_time: string

BoardSL

- + name: string
- + backlog: List<TaskSL>
- + in_progrerss: List<TaskSL>
- + done: List<TaskSL>

UserSL

- + un: string
- + pass: string
- + email: string
- + boards: List<BoardBL>

Response

- + ErrorMsg: string
- + RetVal: object

Business Layer

UserFacade

- users: dict<un, User>
- + Register(un, pass, email): UserBL
- + Login(un, pass): UserBL
- + Logout(): void

User package

UserBL

- +,- un: string
- pass: string
- email: string
- +,- boards: List<BoardBL>
- + Login(pass): bool
- + Logout(): void

ColumnBL

- + limit: int
- + tasks: List<TaskBL>
- + LimitColumn(limit, email): void
- + AddTask(task, email): void
- + DeleteTask(id, email): void
- + UpdateTask(title, description, due, id, email): void

BoardBL

- + name: string
- + columns: List<ColumnBL>
- + MoveTask(column, id, email): void
- + DeleteBoard(email): void
- + LimitColumn(column, limit, email): void
- + AddTask(title, description, due, id, email): void
- + UpdateTask(title, description, due, id, email, column): void

Board package

BoardFacade

- boards: dict<name, board>
- + CreateBoard(bn, email): BoardBL
- + DeleteBoard(bn, email): void
- + MoveTask(bn, column, id, email): void
- + AddTask(bn, title, description, due, email, id): TaskBL
- + UpdateTask(bn, title, description, due, id, email, column): TaskBL
- + LimitColumn(bn, column, limit, email): void

TaskBL

- + title: string
- + due: string
- + description: string
- + creation time: string
- + id: int
- + UpdateTask(title, description, due, id): void