IntervalTime

+ IntervalTime(GregorianCalendar, GregorianCalendar):

- endDate: GregorianCalendar
- startDate: GregorianCalendar

fullTime: String
startDate: GregorianCalendar
endDate: GregorianCalendar

Capsule

+ Capsule(String, String, String, String, String):

- id: int

+ setApproved(): void
+ generateHTML(): void
+ printCapsule(): void

keyWords: String

Utils

+ Utils():

+ print(Object): void
+ getGregorianCalendar(int, int, int): GregorianCalendar
+ menu(): void
+ validateLearning(): String
+ validateType(): String

---

1 plannedTime   * capsules

«create»

1 1

«create»

Stage

+ Stage(String):

- capsules: Capsule[]
- active: boolean
- stageName: String

+ setApproved(): void
+ searchCapsuleByKeyword(String): void
+ setInactive(): void
+ setActive(): void
+ addCapsule(Capsule): void

approvedStatus: boolean
active: boolean
stageName: String
realIntervalTime: String
plannedIntervalTime: IntervalTime
capsules: Capsule[]
plannedStartIntervalTime: IntervalTime

«create»

* stages

1

Stages

+ Stages():

- activeStage: int

- createStages(): void

activeStage: Stage

<<enumeration>>
State

+ State():

+ values(): State[]
+ valueOf(String): State

1 state

---

stages 1

«create»

1

Project

+ Project():

- stages: Stages

+ getPlannedDateOfEnd(): void
+ reject(): void
+ getClientName(): void
+ getPlannedDateOfStart(): void
+ approve(): void
+ getState(): void
+ getProjectManagersName(): void
+ getProjectName(): void
+ getProjectManagersPhone(): void
+ getBudget(): void

stages: Stages
date: GregorianCalendar
stage: Stage

---

* projects   «create»

«create»

1

Main

+ Main():

- validateProject(): int
+ main(String[]): void