



# SWOP – TaskMan iteratie 1

Door Tim Van Den Broecke,  
Joran Van de Woestijne,  
Vincent Van Gestel en  
Eli Vangrieken

# Inleiding

- 1.High level design

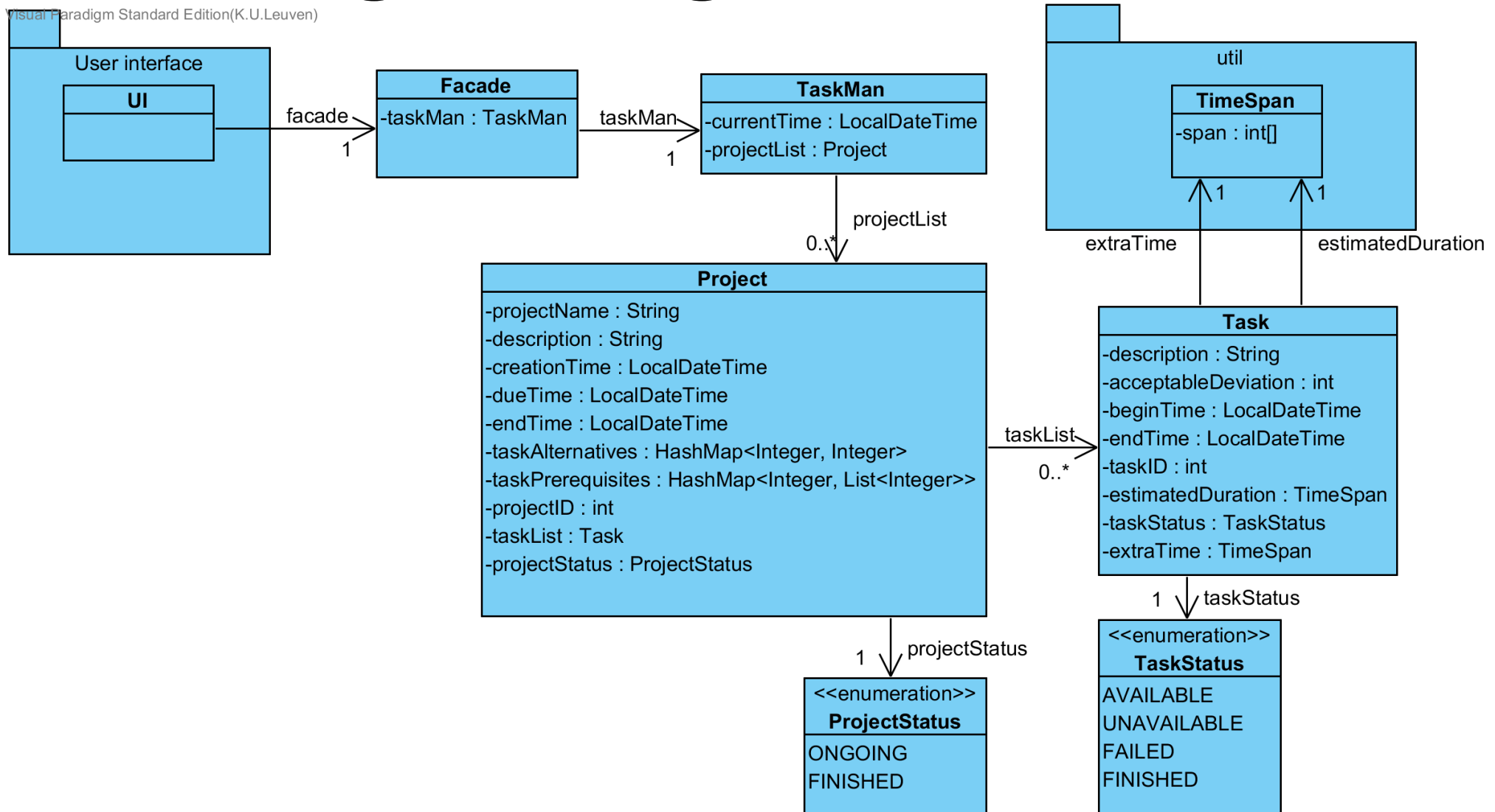
- 2.Detail design

- 3.Domain strength

- 4.Testing

# Het design – High level

Visual Paradigm Standard Edition(K.U.Leuven)



# Het design – Detail TaskMan

TaskMan
<<Property>> -currentTime : LocalDateTime -projectList : Project
-getProject(projectID : int) : Project +createProject(name : String, description : String, creationTime : LocalDateTime, dueTime : LocalDateTime) : boolean +getProjectName(projectID : int) : String +getProjectDescription(projectID : int) : String +getProjectCreationTime(projectID : int) : LocalDateTime +getProjectDueTime(projectID : int) : LocalDateTime +getProjectEndTime(projectID : int) : LocalDateTime +getProjectStatus(projectID : int) : String +TaskMan() +TaskMan(time : LocalDateTime) +advanceTimeTo(time : LocalDateTime) : boolean +createProject(name : String, description : String, dueTime : LocalDateTime) : boolean +createTask(projectID : int, description : String, estimatedDuration : int, acceptableDeviation : int, taskStatus : String, alternativeFor : Integer, prerequisiteTasks : List<Integer>, startTime : LocalDateTime, endTime : LocalDateTime) : boolean +createTask(projectID : int, description : String, estimatedDuration : int, acceptableDeviation : int, alternativeFor : Integer, prerequisiteTasks : List<Integer>) : boolean +getProjectDelay(projectID : int) : int [] +getAvailableTasks() : HashMap<Integer, List<Integer>> +getAvailableTasks(projectID : int) : List<Integer> +getProjectAmount() : int +getTaskAmount(projectID : int) : int

# Het design – Detail TaskMan (2)

```
+getTaskDescription(projectID : int, taskID : int) : String
+getTaskStartTime(projectID : int, taskID : int) : LocalDateTime
+getEstimatedTaskDuration(projectID : int, taskID : int) : int
+getAcceptableTaskDeviation(projectID : int, taskID : int) : int
+hasTaskEnded(projectID : int, taskID : int) : boolean
+getTaskEndTime(projectID : int, taskID : int) : LocalDateTime
+getTaskStatus(projectID : int, taskID : int) : String
+hasTaskPrerequisites(projectID : int, taskID : int) : boolean
+getTaskPrerequisitesFor(projectID : int, taskID : int) : List<Integer>
+hasTaskAlternative(projectID : int, taskID : int) : boolean
+getTaskAlternativeTo(projectID : int, taskID : int) : int
+setTaskFinished(projectID : int, taskID : int, startTime : LocalDateTime, endTime : LocalDateTime) : boolean
+setTaskFailed(projectID : int, taskID : int, startTime : LocalDateTime, endTime : LocalDateTime) : boolean
+isProjectOnTime(projectID : int) : boolean
+isTaskUnacceptableOverdue(projectID : int, taskID : int) : boolean
+isTaskOnTime(projectID : int, taskID : int) : boolean
+getTaskOverTimePercentage(projectID : int, taskID : int) : int
+isProjectFinished(projectID : int) : boolean
+isProjectEstimatedOnTime(projectID : int) : boolean
+getEstimatedProjectDelay(projectID : int) : int []
-isValidProjectID(PID : int) : boolean
```

# Het design – Detail Project

## Project

```
-description : String
-creationTime : LocalDateTime
-dueTime : LocalDateTime
-endTime : LocalDateTime
-taskAlternatives : HashMap<Integer, Integer>
-taskPrerequisites : HashMap<Integer, List<Integer>>
-taskList : Task
-projectStatus : ProjectStatus
-projectID : int
-projectName : String

+Project(projectID : int, projectName : String, description : String, creationTime : LocalDateTime, dueTime : LocalDateTime)
-updateTaskStatus(task : Task) : void
-getTask(taskID : int) : Task
-isValidTaskID(taskID : int) : boolean
+getProjectDescription() : String
+getProjectCreationTime() : LocalDateTime
+getProjectDueTime() : LocalDateTime
+getProjectEndTime() : LocalDateTime
+getAllAlternatives() : HashMap<Integer, Integer>
+createTask(description : String, estimatedDuration : int, acceptableDeviation : int, taskStatus : String, alternativeFor : int, prerequisiteTasks : List<Integer>, startTime : LocalDateTime, endTime : LocalDateTime) : boolean
+createTask(description : String, estimatedDuration : int, acceptableDeviation : int, alternativeFor : int, prerequisiteTasks : List<Integer>) : boolean
-recalculateProjectStatus() : void
+getProjectStatus() : String
+getTaskIDs() : ImmutableList<Integer>
+hasAlternative(taskID : Integer) : boolean
-addAlternative(toReplace : int, alternative : int) : boolean
-isValidAlternative(toReplace : int, alternative : int) : boolean
-addPrerequisites(taskID : int, pre : List<Integer>) : boolean
-isValidPrerequisites(task : int, prerequisites : List<Integer>) : boolean
```

# Het design – Detail Project (2)

```
+getTaskAmount() : int
+getAvailableTasks() : List<Integer>
+getTaskDescription(taskID : int) : String
+getTaskStartTime(taskID : int) : LocalDateTime
+getEstimatedTaskDuration(taskID : int) : int
+getAcceptableTaskDeviation(taskID : int) : int
+hasTaskEnded(taskID : int) : boolean
+getTaskEndTime(taskID : int) : LocalDateTime
+getTaskStatus(taskID : int) : String
+hasPrerequisites(taskID : int) : boolean
+getPrerequisites(taskID : int) : List<Integer>
+isOnTime(current : LocalDateTime) : boolean
+getDelay(current : LocalDateTime) : int []
+setTaskFinished(taskID : int, startTime : LocalDateTime, endTime : LocalDateTime) : boolean
+setTaskFailed(taskID : int, startTime : LocalDateTime, endTime : LocalDateTime) : boolean
+getExtraTime(taskID : int) : TimeSpan
-hasFinishedAlternative(task : int) : boolean
-isValidNewTaskID(taskID : int) : boolean
+getAlternative(task : int) : int
+isTaskUnacceptableOverdue(taskID : int) : boolean
+isTaskOnTime(taskID : int) : boolean
+getTaskOverTimePercentage(taskID : int) : int
+isFinished() : boolean
+getEstimatedProjectDelay(currentTime : LocalDateTime) : int []
-getMaxDelayChain(taskID : int) : TimeSpan
-isPrerequisite(taskID : int) : boolean
-getDependants(taskID : int) : List<Integer>
+isEstimatedOnTime(currentTime : LocalDateTime) : boolean
-hasAvailableTasks() : boolean
```

# Het design – Detail Task

Task
<<Property>> -description : String <<Property>> -acceptableDeviation : int <<Property>> -beginTime : LocalDateTime <<Property>> -endTime : LocalDateTime <<Property>> -taskID : int -taskStatus : TaskStatus -extraTime : TimeSpan -estimatedDuration : TimeSpan  +isFinished() : boolean +getTimeElapsed(currentTime : LocalDateTime) : TimeSpan +getTimeSpan() : TimeSpan +Task(taskID : int, taskDescription : String, estimatedDuration : int, acceptableDeviation : int, extraTime : TimeSpan) +isFailed() : boolean +isAvailable() : boolean +isUnavailable() : boolean +hasEnded() : boolean +setBeginTime(beginTime : LocalDateTime) : void +setEndTime(endTime : LocalDateTime) : void +setAvailable() : boolean +setUnavailable() : boolean



# Het design – Detail Task (2)

```
-getTaskStatus() : TaskStatus
+getTaskStatusName() : String
+getStatus() : String
+setTaskFinished(startTime : LocalDateTime, endTime : LocalDateTime) : boolean
+setTaskFailed(startTime : LocalDateTime, endTime : LocalDateTime) : boolean
-setTaskStatus(startTime : LocalDateTime, endTime : LocalDateTime, status : TaskStatus) : boolean
-isValidTimeStamps(startTime : LocalDateTime, endTime : LocalDateTime) : boolean
+Task(taskID : int, taskDescription : String, estimatedDuration : int, acceptableDeviation : int, taskStatus : String, beginTime : LocalDateTime, endTime : LocalDateTime, extraTime : TimeSpan)
-isValidDeviation(deviation : int) : boolean
-isValidTaskID(taskID : int) : boolean
-isValidDescription(description : String) : boolean
-isValidDuration(duration : int) : boolean
+isOnTime() : boolean
+isUnacceptableOverdue() : boolean
+getOverTimePercentage() : int
```

# Het domein – Uitbreidbaarheid

## Gebruik van facade

- UI onafhankelijk van het systeem

## Aanpassingen kunnen zeer lokaal gebeuren

# Het domein - Robuustheid

Grote nadruk op defensiviteit

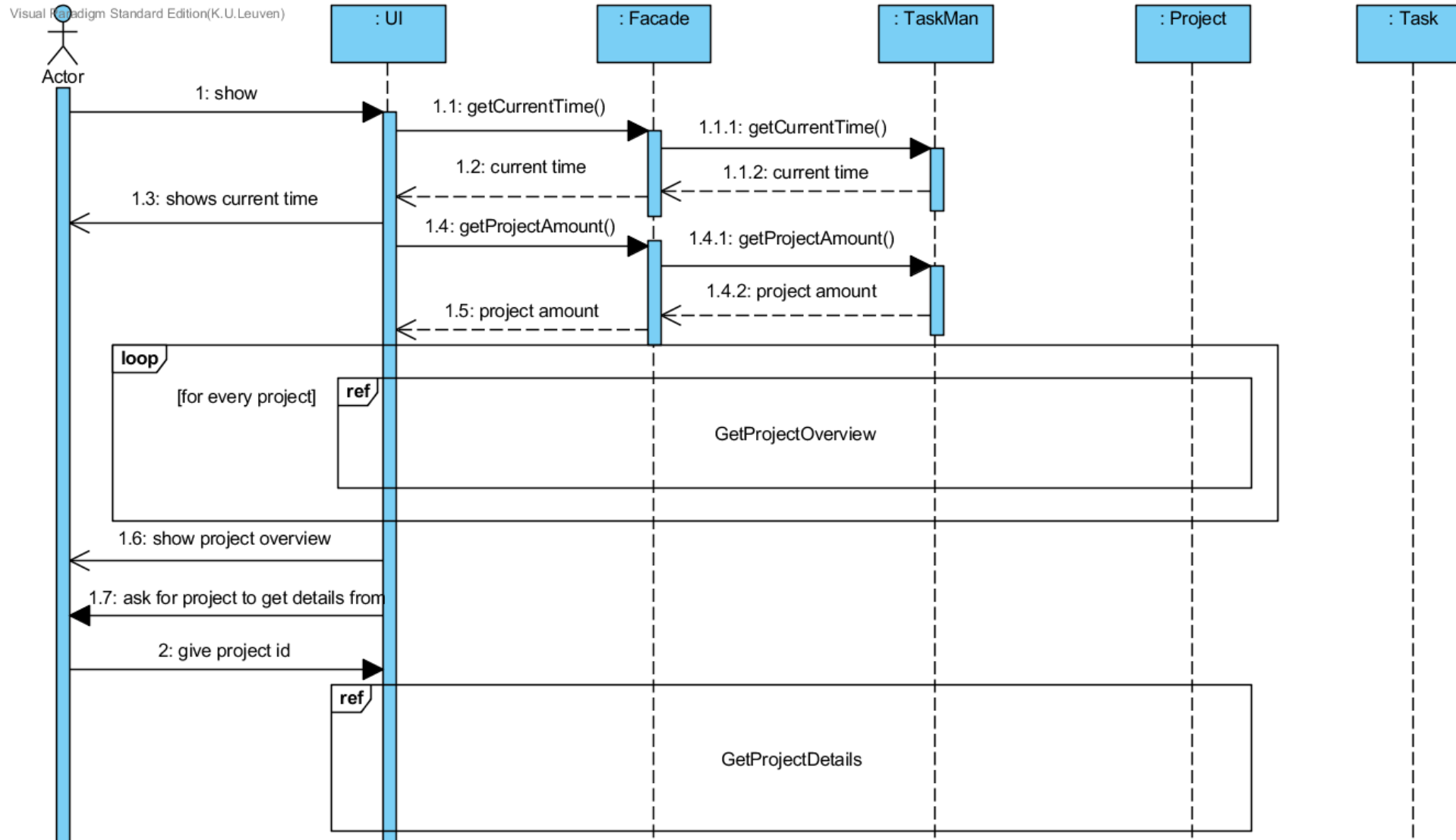
null

correcte identifiers

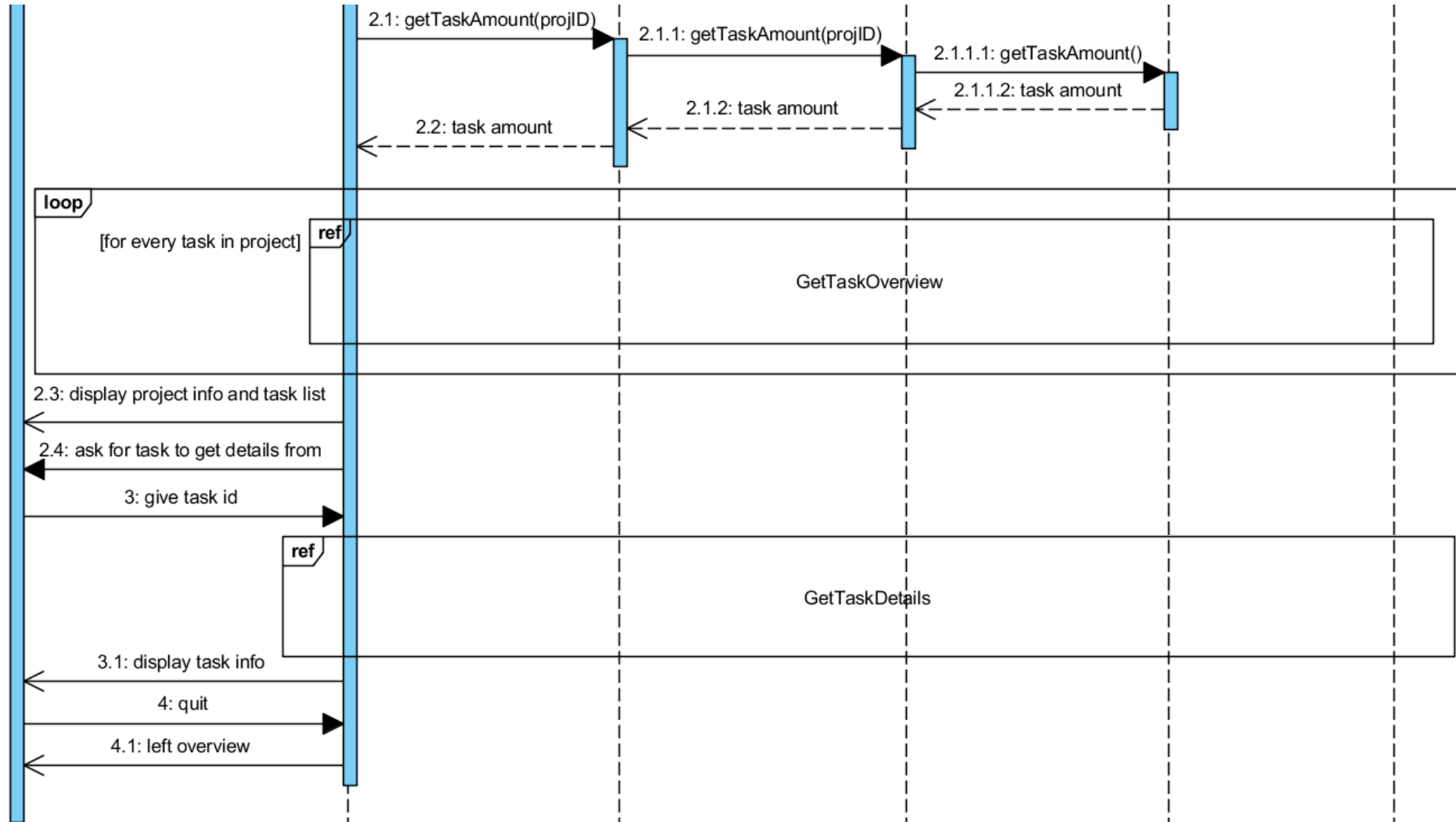
geen inconsistentie

# Testing – Use Case 1

Visual Paradigm Standard Edition(K.U.Leuven)

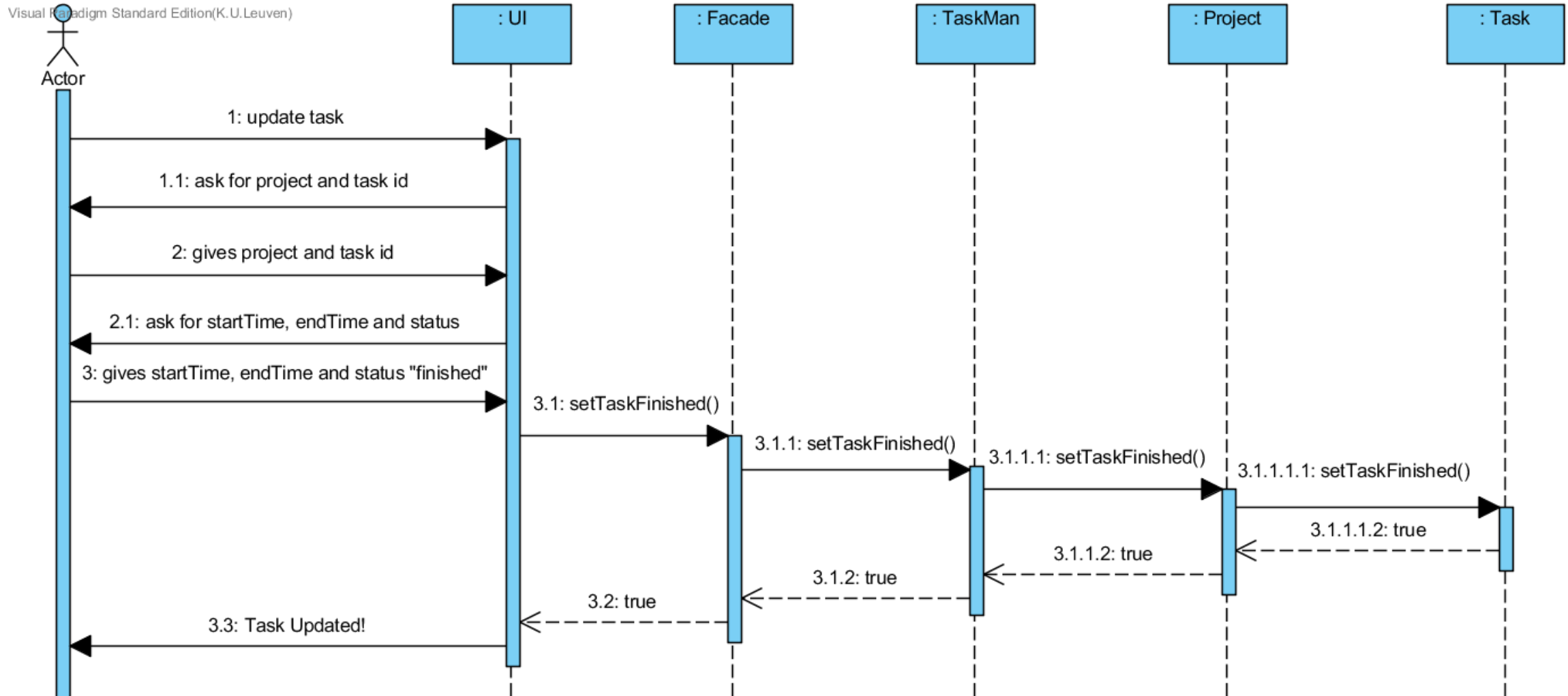


# Testing – Use Case 1 vervolg



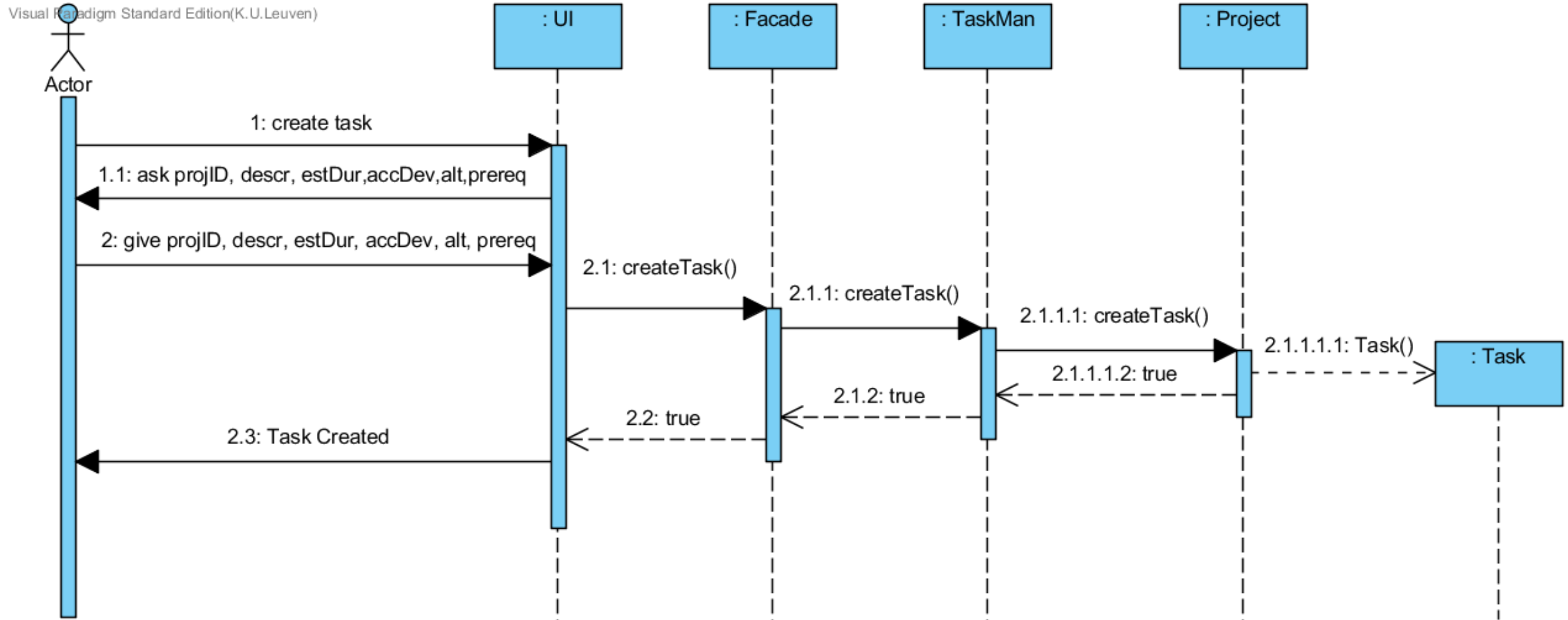
# Testing – Use Case 2

Visual Paradigm Standard Edition(K.U.Leuven)



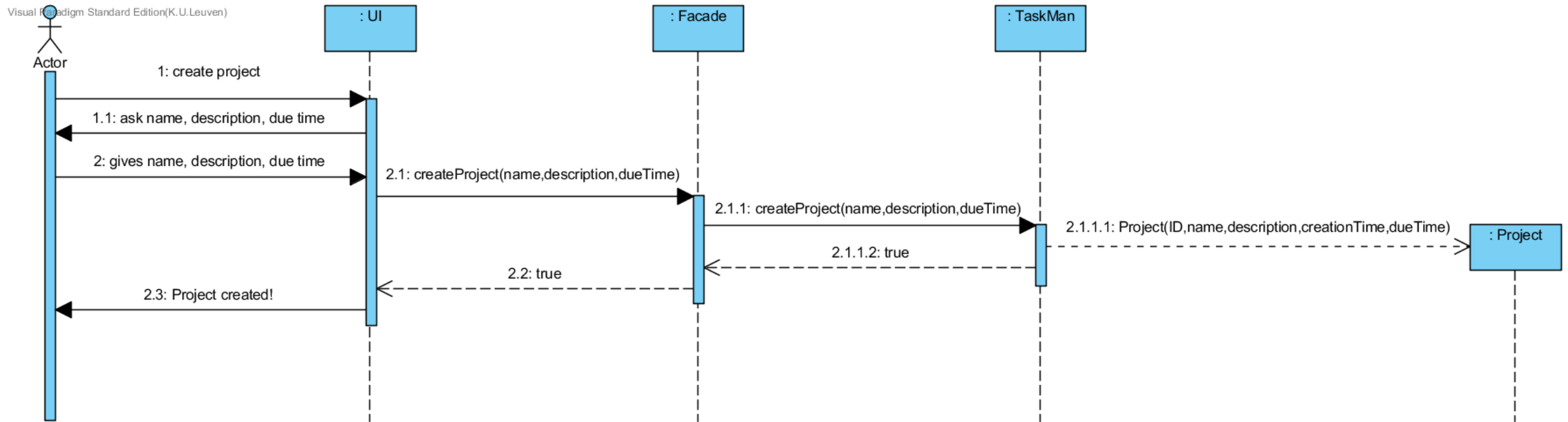
# Testing – Use Case 3

Visual Paradigm Standard Edition(K.U.Leuven)



# Testing – Use Case 4

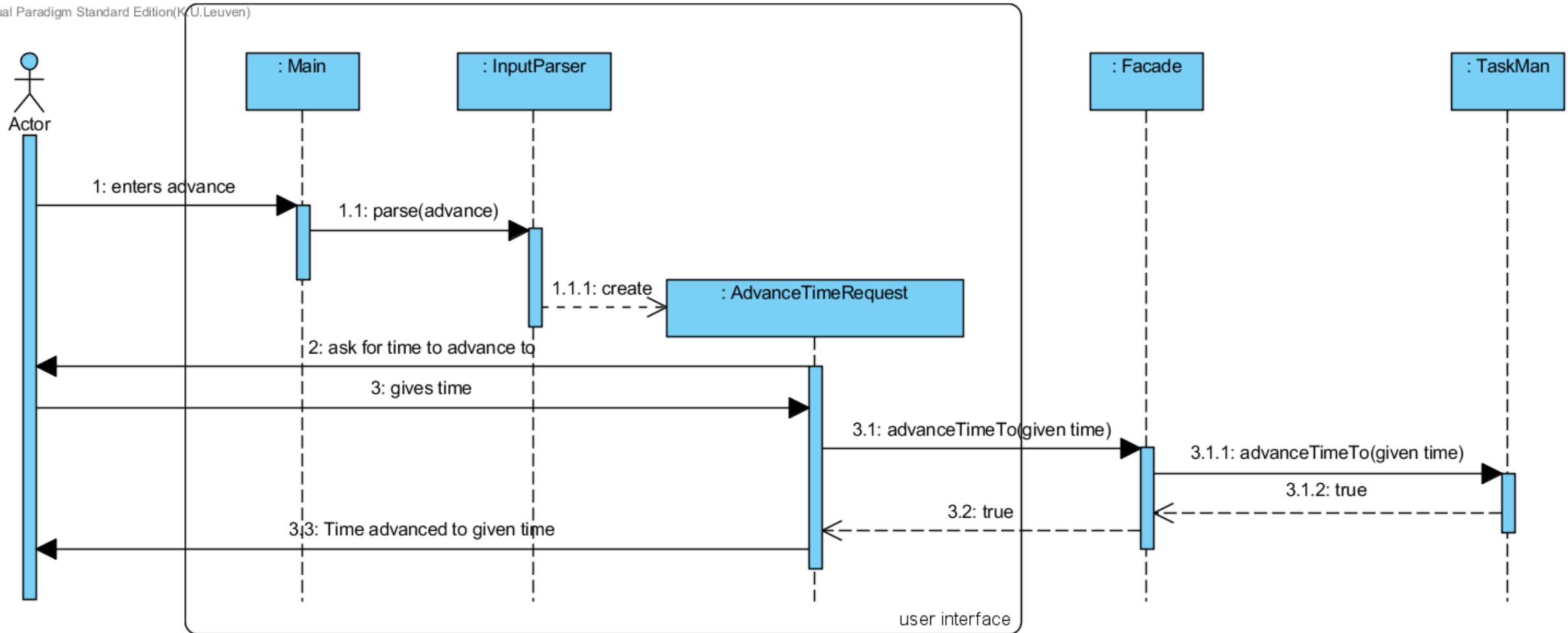
Visual Paradigm Standard Edition(K.U.Leuven)





# Testing – Use Case 5

Visual Paradigm Standard Edition(K.U.Leuven)



# Testing - Testsuite

*Demonstratie*

# Toekomstige aanpassingen

Tijdsberekening verbeteren

Hierarchie toevoegen aan Task

# Werkverdeling

Format: (Estimated) group / individual / study

- Vincent Van Gestel: Domain coordinator now, wildcard next  
45 / 20 / 5
- Joran Van de Woestijne: Design coordinator now, Domain next  
45 / 15 / 10
- Tim Van Den Broecke: Testing coordinator now, Design next  
45 / 10 / 2
- Eli Vangrieken: wildcard now, Testing next  
45 / 15 / 10