

SOFTWARE-DEVELOPMENT

PRESENTATION: ITERATION 2

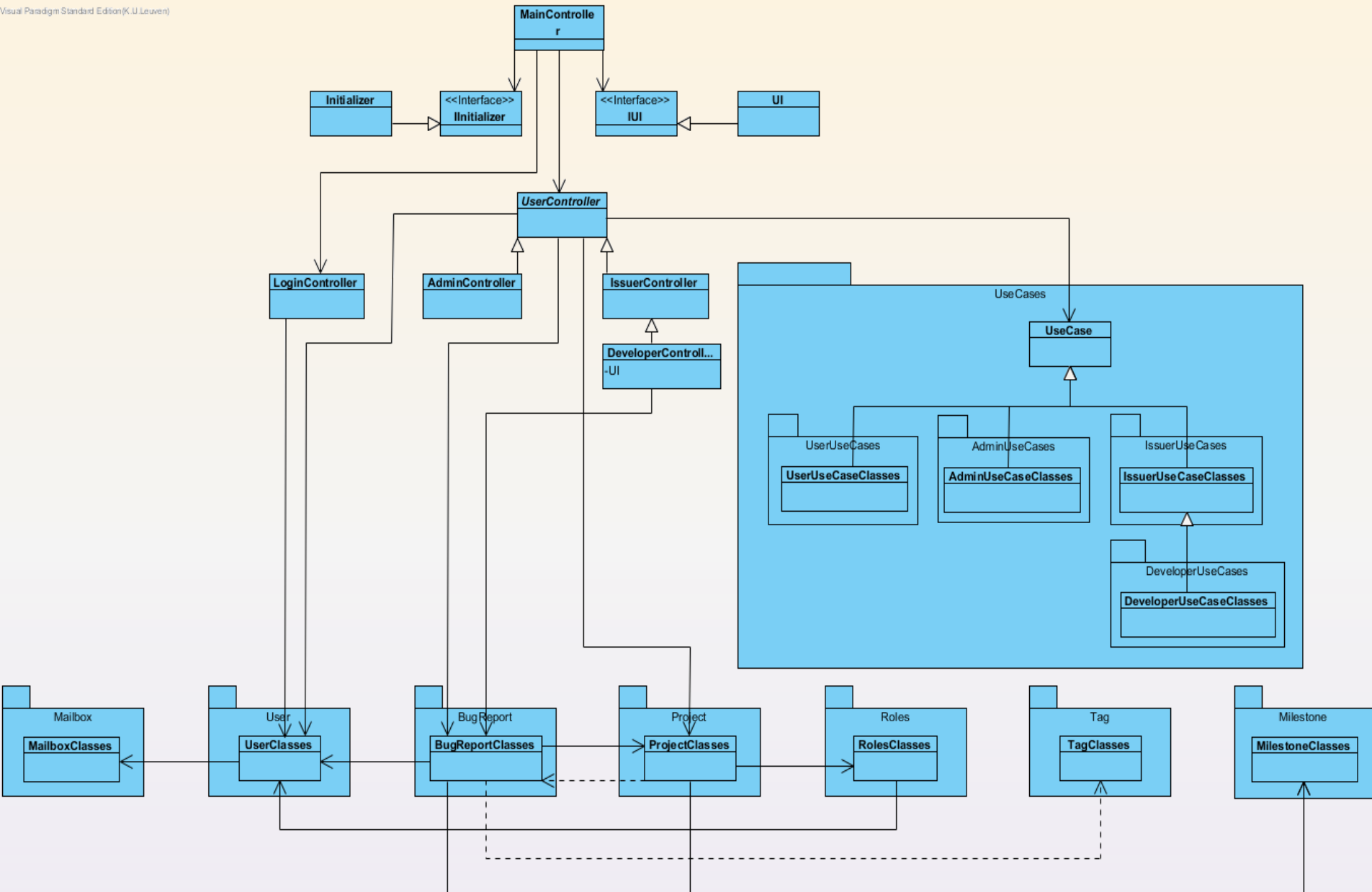
Karina Karapetyan

Laurens Cleemput

Tom Houben

Tri Tran

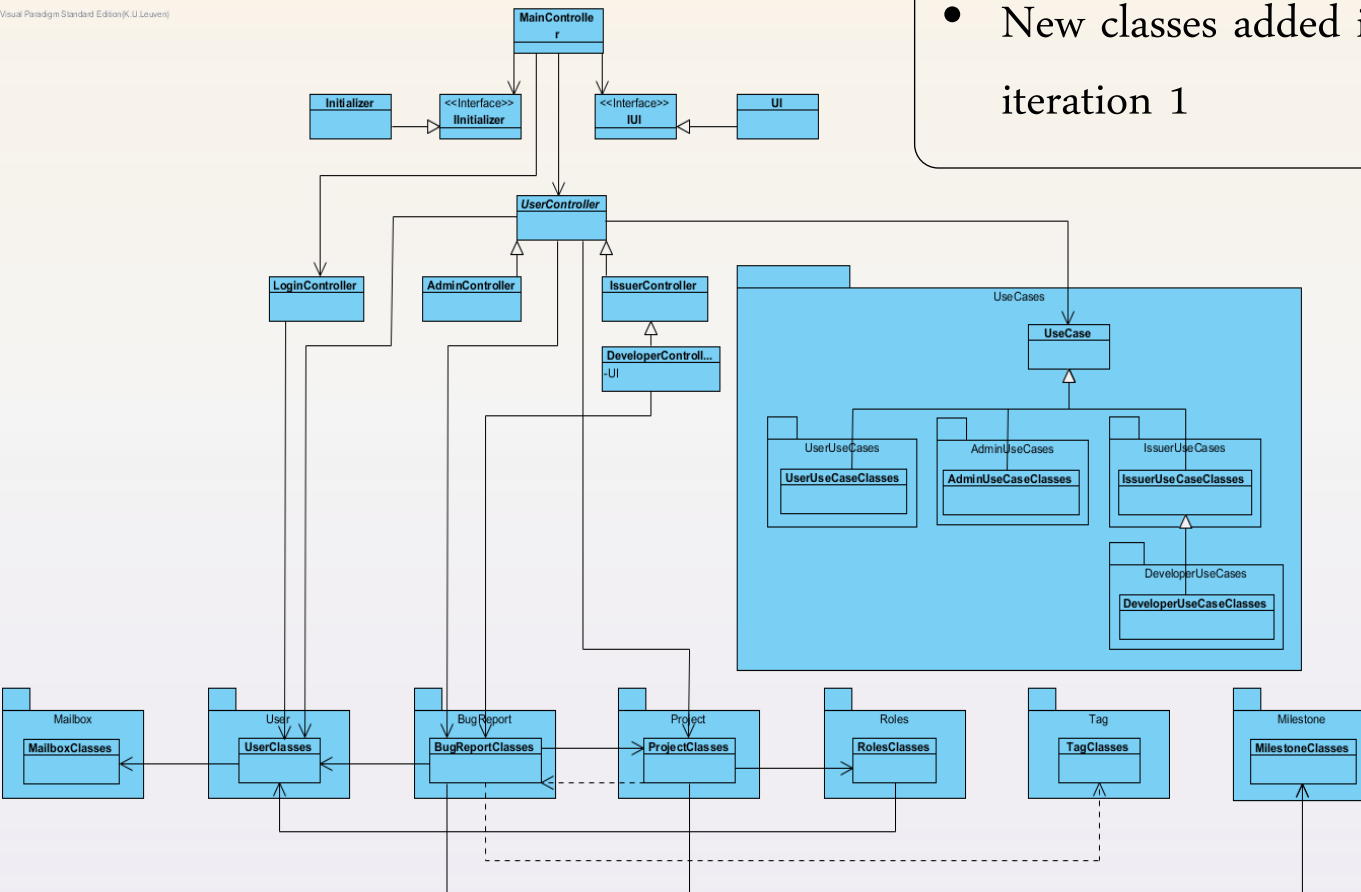
Visual Paradigm Standard Edition(K.U.Leuven)



HIGH LEVEL DESIGN

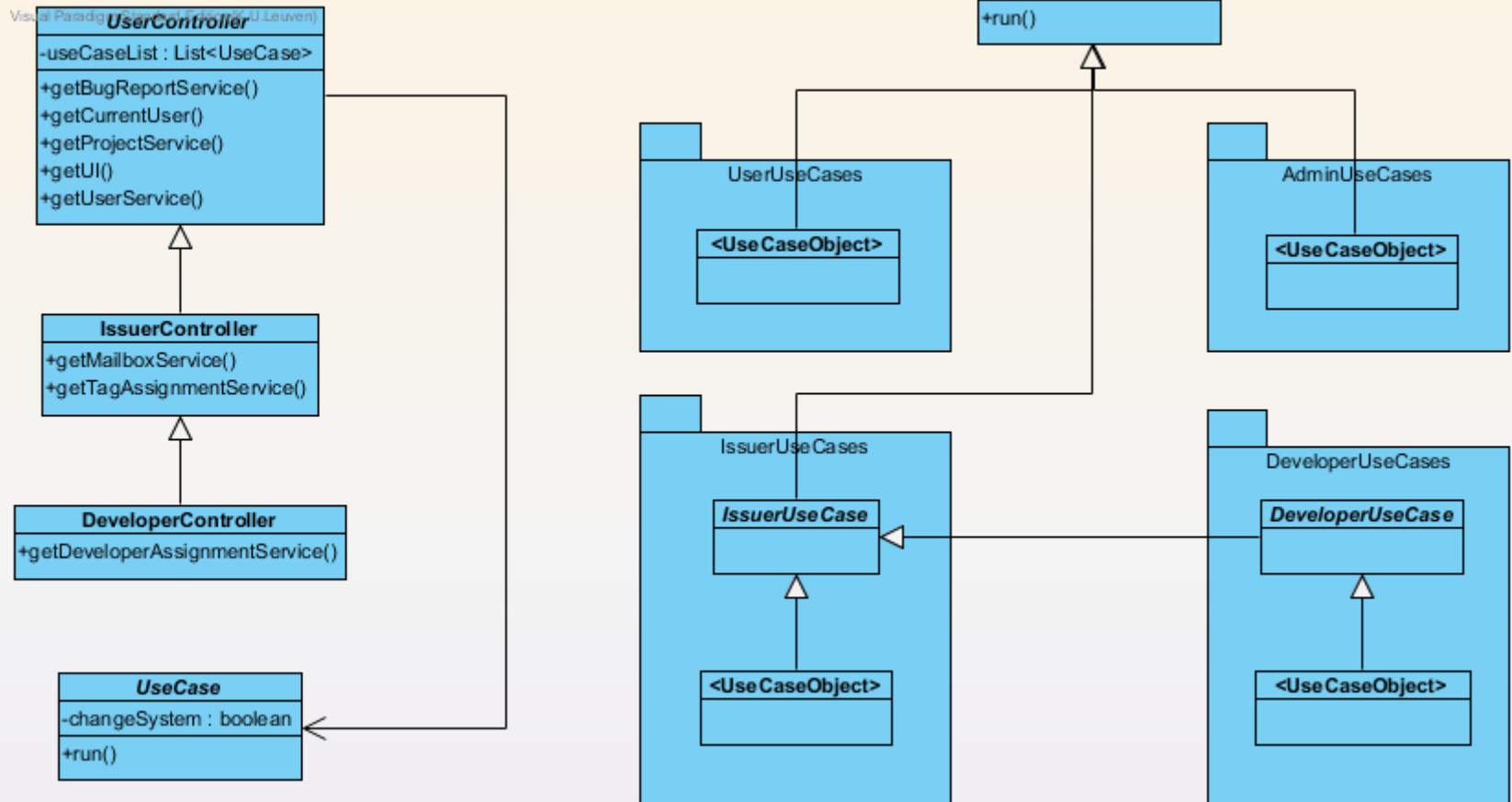
- Each use case in separate class, grouped by user type
- Extensions in domain layer: Mailbox, Milestone
- New classes added in similar fashion to iteration 1

Visual Paradigm Standard Edition (K.U. Leuven)

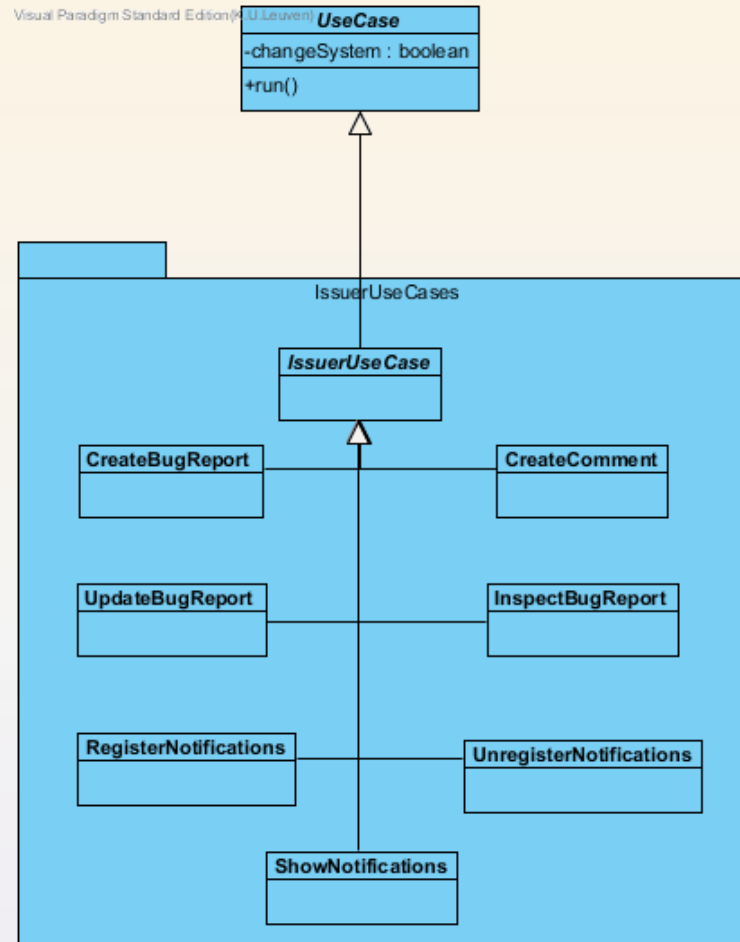


USE CASES

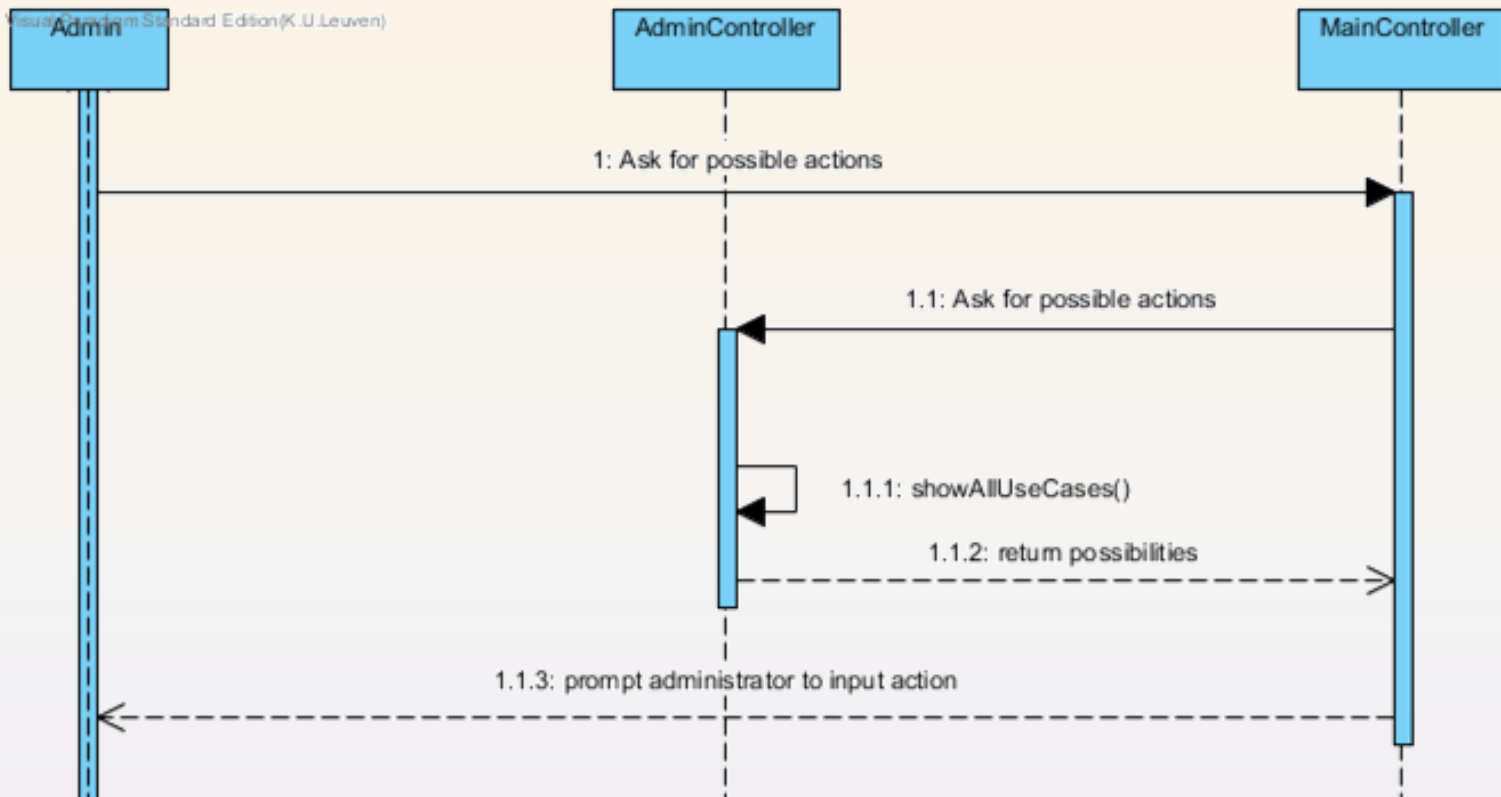
Visual Paradigm Standard Edition (K.U.Leuven)



USE CASES: EXAMPLE



USE CASES: EXAMPLE



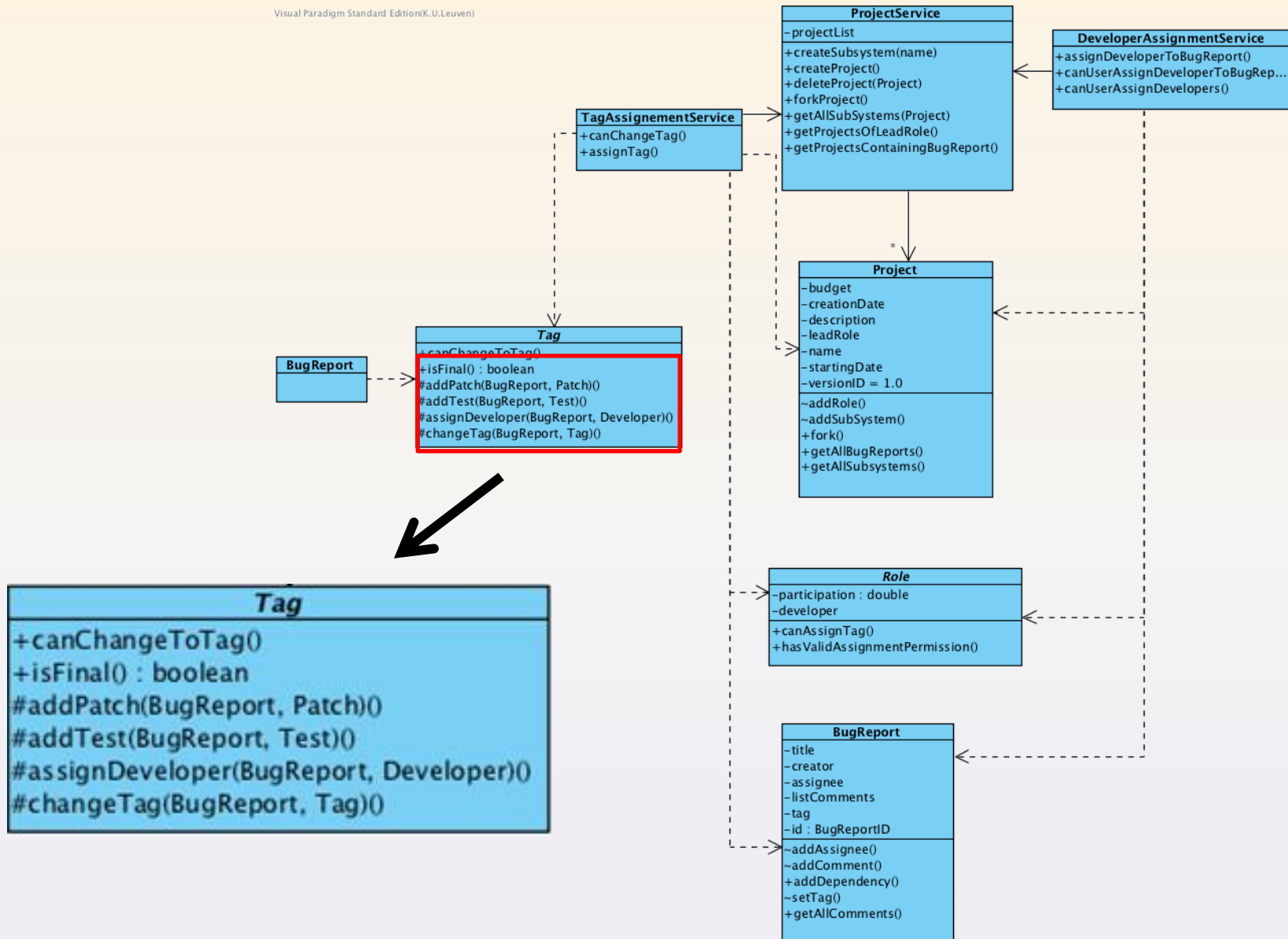
STATE PATTERN

Tags are STATES

The behaviour of a bugreport depends on the tag assigned
to it

USAGE OF THE DOMAIN LAYER

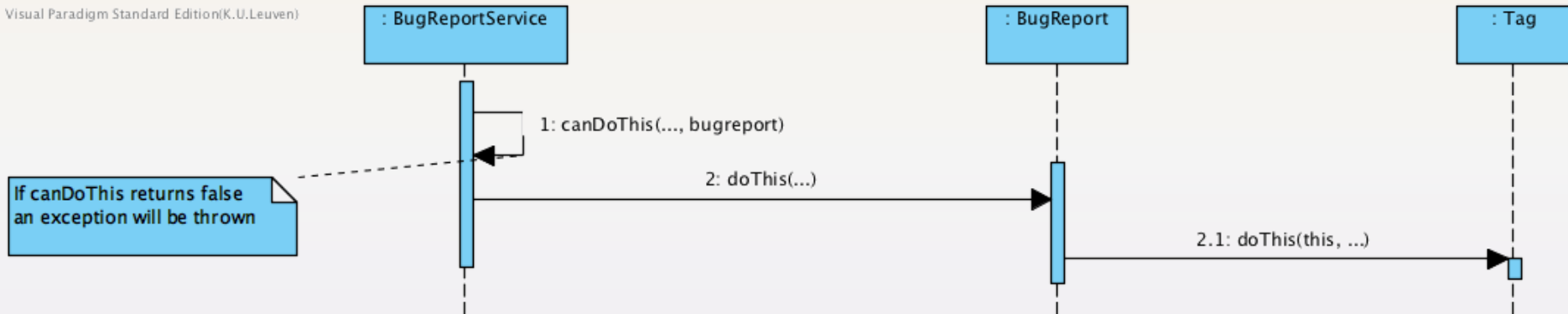
Visual Paradigm Standard Edition(K.U.Leuven)



STATE PATTERN: ABSTRACT

BugReport calls Tag to execute the function

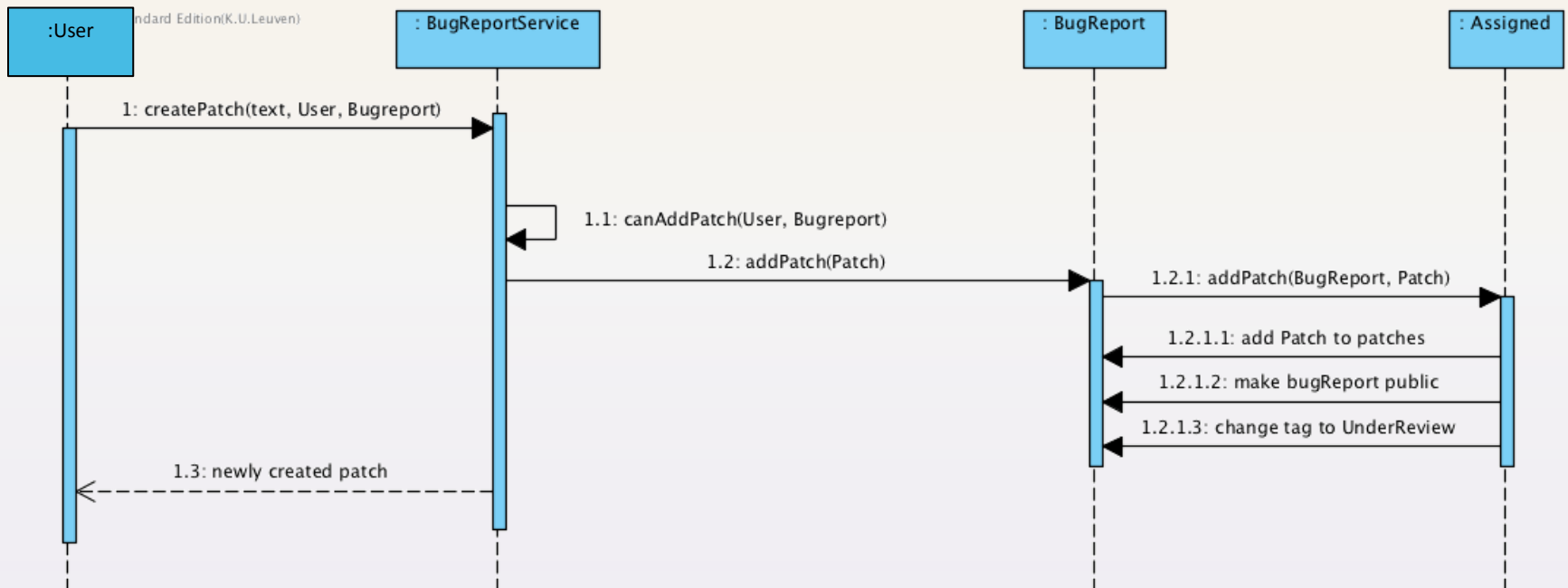
Visual Paradigm Standard Edition(K.U.Leuven)



STATE PATTERN: CONCRETE EXAMPLE

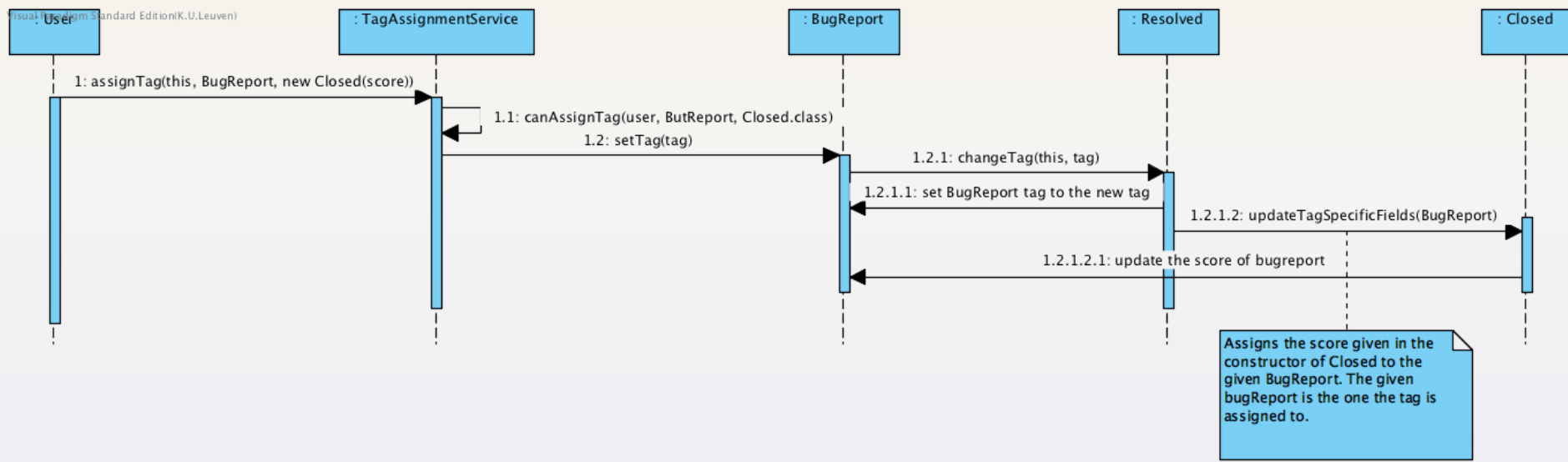
“[...] after the first patch, the tag of the bug report at stake transitions from Assigned to UnderReview.”

“When a bug report is created, it can be marked as private. [...] Once the first patch is submitted for the bug report, it is visible for all the users of BugTrap.”



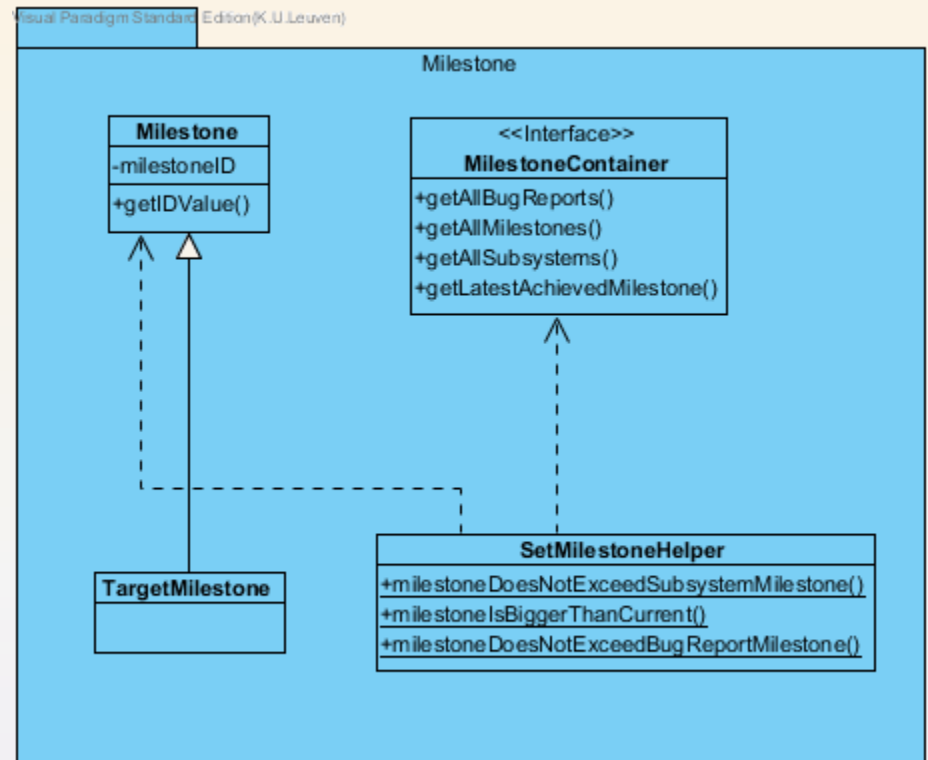
TAG SPECIFIC INFORMATION

“As a final step, the creator of a resolved bug report can close it (i.e. assign the tag Closed) by specifying how satisfied he is with the solution.”

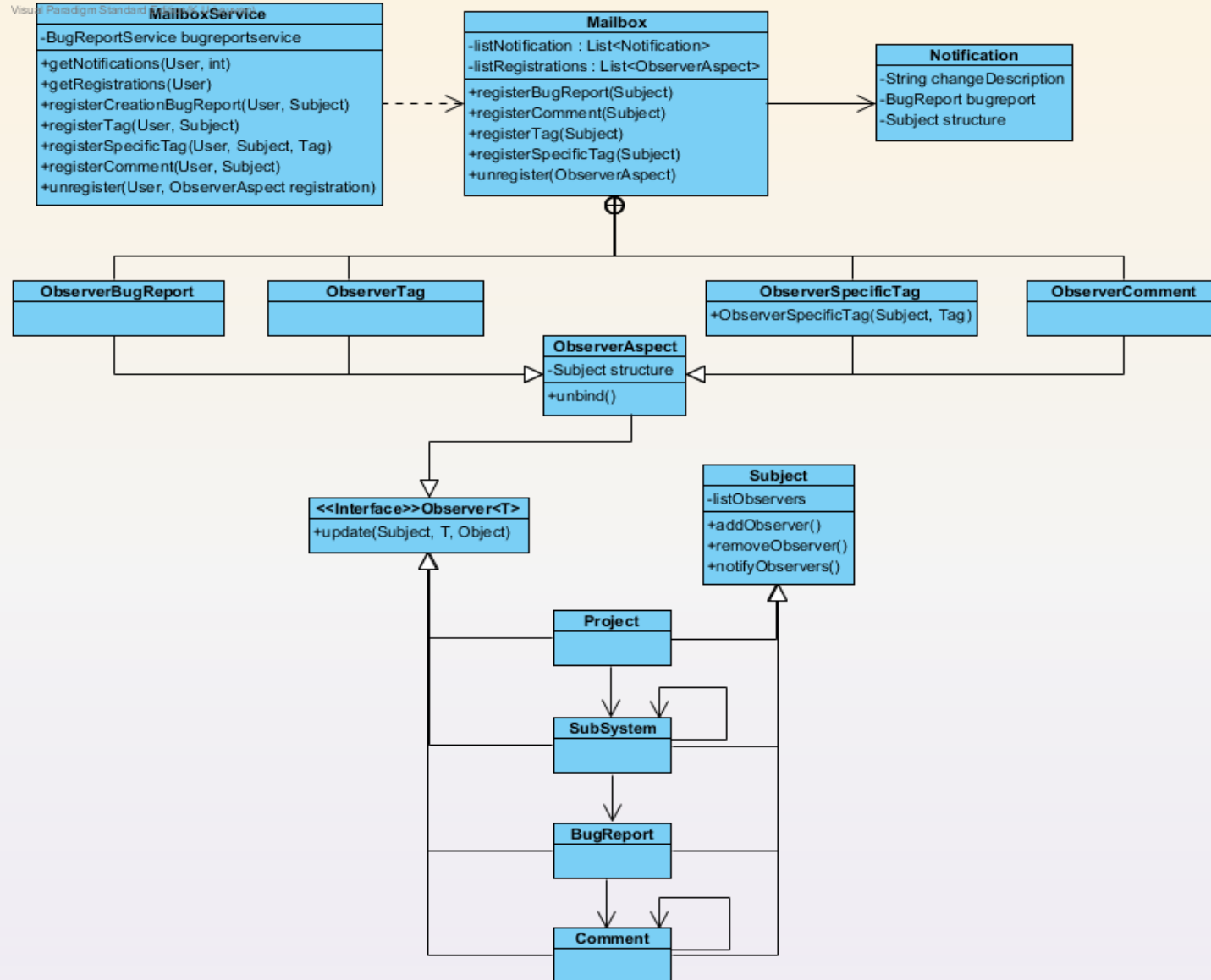


MILESTONES

- Milestone = separate object
Assigned to a project, subsystem or bug report
(Target milestone)
- Separate class for target milestone
→ optional, always initialized with a specific milestone (≠ “M0”)
- Container → needed for checkers
- Helper → provides checkers for milestone

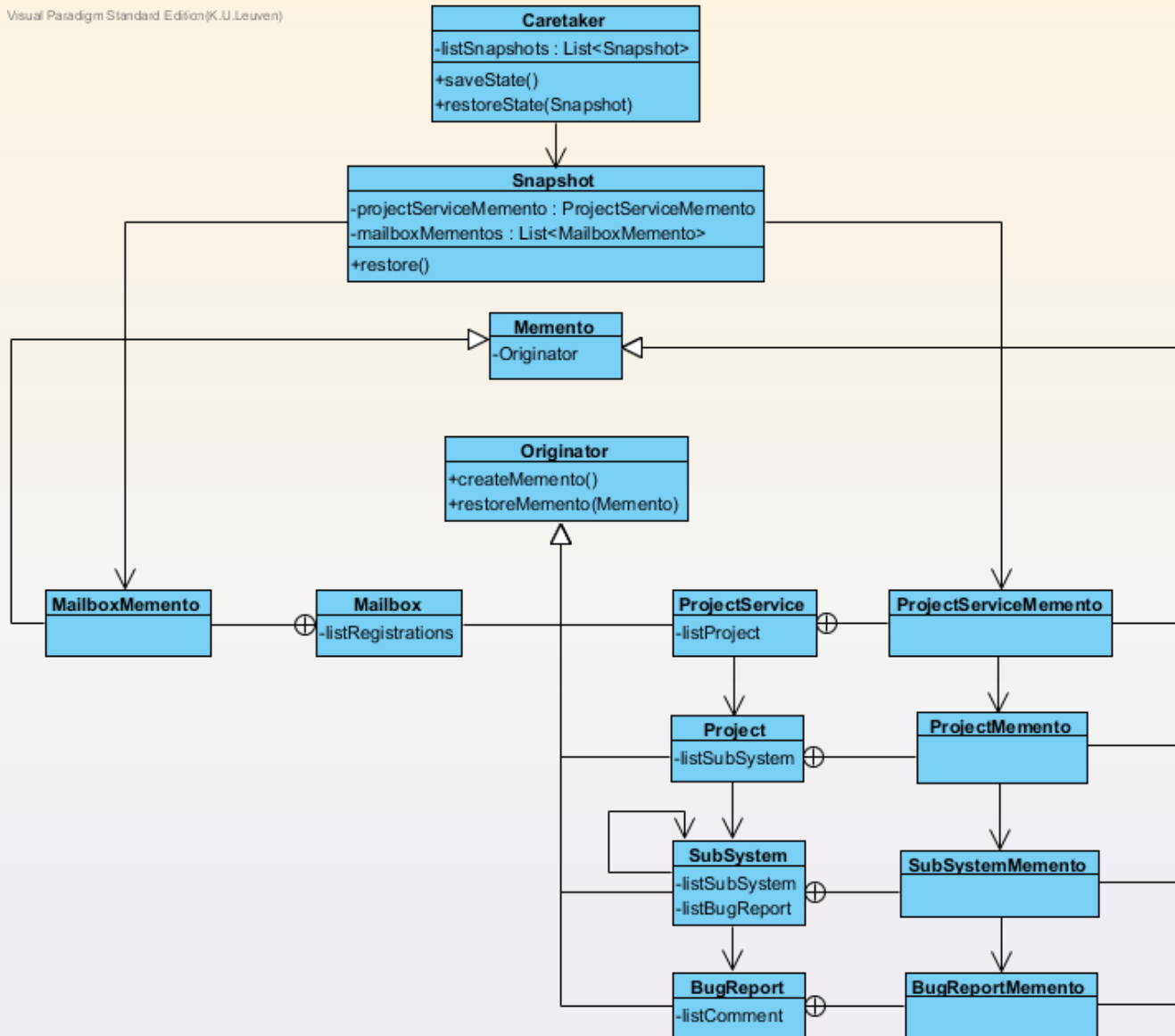


MAILBOX SYSTEM



UNDO MECHANISM

Visual Paradigm Standard Edition (K.U.Leuven)



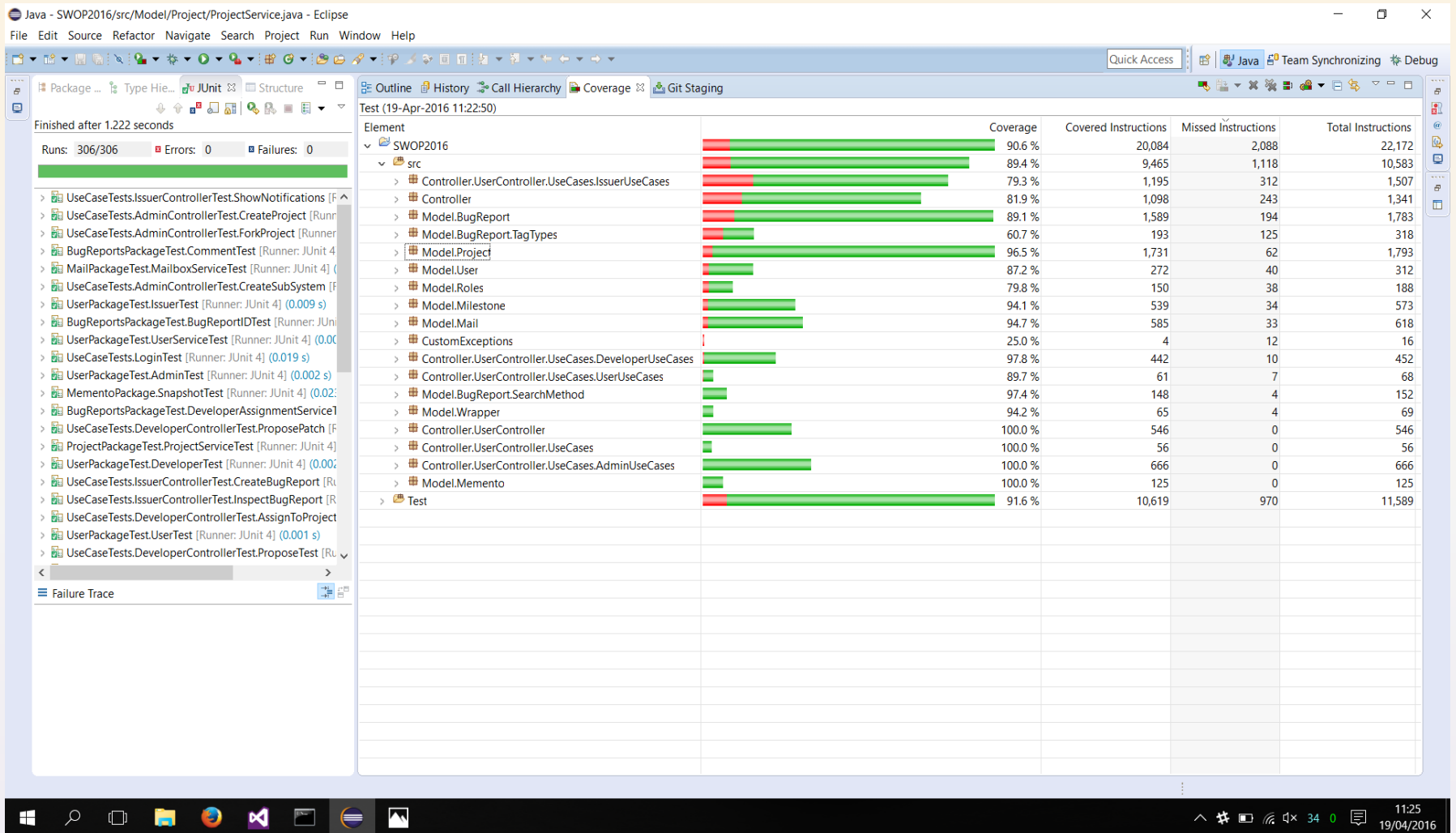
EXTENSIBILITY OF THE SYSTEM:

- Adding use cases
- UI is still completely independent
- Interface for the Initializer
- Adding constraints to a specific tag
- Adding new registration types (for mailboxes) = adding new observers
- Search methods for bug reports = strategy pattern
- Adding new objects that could be undone
- Little changes were needed to implement iteration 2 -> great extensibility of our code from iteration 1

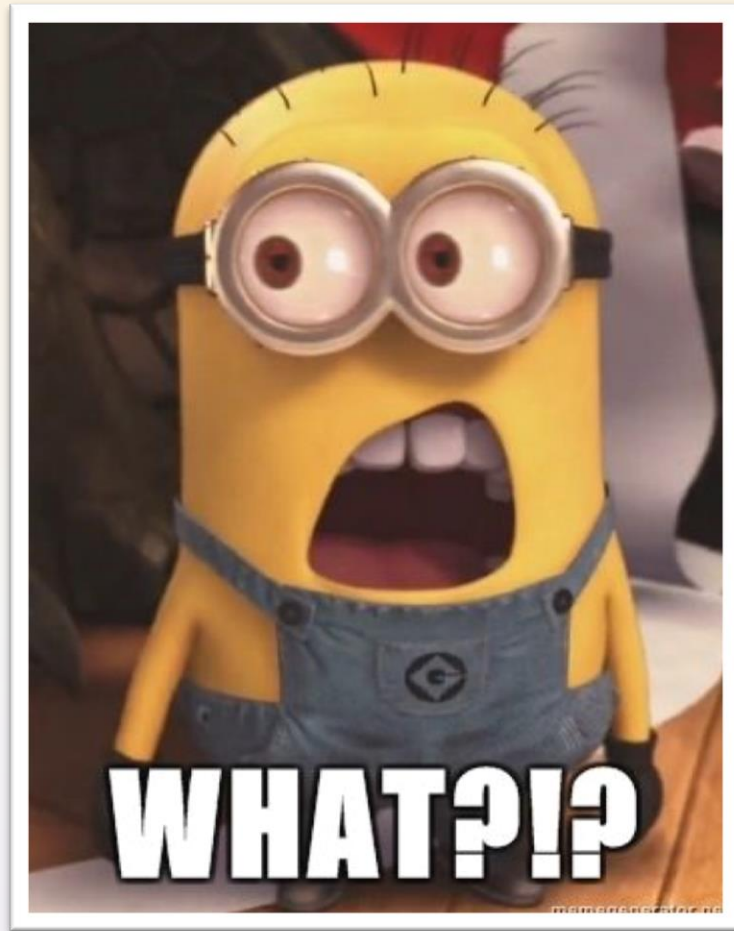
POSSIBLE IMPROVEMENTS

- Mailbox creates Observers, could be a task of the MailboxService
- Milestone: improvements in the code and structure

TESTING APPROACH



QUESTIONS?



PROJECT MANAGEMENT

KARINA KARAPETYAN

Group work: 31h Individual Work: 20h Study: 7h

LAURENS CLEEMPUT

Group work: 31h Individual Work: 23h Study: 6h

TOM HOUBEN

Group work: 31h Individual Work: 15h Study: 4h

TRI TRAN

Group work: 31h Individual Work: 51h Study: 14h

ROLES

	ITERATIE 1	ITERATIE 2	ITERATIE 3
DESIGN COORDINATOR	KARINA KARAPETYAN LAURENS CLEEMPUT	TOM HOUBEN KARINA KARAPETYAN	TRI TRAN TOM HOUBEN
TESTING COORDINATOR	TRI TRAN	LAURENS CLEEMPUT	KARINA KARAPETYAN
DOMAIN COORDINATOR	TOM HOUBEN	TRI TRAN	LAURENS CLEEMPUT