

► Upside

Better organization

Clear naming

Clear domain ;)

Easy to read

► Downside

Still has circular dependencies

Hard to maintain

Hard to evolve

MOVEMENTS NOT FEARED NEXT

 application

 src

 main

 java

controllers

☐ AllocationController.java

☐ StoryController.java

☐ TimeEntryController.java

dataaccess

☐ AllocationDAO.java

☐ StoryDAO.java

☐ TimeEntryDAO.java

models

☐ Allocation.java

☐ Story.java

☐ TimeEntry.java

☐ User.java

☐ App.java

FEATURE GROUPS (BUNDLED CONTEXTS)



story.java













Allocationn.java



AllocationController.java





Apprijava





controllers



data access



models





























































































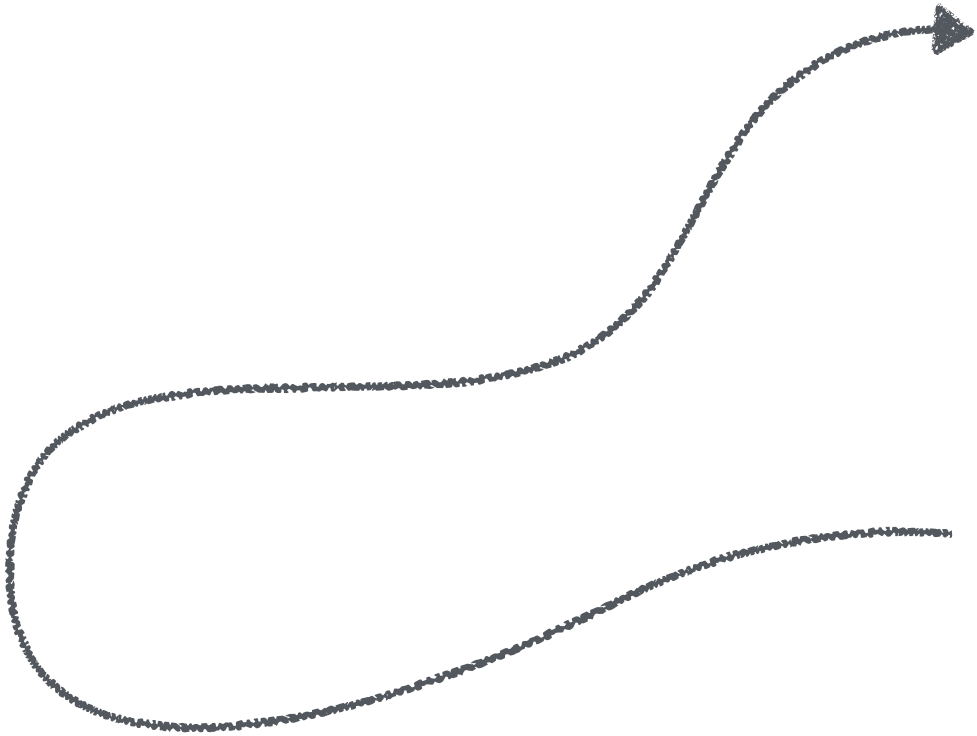




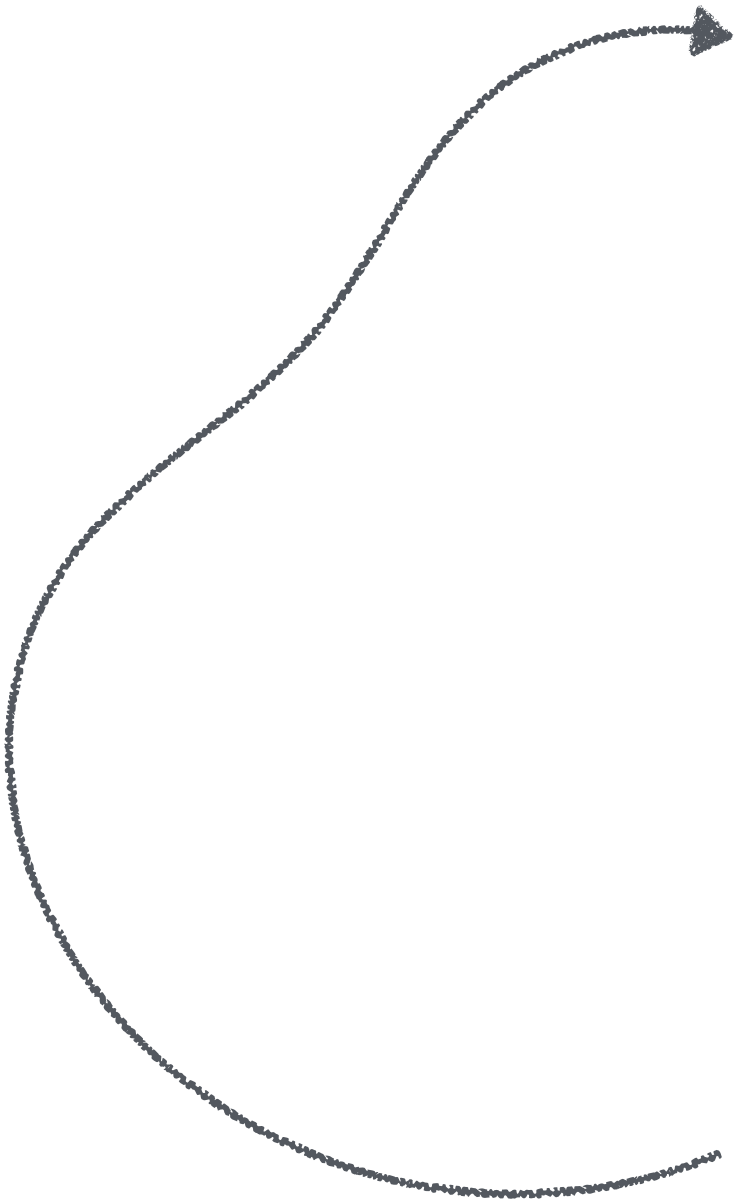




















































W







































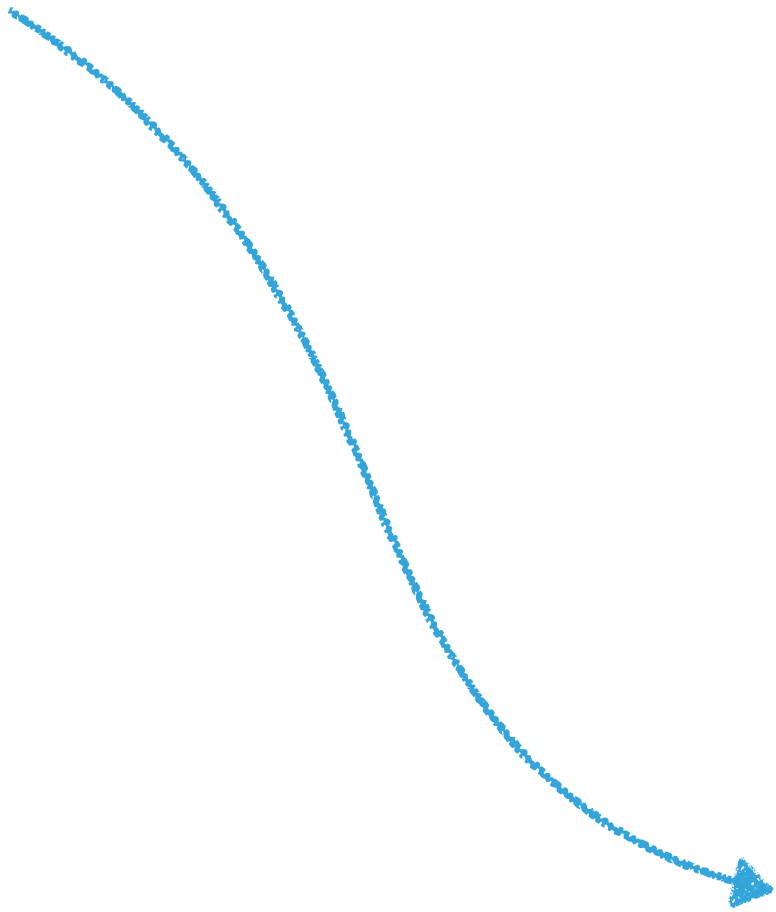


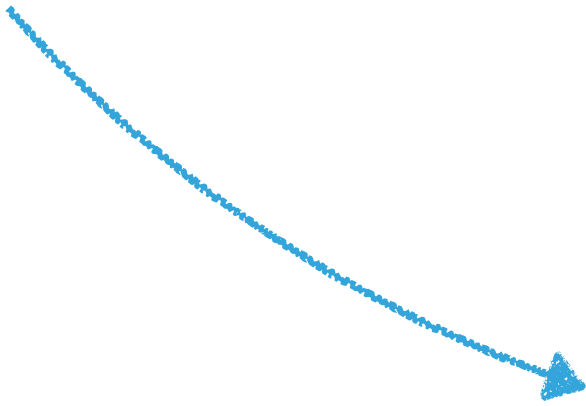


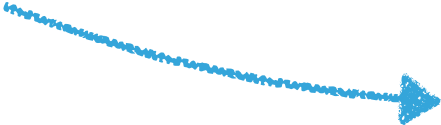




Better organization











all locations



backlog



timesheets

The compiler still allows circular dependencies ;)

has_many

belongs_to

FEATURE GROUPS (BOUNDED CONTEXTS)



► Upside

Better organization

Clear naming

Clear domain ;)

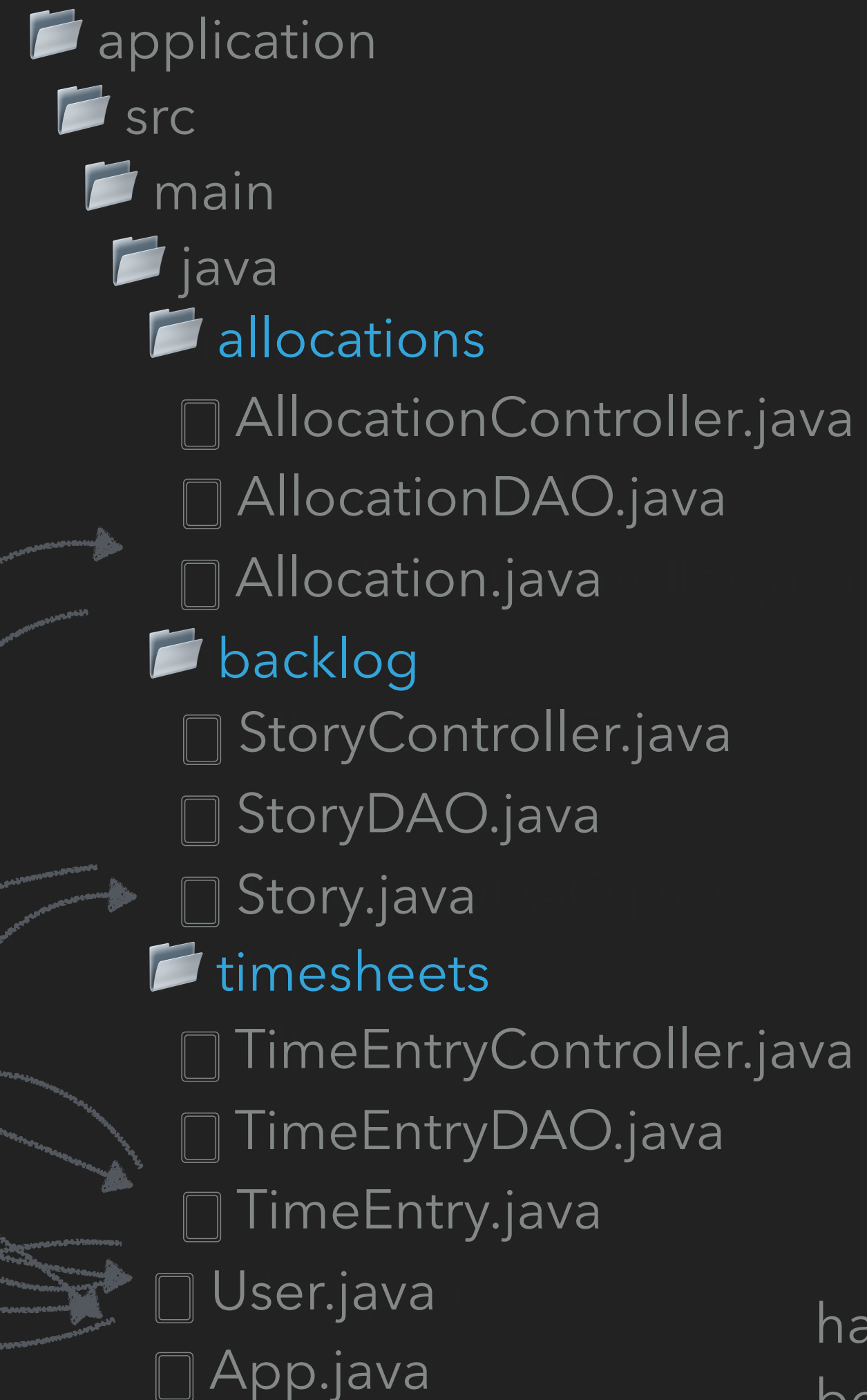
Easy to read

► Downside

Still has circular dependencies

Hard to maintain

Hard to evolve



The compiler still allows circular dependencies ;(

has_many
belongs_to

EACH COMPONENT IS INDIVIDUALLY BUILT AND TESTED

SINGLE APPLICATION AND COMPONENTS



- application
 - src
 - main
 - java
 - allocations
 - Allocation.java
 - AllocationController.java
 - AllocationDAO.java
 - backlog
 - Story.java
 - StoryController.java
 - StoryDAO.java
 - timesheets
 - TimeEntry.java
 - TimeEntryController.java
 - TimeEntryDAO.java
 - App.java