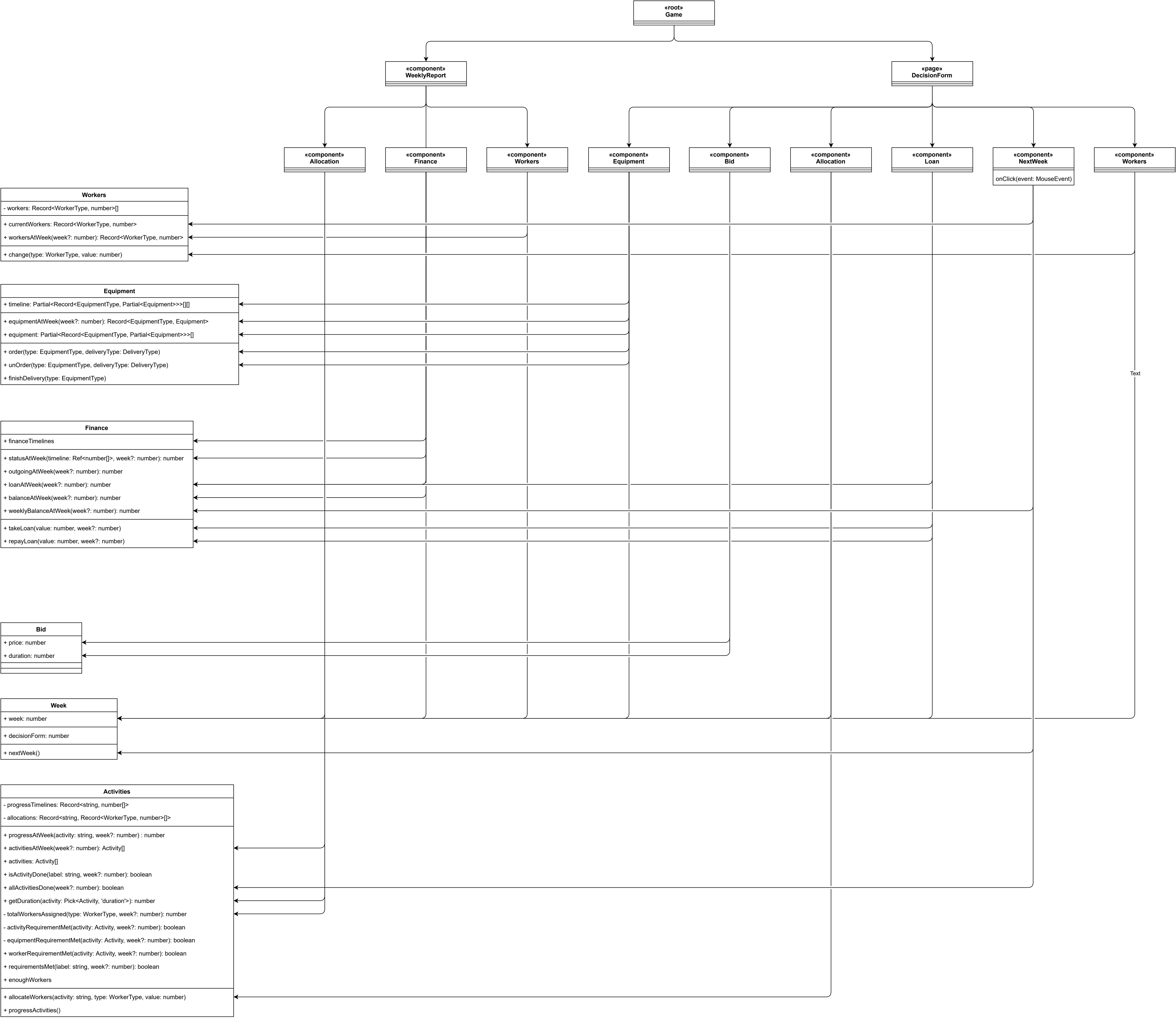


nt»



Workers
labour: number
skilled: number
electrician: number
change(key: WorkerType, difference: number)

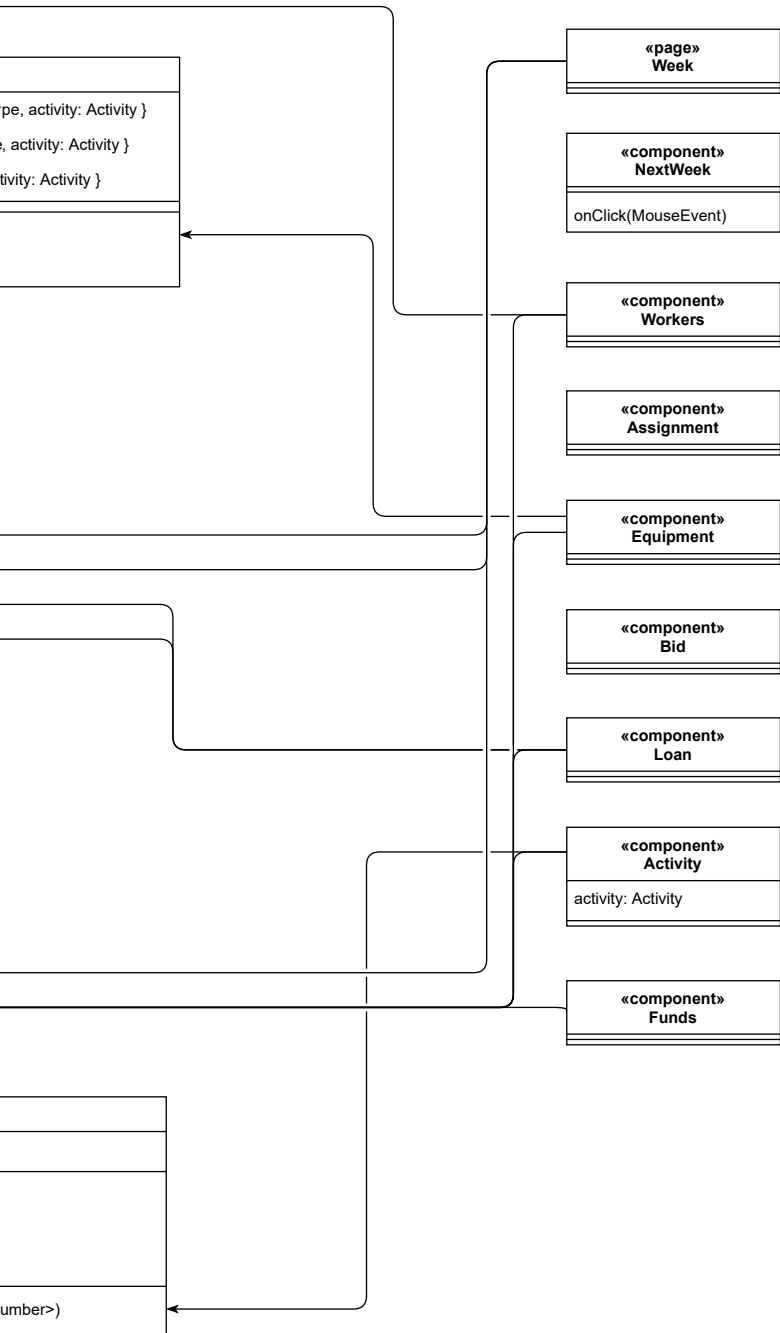
Equipment
steelwork: { status: EquipmentStatus, deliveryType: DeliveryTy
Interior: { status: EquipmentStatus, deliveryType: DeliveryType
tbs: { status: EquipmentStatus, deliveryType: DeliveryType, ac
order(key: EquipmentType, deliveryType: DeliveryType)
finishDelivery(key: EquipmentType)

Finance
incoming: number
outgoing: number
loan: number
balance(): number
addIncoming(value: number)
addOutgoing(value: number)
takeLoan(value: number)
repayLoan(value: number)

Bid
cost: number
time: number

Week
week: number
weekListeners: Function[]
nextWeek()
addWeekListener(listener: Function)
removeWeekListener(listener: Function)

Activities
activities: Activity[]
isActivityDone(label: string): boolean
allActivitiesDone(): boolean
requirementsMet(label: string): boolean
allocateWorkers(label: string, workers: Record<WorkerType, n



«component»
Workers

«component»
Equipment

«component»
Funds

«com
L

«component»
loan

«component»
Bid

«component»
Allocation

«root»
App

«component»
NextWeek

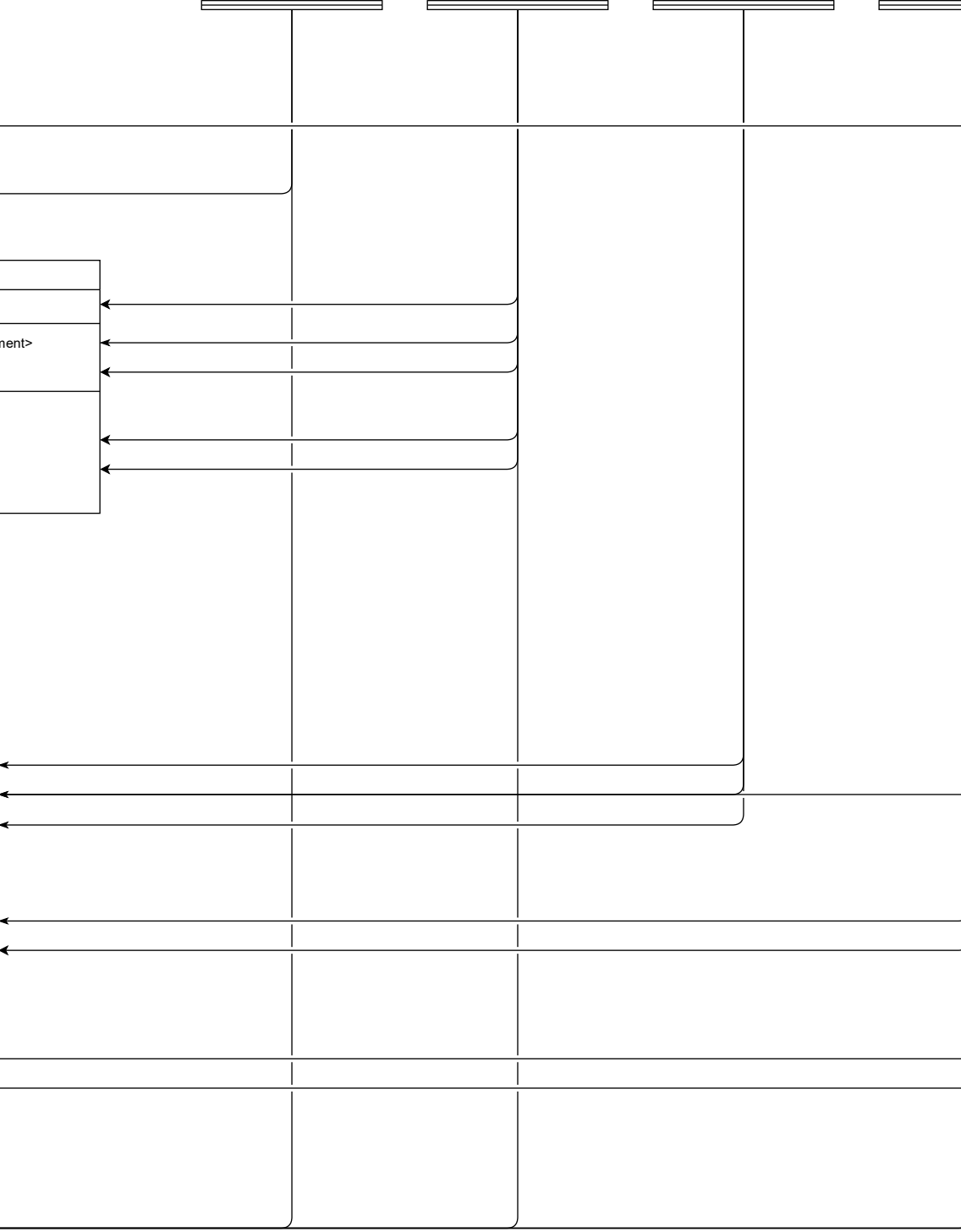
Workers
- workers: Record<WorkerType, number>[]
+ currentWorkers: Record<WorkerType, number>
+ workersAtWeek(week?: number): Record<WorkerType, number>
+ change(type: WorkerType, value: number)

Equipment
+ timeline: Partial<Record<EquipmentType, Partial<Equipment>>>>[]
+ equipmentAtWeek(week?: number): Record<EquipmentType, Equipment>
+ equipment: Partial<Record<EquipmentType, Partial<Equipment>>>>[]
- populateTimeline()
+ order(type: EquipmentType, deliveryType: DeliveryType)
+ unOrder(type: EquipmentType, deliveryType: DeliveryType)
+ finishDelivery(type: EquipmentType)

Finance
- incomingTimeline: number[]
- outgoingTimeline: number[]
- loanTimeline: number[]
+ statusAtWeek(timeline: Ref<number[]>, week?: number): number
+ incoming: number
+ outgoing: number
+ loan: number
+ balance: number
+ addIncoming(value: number, week?: number)
+ addOutgoing(value: number, week?: number)
+ takeLoan(value: number, week?: number)
+ repayLoan(value: number, week?: number)

Bid
+ price: number
+ duration: number

Week
+ week: number



onClick(event: MouseEvent)

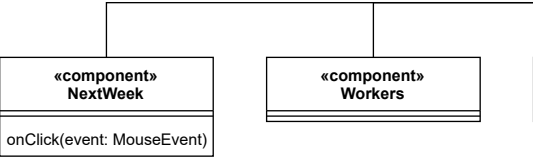
Category		Item		Status	
Category 1	Item 1.1	Item 1.2		Status 1.1	
	Item 1.1	Item 1.2		Status 1.1	
	Item 1.1	Item 1.2		Status 1.1	
	Item 1.1	Item 1.2		Status 1.1	
Category 2	Item 2.1	Item 2.2		Status 2.1	
	Item 2.1	Item 2.2		Status 2.1	
	Item 2.1	Item 2.2		Status 2.1	
	Item 2.1	Item 2.2		Status 2.1	
Category 3	Item 3.1	Item 3.2		Status 3.1	
	Item 3.1	Item 3.2		Status 3.1	
	Item 3.1	Item 3.2		Status 3.1	
	Item 3.1	Item 3.2		Status 3.1	
Category 4	Item 4.1	Item 4.2		Status 4.1	
	Item 4.1	Item 4.2		Status 4.1	
	Item 4.1	Item 4.2		Status 4.1	
	Item 4.1	Item 4.2		Status 4.1	
Category 5	Item 5.1	Item 5.2		Status 5.1	
	Item 5.1	Item 5.2		Status 5.1	
	Item 5.1	Item 5.2		Status 5.1	
	Item 5.1	Item 5.2		Status 5.1	
Category 6	Item 6.1	Item 6.2		Status 6.1	
	Item 6.1	Item 6.2		Status 6.1	
	Item 6.1	Item 6.2		Status 6.1	
	Item 6.1	Item 6.2		Status 6.1	
Category 7	Item 7.1	Item 7.2		Status 7.1	
	Item 7.1	Item 7.2		Status 7.1	
	Item 7.1	Item 7.2		Status 7.1	
	Item 7.1	Item 7.2		Status 7.1	
Category 8	Item 8.1	Item 8.2		Status 8.1	
	Item 8.1	Item 8.2		Status 8.1	
	Item 8.1	Item 8.2		Status 8.1	
	Item 8.1	Item 8.2		Status 8.1	
Category 9	Item 9.1	Item 9.2		Status 9.1	
	Item 9.1	Item 9.2		Status 9.1	
	Item 9.1	Item 9.2		Status 9.1	
	Item 9.1	Item 9.2		Status 9.1	
Category 10	Item 10.1	Item 10.2		Status 10.1	
	Item 10.1	Item 10.2		Status 10.1	
	Item 10.1	Item 10.2		Status 10.1	
	Item 10.1	Item 10.2		Status 10.1	

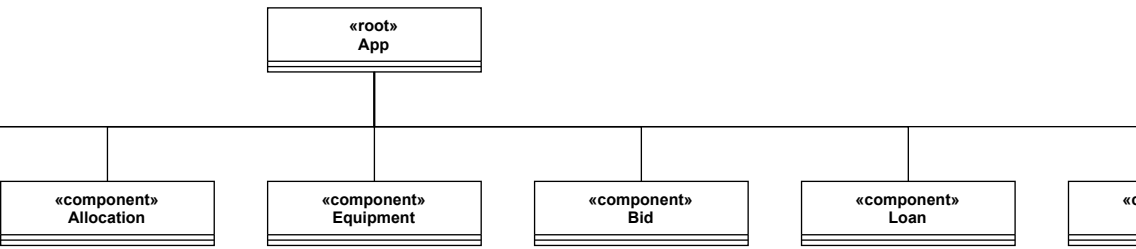
+ decisionForm: number	←
+ nextWeek()	←

Activities
- progressTimelines: Record<string, number[]> - allocations: Record<string, Record<WorkerType, number>[]>
+ progressAtWeek(activity: string, week?: number) : number + activitiesAtWeek(week?: number): Activity[] + activities: Activity[] + isActivityDone(label: string, week?: number): boolean + allActivitiesDone(week?: number): boolean + getDuration(activity: Pick<Activity, 'duration'>): number - totalWorkersAssigned(type: WorkerType, week?: number): number - activityRequirementMet(activity: Activity, week?: number): boolean - equipmentRequirementMet(activity: Activity, week?: number): boolean + workerRequirementMet(activity: Activity, week?: number): boolean + requirementsMet(label: string, week?: number): boolean
+ allocateWorkers(activity: string, type: WorkerType, value: number) + progressActivities()

The diagram illustrates a vertical bar divided into four sections by three horizontal lines. Three horizontal arrows point from the right towards the boundaries between these sections. The top section is white, the second is light gray, the third is white, and the bottom is light gray.

[illegible]





component»
Funds

StoreName
- state: type
+ getter1: returnType
+ getter2(argument: type): returnType
+ action(argument: type): returnType

«type» Name
prop: Type
event(argument: Type)

Bid:	Current week: 0
Expected cost: 1000000 £	Decision form: 1
Expected time: 8	

Hire or fire workers		Order Equipment	
Worker	Amount	Equipment	Order
Labour	10	Steelwork (Task A)	2
Skilled		Interior (Task B)	2
Electrician	2	TBS (Task C)	2
Hint: Positive = hire, negative = fire.		Hint: '1' for regular or '2' for express.	

Activity	Allocate workers to activities		
	LAB	SKI	ELE
A			
B			
C			
D			
E			
F			
G			
H			
I			
J			
K			
L			

Hint: If a category is highlited 'red', then you've allocated more workers than available on site.
Hint: Remember to allocate workers in every 'Decision Form'.
Hint: Remember to allocate a sufficient amount of workers for the task to progress.

Bank loan	
New loan:	160000
Repay amount:	
Hint: If you wish to repay the entire remaining loan; write: 100%	

Current money: 0 £
Current money spent: 0 £

NEXT WEEK

