Team Activity
OOPT - A New OOO Digital Watch

A New OOO Digital Watch

- Supposed to develop a new OOO digital watch.

- Let's analyze and design your own new OOO digital watch.
  - OOAD development method : OOPT
  - Use a UML tool
    - Not use Communication, Activity, Package, Deployment Diagrams, for now.
  - Basic Requirements & Assumptions :
    - A set of predefined/fixed hardware (1 LCD, 4 buttons, 1 buzzer, 1 SW controller)
    - Dynamic SW Configuration (4 activated in 6 functions)
    - 4 alarms
  - Instructions
    - Take care of the layered architecture of your system under development.
    - Take care of your system context - embedded system!
    - Make every assumptions clear, feasible and consistent.

- **Team activities:**
  1. Stage 1000 : Plan
  2. Stage 2000 > 2030 : Analyze
  3. Stage 2000 > 2040 : Design
  4. Stage 2000 > 2050 : Implementation

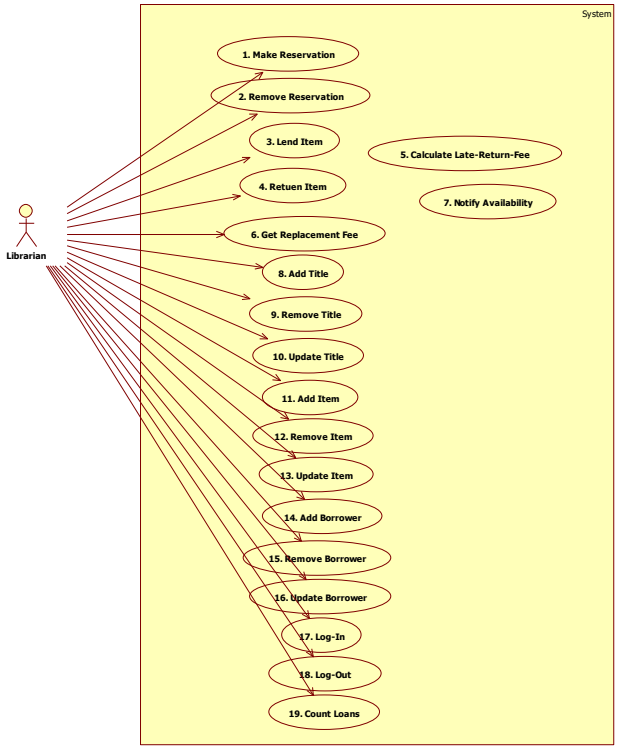
[Team Activity #1] Planning

Functional Requirements	Use Cases	Category
R1.1 Make reservation	1. Make Reservation	Evident
R1.2 Remove reservation	2. Remove Reservation	Evident
R1.3 Lend Item	3. Lend Item	Evident
R1.4.1 Return title	4. Return Title	Evident
R1.4.2 Calculate Late-Return-Fee	5. Calculate Late-Return-Fee	Hidden
R1.5 Calculate Replacement Fee	6. Get Replacement Fee	Evident
R1.6 Notify Availability	7. Notify Availability	Hidden
R2.1 Add title	8. Add Title	Evident
R2.2 Remove title	9. Remove Title	Evident
R2.3 Update title	10. Update Title	Evident
R2.4 Add items	11. Add Item	Evident
R2.5 Remove item	12. Remove Item	Evident
R2.6 Update item	13. Update Item	Evident
R3.1 Add borrower	14. Add Borrower	Evident
R3.2 Remove borrower	15. Remove Borrower	Evident
R3.3 Update borrower	16. Update Borrower	Evident
R4.1 Validates system access	17. Log-IN	Evident
R4.2 Validates system access	18. Log-Out	Evident
R5.1 Compute total # of items checked out	19. Count Loans	Evident

Functional Requirements ≈ Use Case

Use Case Description (Brief)

Use Case	1. Make Reservation
Actors	Librarian
Description	- This use case begins when a borrower arrives at the counter and then requests reservation. - For a registered borrower, it makes a reservation slip (software-wise). - For an unregistered borrower, the librarian registers the person and makes a reservation for the person.



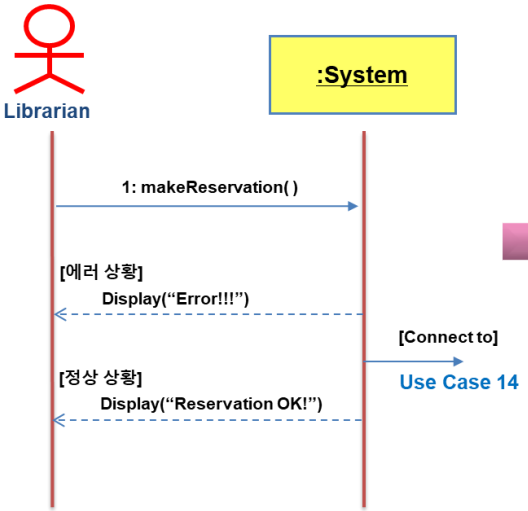
Use Case Diagram

[Team Activity #2] Object-Oriented Analysis

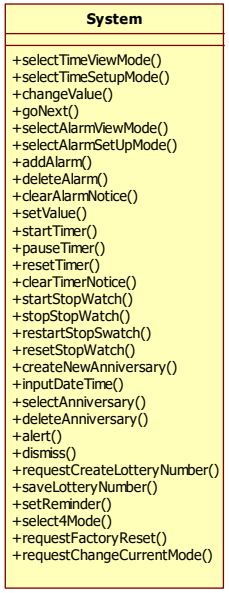
USE CASE: 1. Make Reservation

1. (A) A librarian requests the reservation of title.
2. (S) Check if corresponding title exist.
3. (S) Check if corresponding borrower exist.
4. (S) If the borrower does not exist, invoke "Add Borrower". (→ connect to other Use Case)
5. (S) Create reservation information.

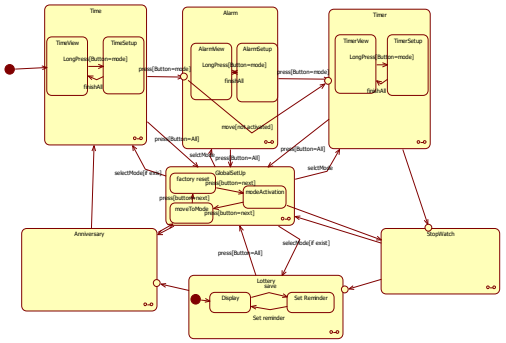
Use Cases
(Brief/Casual)



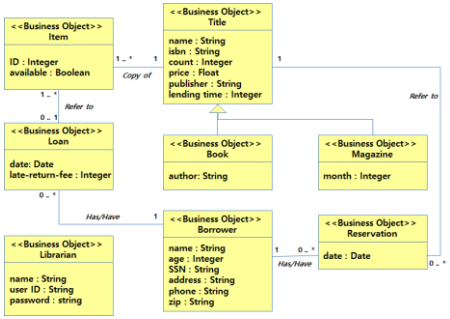
System Sequence Diagrams



System Operations
(System Interface)

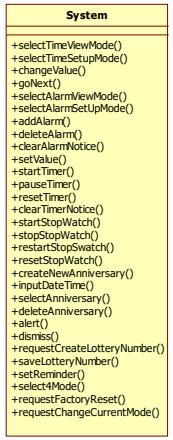


Statechart Diagram



Domain Model

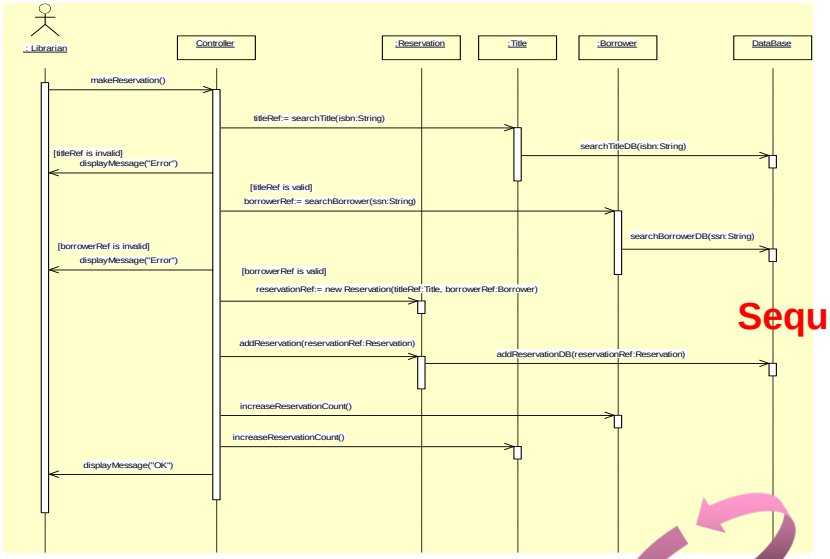
[Team Activity #3] Object-Oriented Design



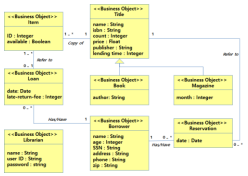
System Operations

Use Case	1. Make Reservation
Actor	Librarian
Purpose	Create a new reservation
Overview	(As in the business use case)
Type	Primary and Real
Cross Reference	System Functions: R1.1, R3.1 Use Case: "Add Borrower"
Pre-Requisites	A borrower should be registered.
Typical Courses of Events	(A) : Actor, (S) : System 1. (A) A librarian inputs an isbn and ssn of the title 2. (S) Find a corresponding title 3. (S) Find a corresponding borrower 4. (S) Create a new reservation 5. (S) Store the new reservation 6. (S) Increase reservationCount in the borrower 7. (S) Increase reservationCount in the title
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 2: If the title does not exist, display an error message. Line 3: If the borrower does not exist, display an error message.

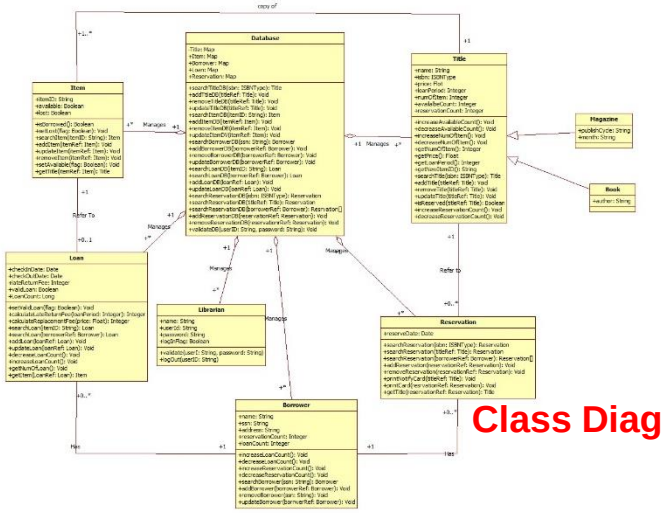
Use Cases
(Fully dressed-up)



Sequence Diagrams

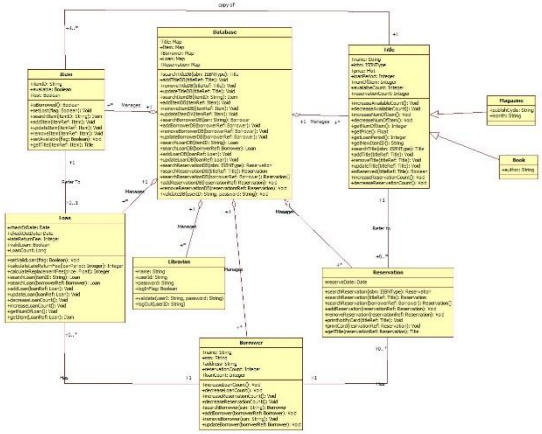


Domain Model



Class Diagram

[Team Activity #4] Object-Oriented Implementation



Class Diagram
(Static Model)

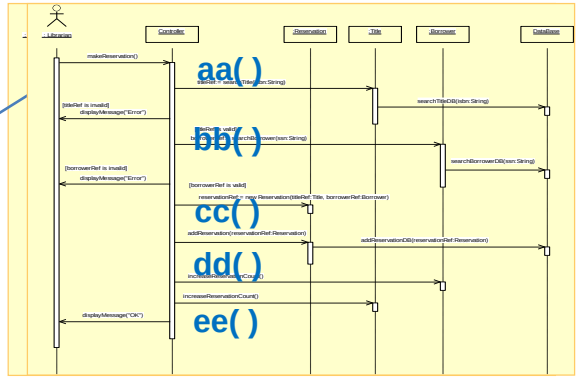
Code Generation

Skeleton Code

```
class  A
{
  variables:

  aa();
  bb();
…
}
```

System Operation



Sequence Diagrams
(Dynamic Model)

Realization each method for system operation !!!

```
class  A
{
  variables:

  aaa()
  {
     aa();
     bb();
     cc();
  }
  bbb();
…
}
```

+ 변수 선언
+ 컨디션문
+ 초기화
+ Visibility 확보

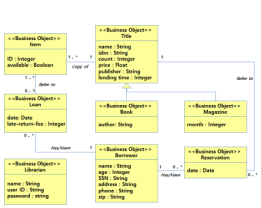
[Team Activity #5] Discussion

- 현업에서는 SW를 어떻게 개발하나요?

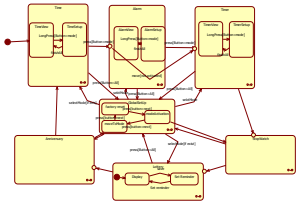
- 현업에서 이 기법을 적용해서 좋은 성과를 얻기 위해서는 어떻게 해야 할까요?
  - SW개발방법론인 OOPT (UP)를 적용해 본 소감을 자유롭게 논의해 봅니다.
    - OOAD 개발방법론의 장·단점을 논의하세요.
    - 어느 상황에서 가장 효과적으로 적용 가능할까요?
    - 어느 상황에서는 오히려 역효과가 발생할까요?
  - SW개발방법론인 OOPT (UP)에서 어떤 내용들이 수정·보완되면 더 효과적으로 적용이 가능할까요?

- 처음부터 개발하는 것이 아니라, 특정 부분만 개발하거나 또는 open source / 구글링 등 기존 코드로부터 개발을 시작하는 경우에는, OOPT(UP)를 어떻게 수정·확장하면 좋을까요?

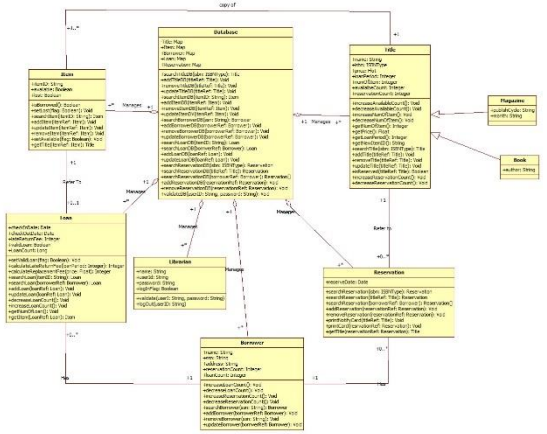
OOPT of OOAD, in Summary



Domain Model



Statechart Diagram

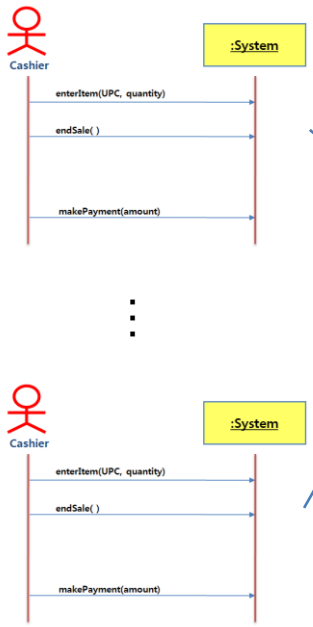


Class Diagram

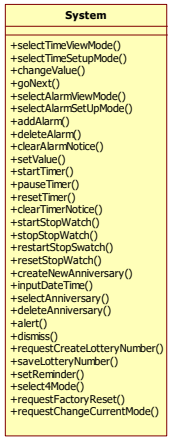
Ref. #	Function	Use Case Number & Name	Category	Category
R1.1	Make reservation	1. Make Reservation	Primary	Evident
R1.2	Remove reservation	2. Remove Reservation	Primary	Evident
R1.3	Lend Item	3. Lend Item	Primary	Evident
R1.4.1	Return title	4. Return Title	Primary	Evident
R1.4.2	Calculate Late-Return-Fee	5. Calculate Late-Return-Fee	Primary	Hidden
R1.5	Calculate Replacement Fee	6. Get Replacement Fee	Primary	Evident
R1.6	Notify Availability	7. Notify Availability	Primary	Hidden
R2.1	Add title	8. Add Title	Primary	Evident
R2.2	Remove title	9. Remove Title	Primary	Evident
R2.3	Update title	10. Update Title	Primary	Evident
R2.4	Add items	11. Add Item	Primary	Evident
R2.5	Remove item	12. Remove Item	Primary	Evident
R2.6	Update item	13. Update Item	Primary	Evident
R3.1	Add borrower	14. Add Borrower	Primary	Evident
R3.2	Remove borrower	15. Remove Borrower	Primary	Evident
R3.3	Update borrower	16. Update Borrower	Primary	Evident
R4.1	Validates system access	17. Log-IN	Secondary	Evident
R4.2	Validates system access	18. Log-Out	Secondary	Evident
R5.1	Compute total # of items checked out	19. Count Loans	Secondary	Evident

User (Functional) Requirements

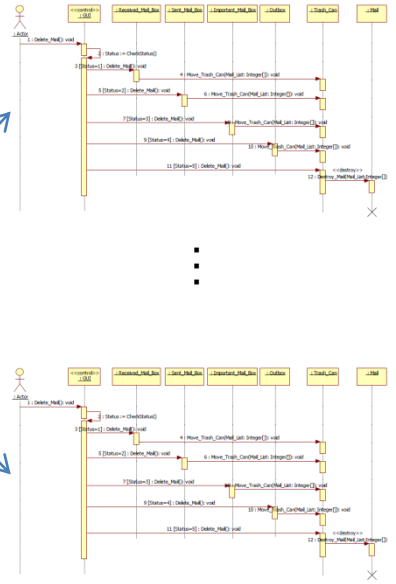
Use Cases



System Sequence Diagrams



System Operations



Sequence Diagrams

Stage 1000. Plan

Stage 2030. Analyze

Stage 2040. Design

Traceability Analysis (Example)

Essential UseCase	S-Link
	S1
	S2, S3, S4
	S5, S4
	S6, S3, S4
	S7
	S8
	S16
	S9
	S10
	S11, S3, S4
	S12
	S13
	S14
	S17
	S15
	S4.1
	S4.2
	S4.3
	S4.4
	S5.1, S5.2
	S5.2, S5.3
	S5.3, S5.4
	S5.5
	S6.1
	S6.2, S6.3
	S5, S4
	S7.1
	S7.2
현새 보느 면완	S7.3

SID	Operation in Sequence Diagram	M-Link
S1	selectTimeViewMode	M15,M1
S2	selectTimeSetupMode	M16,M2,M12,M5
S3	changeValue	M13,M12
S4	goNext	M14,M12
S5	selectAlarmViewMode	M18,M3,M14
S6	selectAlarmSetupMode	M17,M4,M5,M12
S7	addAlarm	M20,M17,M12,M3
S8	deleteAlarm	M19,M3
S9	clearAlarmNotice	M21,M11,M6
S10	selectTimerViewMode	M26,M7
S11	selectTimerSetupMode	M27,M8,M5,M12
S12	startTimer	M25,M7
S13	pauseTimer	M22,M7
S14	resetTimer	M23,M7
S15	clearTimerNotice	M24,M7
S16	alarmBeep	M31,M3
S17	timerBeep	M31,M7
S4.1	startStopWatch()	M4.1, M4.2, M4.3, M4.4, M4.5
S4.2	stopStopWatch()	M4.6, M4,7
S4.3	restartStopWatch()	M4.2, M4.3, M4.4, M4.5, M4.8
S4.4	resetStopWatch()	M4.5, M4.9, M4.10
S5.1	createNewAnniversary()	M5.1□M5.2□
S5.2	inputDateTime()	□M5.3□M5.4□M5.5□M5.6□M5.7□M5.8□
S5.3	selectAnniversary()	M5.9, M5.2
S5.4	deleteAnniversary()	M5.10, M5.11, M5.12
S5.5	dismiss()	M5.13, M5.14, M5.15
S6.1	requestCreateLotteryNumber	M6.1,M6.6, □M6.7, M6.10, M6.11
S6.2	saveLotteryNumber	M6.5
S6.3	setReminder	M6.6
S7.1	select4Mode	M6.2, M6.3, M6.
S7.2	requestFactoryReset	M6.9
S7.3	requestChangeCurrentMode	M6.13

MID	Method	Class
M1	displayCurrentTime	DisplayManager
M2	displaySetupMode	
M3	displayAlarm	
M4	displayNextAlarm	
M5	blinkSetupItem	
M6	displayCurrentMode	
M7	displayTimer	
M8	displaySetupMode	
M9	viewMode	Mode
M10	setupMode	
M11	getPreviousMode	
M12	saveValue	
M13	changeValue	
M14	goNext	
M15	selectTimerViewMode	TimeMode
M16	selectTimeSetupMode	
M17	selectAlarmSetupMode	AlarmMode
M18	selectAlarmViewMode	
M19	deleteCurrentAlarm	
M20	addNewAlarm	
M21	clearAlarm	
M22	pauseTimerVlaue	TimerMode
M23	resetTimerValue	
M24	clearTimer	
M25	runTimer	
M26	selectTimerViewMode	
M27	selectTimerSetupMode	
M28	registerTickObserver	TimeManager
M29	setTime	
M30	tick	TickObserver
M31	beep	BeepManager
M32	(Input Event 생성)	InputProcessor

M4.1	startStopWatch()	StopWatchMode
M4.2	registerTickObserver()	TimeManager
M4.3	startTick()	TimeManager
M4.4	tick()	TimeManager
M4.5	updateTime()	DisplayManager
M4.6	stopStopWatch()	StopWatchMode
M4.7	stopTick()	TimeManager
M4.8	restartStopWatch()	StopWatchMode
M4.9	resetStopWatch()	StopWatchMode
M4.10	unregisterTick()	TimeManager
M5.1	createNewAnniversary()	AnniversaryMode
M5.2	getSlot()	AnniversaryStorage
M5.3	inputDateTime()	AnniversarySlot
M5.4	updateDateTime()	AnniversarySlot
M5.5	save()	AnniversarySlot
M5.6	setAlarm()	AlarmManager
M5.7	updateDate()	DisplayManager
M5.8	updateTitle()	DisplayManager
M5.9	selectAnniversary()	AnniversaryMode
M5.10	deleteAnniversary()	AnniversaryMode
M5.11	deleteSlot()	AnniversaryStorage
M5.12	deleteAlarm()	AlarmManager
M5.13	dismiss()	AnniversaryAlarm
M5.14	stop()	LightBuzzerManager
M5.15	turnOff()	LightBuzzerManager
M6.1	displayLotteryNumber	DisplayManager
M6.2	select4Mode	
M6.3	displayModeList	
M6.4	updateModeList	
M6.5	saveLotteryNumber	LotteryStorage
M6.6	sortLotteryNumber	Lottery
M6.7	setReminder	LotteryAlarm
M6.8	save4Mode	SettingsStorage
M6.9	resetData	
M6.10	sortLotteryNumber	Lottery
M6.11	generateLotteryNumber	RandomGenerator
M6.13	changeCurrentMode	ModeManager

중복 methods

M2 M8	displaySetupMode
M28, M	registerTickObserver()
M15, M	selectTimerViewMode
M6.6 M	sortLotteryNumber
M30, M	tick()