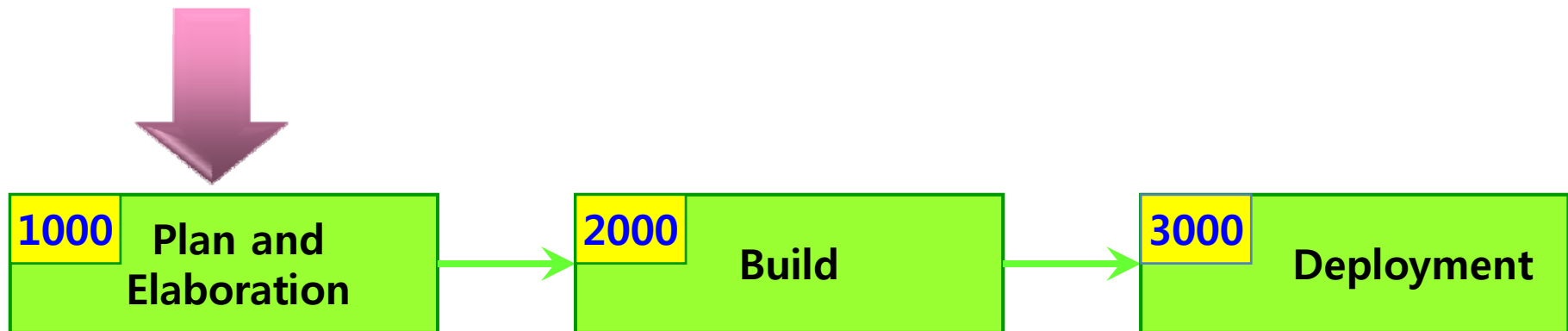


Case Study : Plan

- Library Management System

Lecturer: JUNBEOM YOO  
jbyoo@konkuk.ac.kr

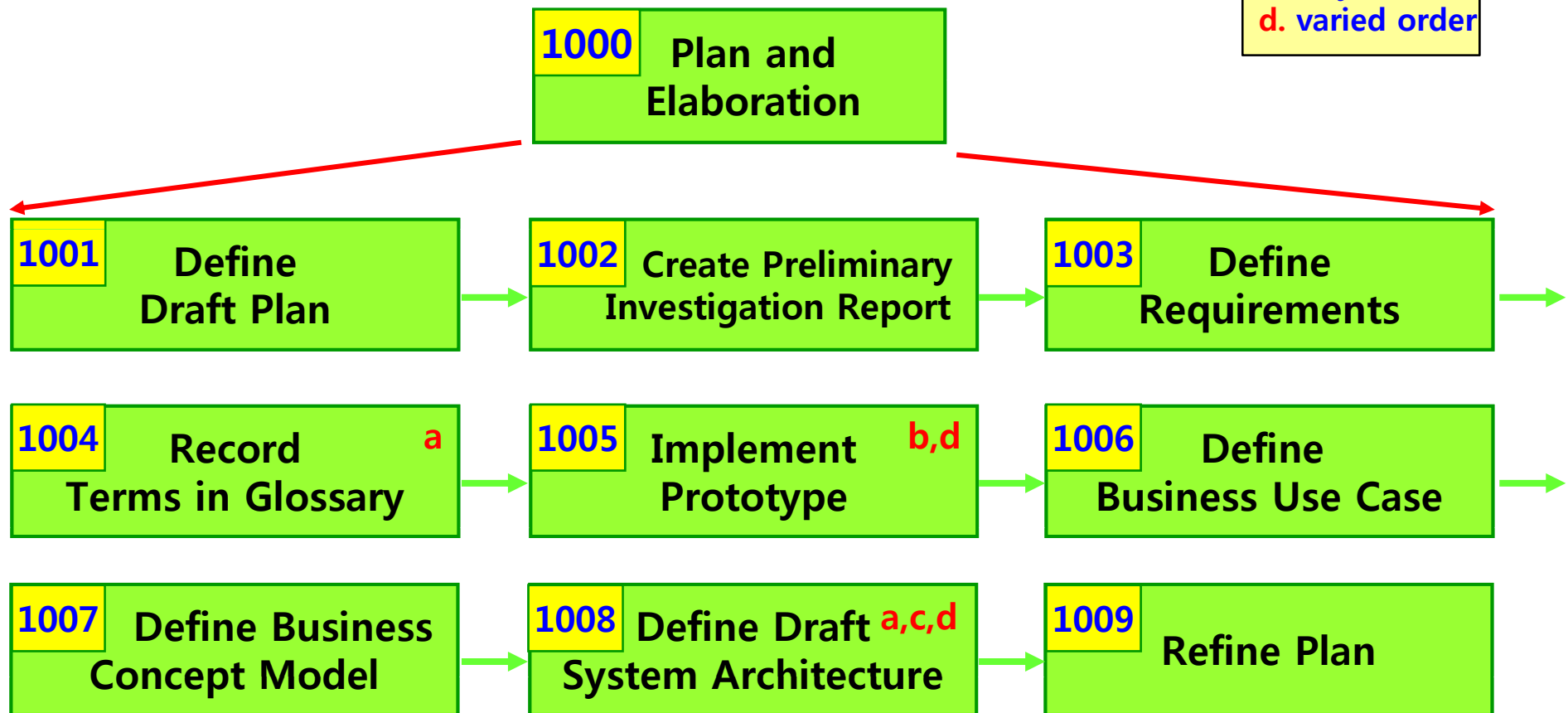
# Stage 1000. Plan and Elaboration



# Stage 1000. Plan and Elaboration

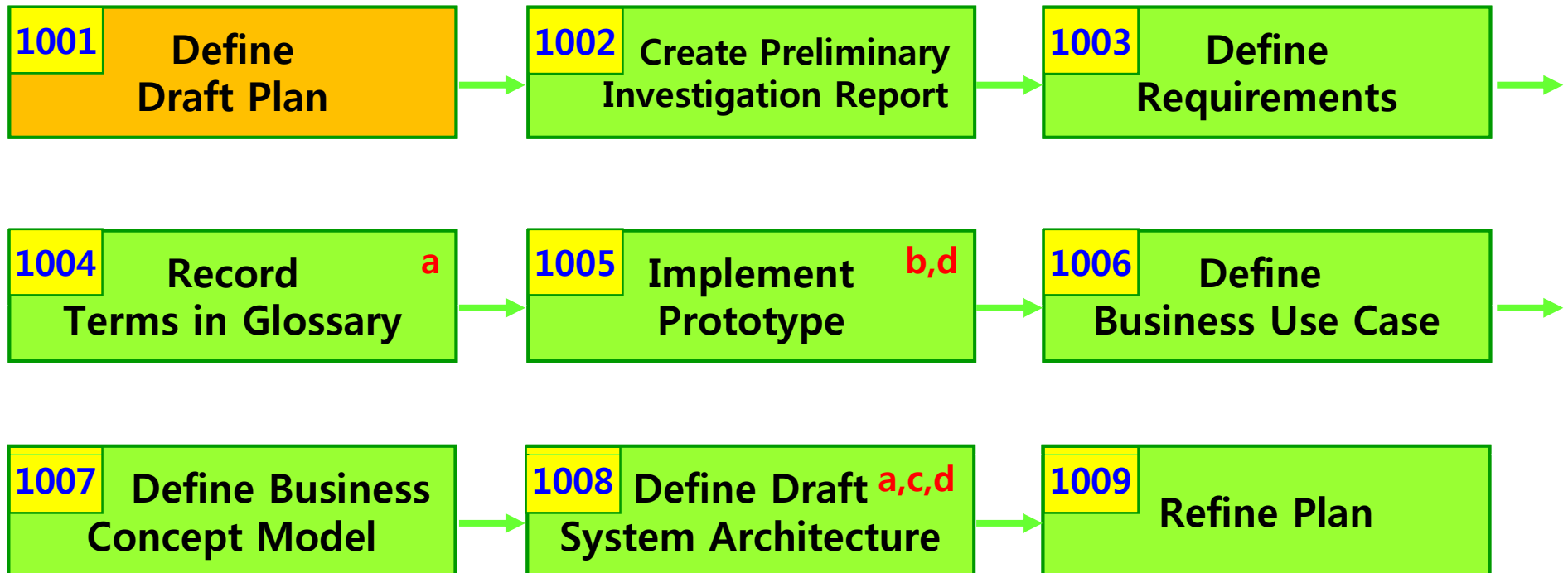
- Stage 1000 Activities

a. ongoing  
b. optional  
c. may defer  
d. varied order



# Activity 1001. Define Draft Plan

a. ongoing  
b. optional  
c. may defer  
d. varied order



# Activity 1001. Define Draft Plan

- Motivation
  - The size of title volumes and the number of users for a city library are sharply increasing.
  - Hence, the city wants to develop a 'Library Management System' in order to automate most of the library operations.
  - Among the various library operations, they want to automate the most commonly used operations such as loan, reservation, purchase, discarding old books, and simple statistics.
- Project Objectives
  - To develop a computerized library management software, that provides typical library operations such as:
    - Lend and return books, Reserve books, Maintaining Borrow information, and Purchasing new books.
  - The new software should be easy to learn and use, and efficient.

# Activity 1001. Define Draft Plan

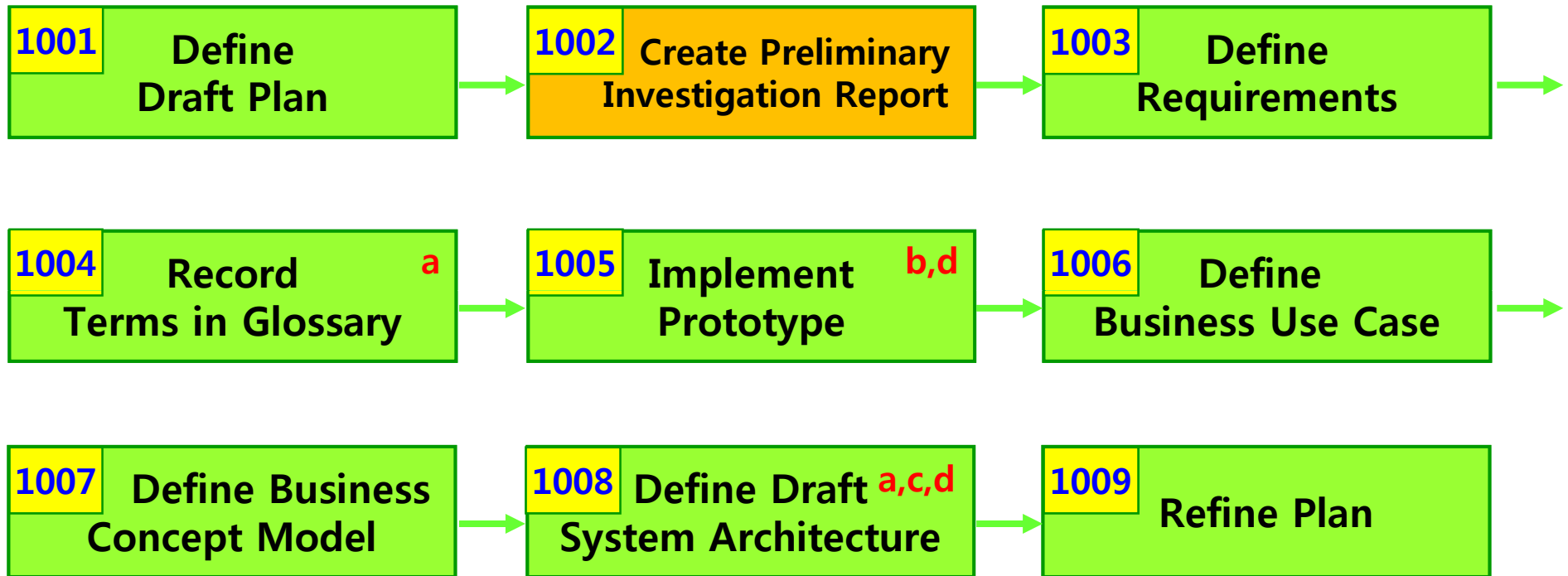
- Functional Requirements
  - Lend titles.
  - Return titles.
  - Reserve titles.
  - Purchase new titles.
  - Discard old titles.
  - Maintain borrower information.
- Non-Functional Requirements
  - The average response time for front desk operations should be less than 5 seconds.
  - The system should be designed to expandable and maintainable.

# Activity 1001. Define Draft Plan

- Resource Estimation
  - Human Efforts(Man-Month): 6-10 M/M ?
  - Human Resource:
  - Project Duration:
  - Cost:
- Other Information
  - Future Version
    - Adopt 3-Tier Client/Server Architecture.
    - Add Web Interface.

# Activity 1002. Create Preliminary Investigation Report

a. ongoing  
b. optional  
c. may defer  
d. varied order





# Activity 1002.

## Create Preliminary Investigation Report

- Alternative Solutions
  - Purchasing such a library managing software, if available.
  - Outsourcing
  - Other Options
- Project Justification (Business Demands)
  - Cost
  - Duration
  - Risk
  - Effect

# Activity 1002.

## Create Preliminary Investigation Report

- Risk Management

Risk	Probability	Significance	Weight
Lack of OO experience	4	4	16
First adoption of OSP	4	5	20
Lack of C++	2	3	6
Lack of domain knowledge	1	5	5
Team communication	3	3	9
Problem of requirements change	1	4	4
Lack of tool skill	2	2	4

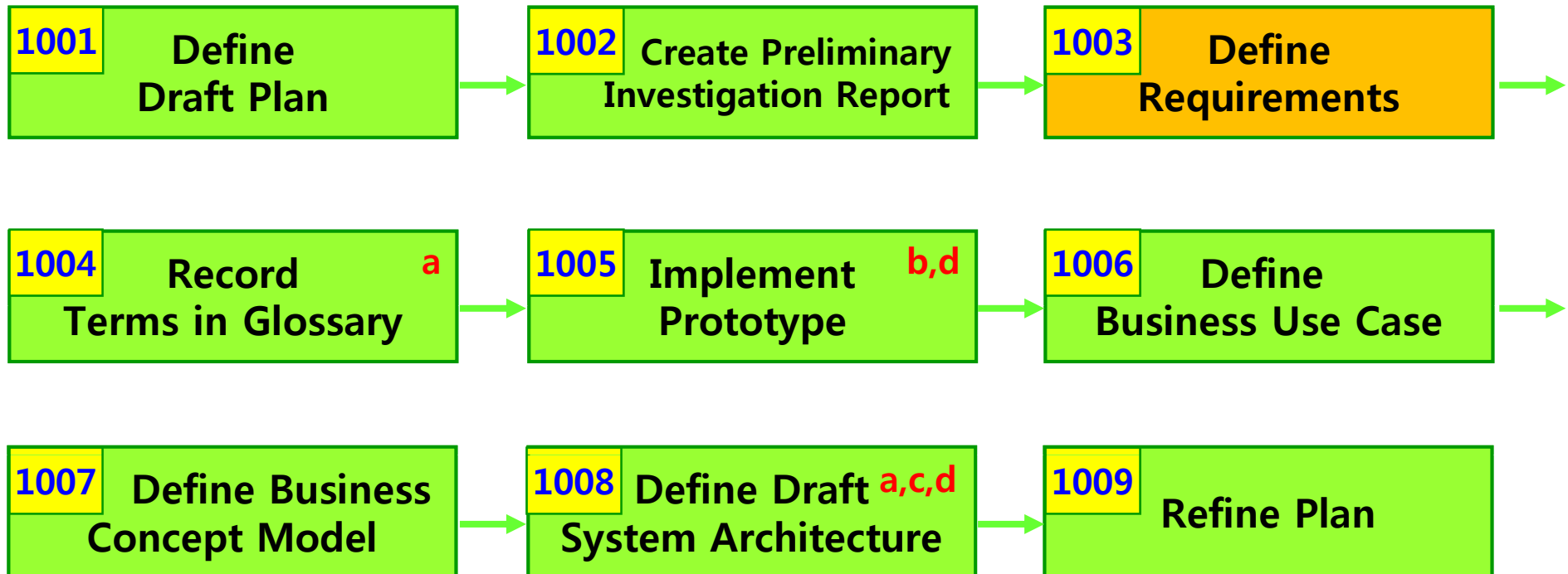
# Activity 1002.

## Create Preliminary Investigation Report

- Risk Reduction Plan
  - First adoption of OSP (20) : Try a pilot project using OSP
  - Lack of OO Project Experience (16) : Take part in a study group
  - Team Communication (9) : Have a team meeting on every Friday night
- Market Analysis
  - A few generic packages are available, however too expensive.
  - May be able to market the software to other similar-scaled libraries.
- Other Managerial Issues
  - The project should be completed by June, 2008.
    - Plan to participate in a SW exhibition.

# Activity 1003. Define Requirements

a. ongoing  
b. optional  
c. may defer  
d. varied order



# Activity 1003. Define Requirements

- Functional Requirements (Version 0.9)
  - A library lends books and magazines to borrowers, who are registered in the system.
  - A library handles the purchase of new titles. Popular titles are bought in multiple copies.
  - Old books and magazines are removed when they are out of date or in poor condition.
  - The librarian is an employee of the library, who interacts with the customers and whose work is supported by the system.
  - A borrower can reserve a book or magazine that is not currently available in the library, so that when it's returned or purchased by the library, that person is notified.
  - The reservation is canceled
    - when the borrower checks out the book or magazine, or
    - through an explicit canceling procedure.
  - The library can easily create, update, and delete information about the titles, borrowers, loans, and reservations in the system.

# Activity 1003. Define Requirements

- User Interviews

Index	Question	Answer
1	Direct Interface with Borrower?	No, indirect
2	Can borrower search books on-line?	No, next version
3	Charge a fee for late return?	Yes, it just calculates the fee, and no direct interface with accounting software.
4	Charge a fee for lost books?	Yes, it just calculates the fee.
5	How to handle unregistered borrower?	First register and then lend items.
6	Is a notification available?	Yes, it can be printed on cards.
7	Calculate total number of titles checked out?	Yes
8	Specify max number of loans per borrower?	Yes
9	Specify max number of days for loans?	Yes
10	Send a kindly-reminder(SMS/Email) for return due?	No
11	Classify adult boos?	Yes
12	Specify qualification for valid borrower?	No
13	Maintain reliable database?	Yes
14	Can control any system access?	Yes, through login and logout.

# Activity 1003. Define Requirements

- Functional Requirements (Version 1.0)
  - A library lends books and magazines to borrowers, who are registered in the system.
  - If the person has not been registered, the system first register the person. Then, lend titles.
  - A library handles the purchase of new titles. Popular titles are bought in multiple copies.
  - Old books and magazines are removed when they are out of date or in poor condition.
  - The librarian is an employee of the library who interacts with the customers(borrowers) and whose work is supported by the system.
  - A borrower can reserve a book or magazine that is not currently available in the library, so that when its returned or purchased by the library, that person is notified.
  - The system automatically prints 'post-cards' to notify the availability of the books. Then, the librarians mail them at the post office.

# Activity 1003. Define Requirements

- Functional Requirements (Version 1.0)
  - For unregistered person, the system first register the person. Then, make reservations
  - The reservation is canceled when the borrower checks out the book or magazine or through a explicit canceling procedure.
  - The library can easily create, update, and delete information about the titles, borrowers, loans, and reservations in the system.
  - Upon request, the system calculates the total # of items checked out.
  - For any over-due items, a late-return fee is calculated and charged.
  - For any items lost, a replacement-fee is computed and charged.
  - The system validates the system access through librarian IDs and passwords.
  - For each title, the librarians specify the maximum number of days that can be held by the borrowers.



# Activity 1003. Define Requirements

- Functional Requirements (Categorized Table)

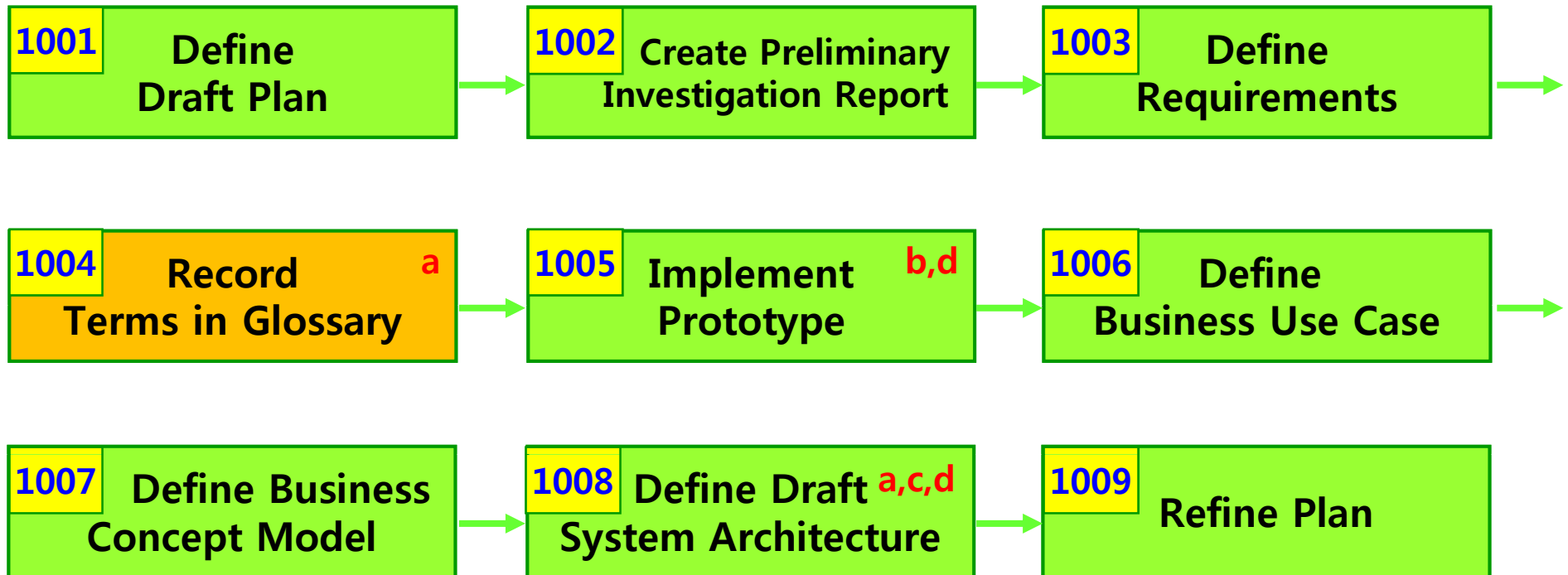
Ref. #	Function	Category
R1.1	Make reservation	Evident
R1.2	Remove reservation	Evident
R1.3	Lend Item	Evident
R1.4.1	Return title	Evident
R1.4.2	Calculate Late-Return-Fee	Hidden
R1.5	Calculate Replacement Fee	Evident
R1.6	Notify Availability	Hidden
R2.1	Add title	Evident
R2.2	Remove title	Evident
R2.3	Update title	Evident
R2.4	Add items	Evident
R2.5	Remove item	Evident
R2.6	Update item	Evident
R3.1	Add borrower	Evident
R3.2	Remove borrower	Evident
R3.3	Update borrower	Evident
R4.1	Validates system access	Evident
R5.1	Compute total # of items checked out	Evident

# Activity 1003. Define Requirements

- Performance Requirements
  - The average response time for front desk operations should be less than 5 seconds.
  - The post-card to notify availability must be printed out immediately after the reserved book becomes available.
- Operating Environment
  - Microsoft Windows 95 and NT
- Interface Requirements
  - The current version may incorporate a menu-driven approach.
  - Next version incorporates windows metaphor.
- Other Requirements
  - The system must control the system access.

# Activity 1004. Record Terms in Glossary

a. ongoing  
b. optional  
c. may defer  
d. varied order

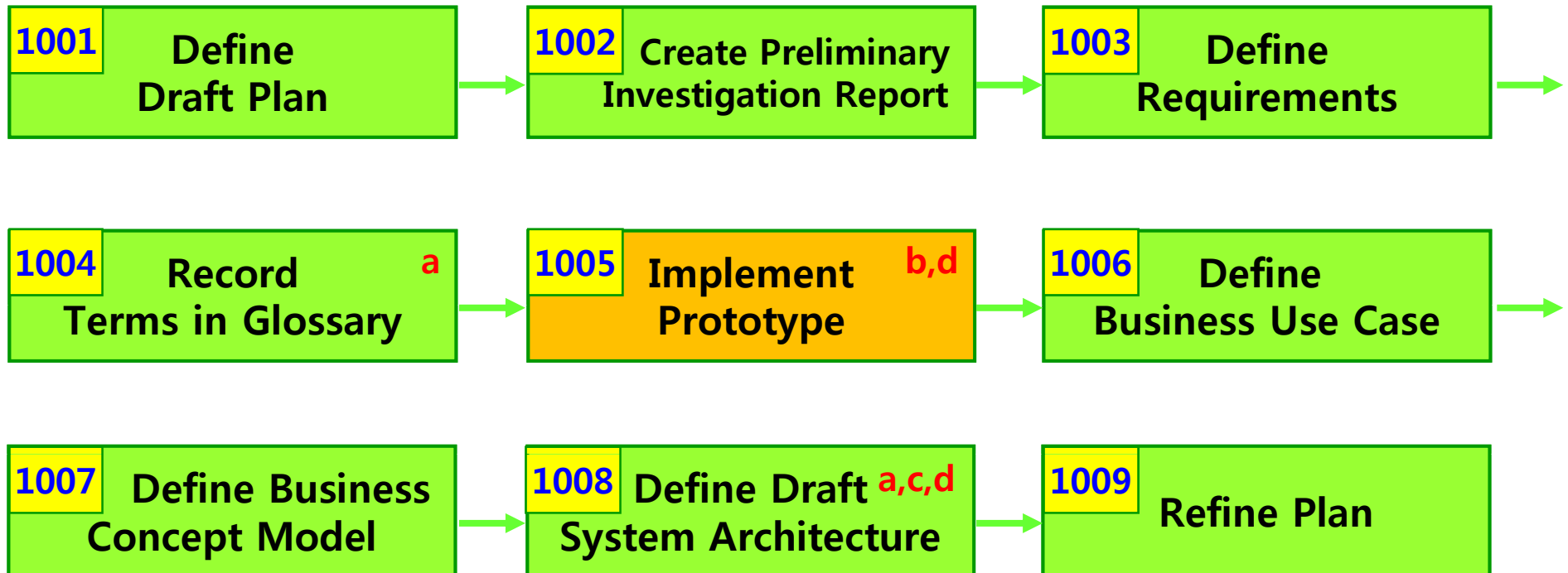


# Activity 1004. Record Terms in Glossary

Term	Description	Remarks
Title	Books or Magazines, which are registered in the library system	
Item	Each copy of books or magazines	
Loan	An action of checking out an item from the library	
Librarian	An employee of the library who handles the requests of borrowers.	

# Activity 1005. Implement Prototype

a. ongoing  
b. optional  
c. may defer  
d. varied order



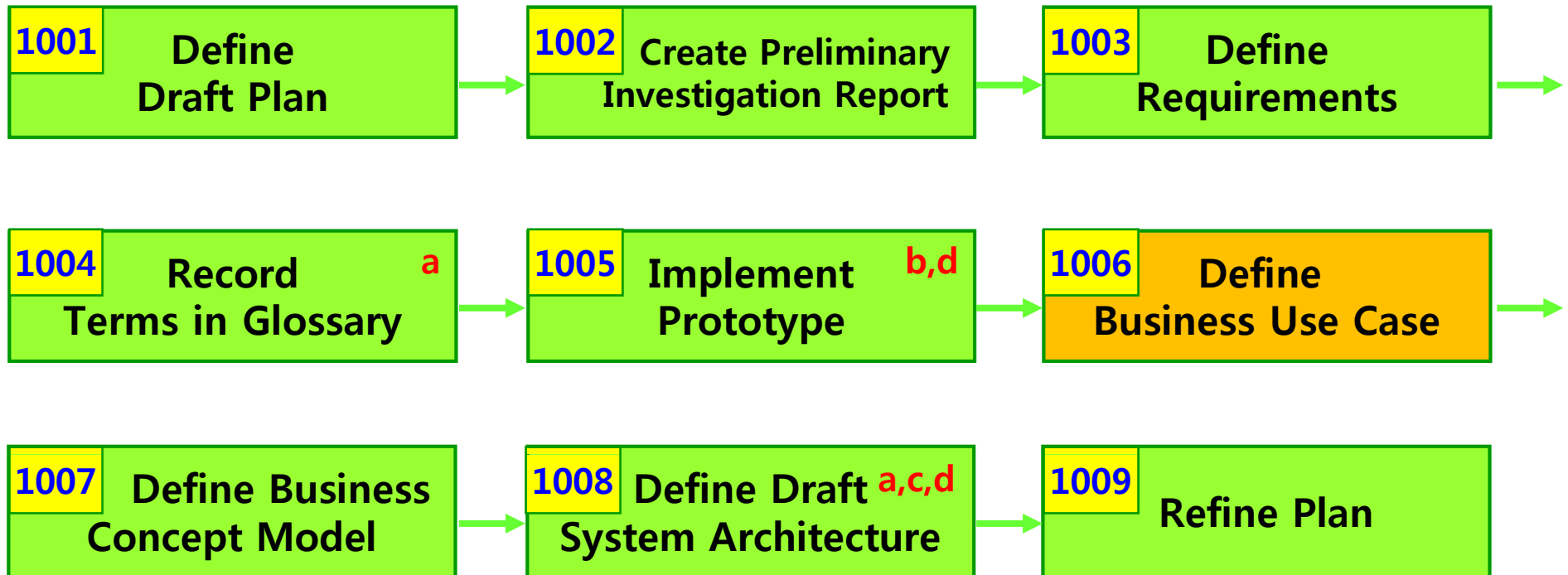
# Activity 1005. Implement Prototype

- User-Interface is sufficient for this project

Authority	Loan	Maintenance	Statistics
Exit	Lend Item Return Item	Add Title Update Title Remove Title	Total # Loans
	Make Reservation Remove Reservation	Add Item Update Item Remove Item	
	Get Replacement Fee	Add Borrower Update Borrower Remove Borrower	

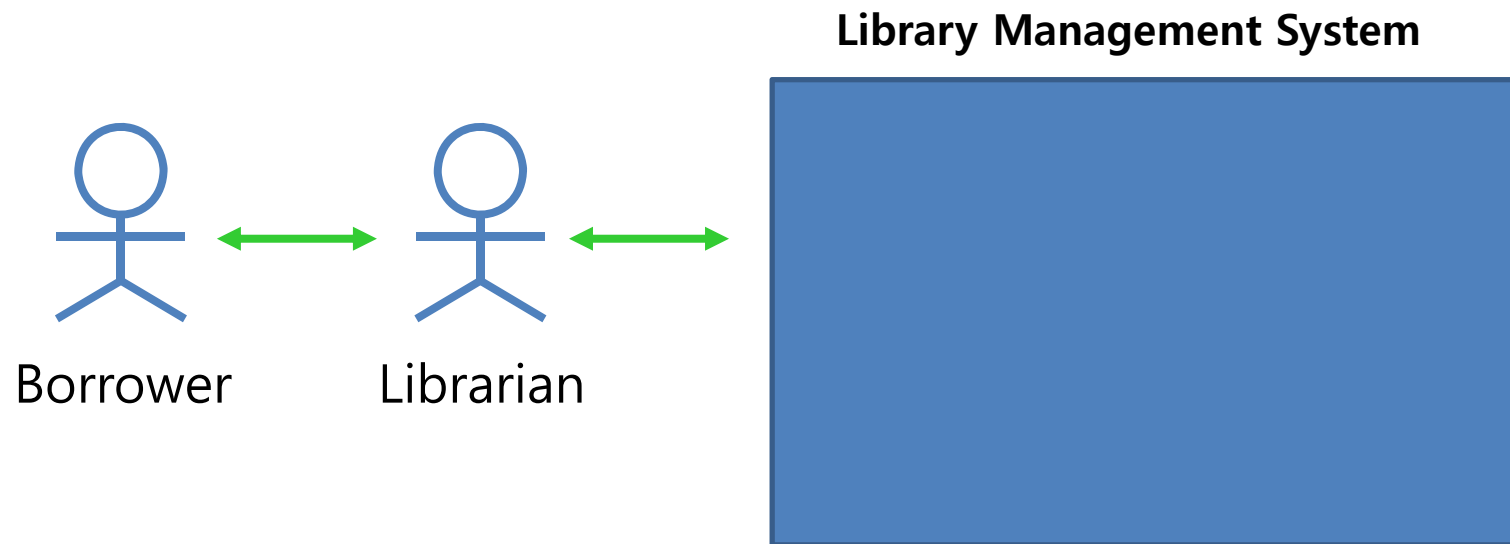
# Activity 1006. Define Business Use Case

a. ongoing  
b. optional  
c. may defer  
d. varied order



# Activity 1006. Define Business Use Case

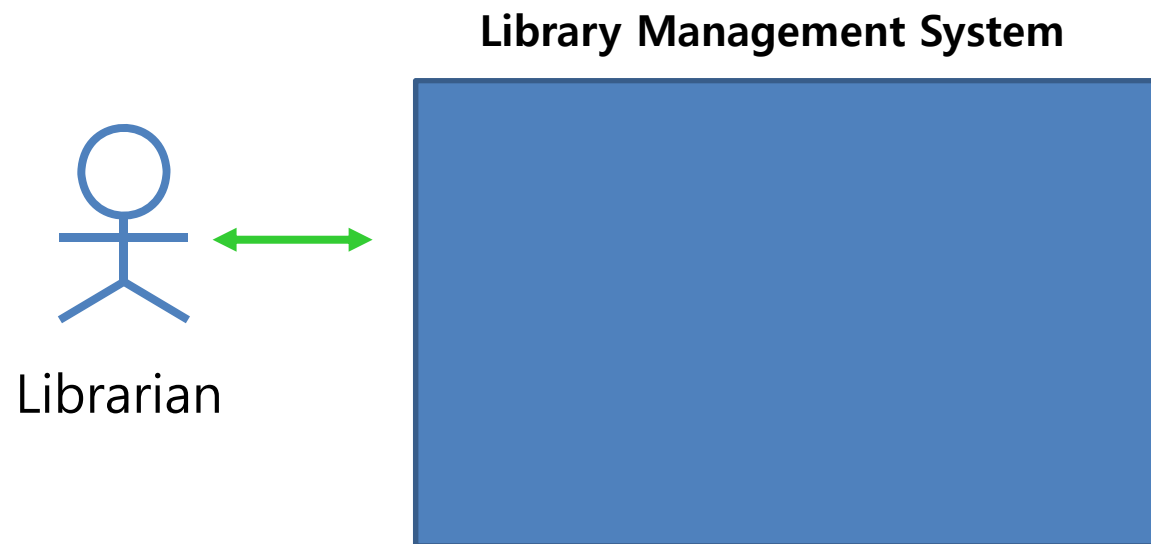
- Step 1. Define system boundary
  - All the functions defined earlier are inside the system boundary.





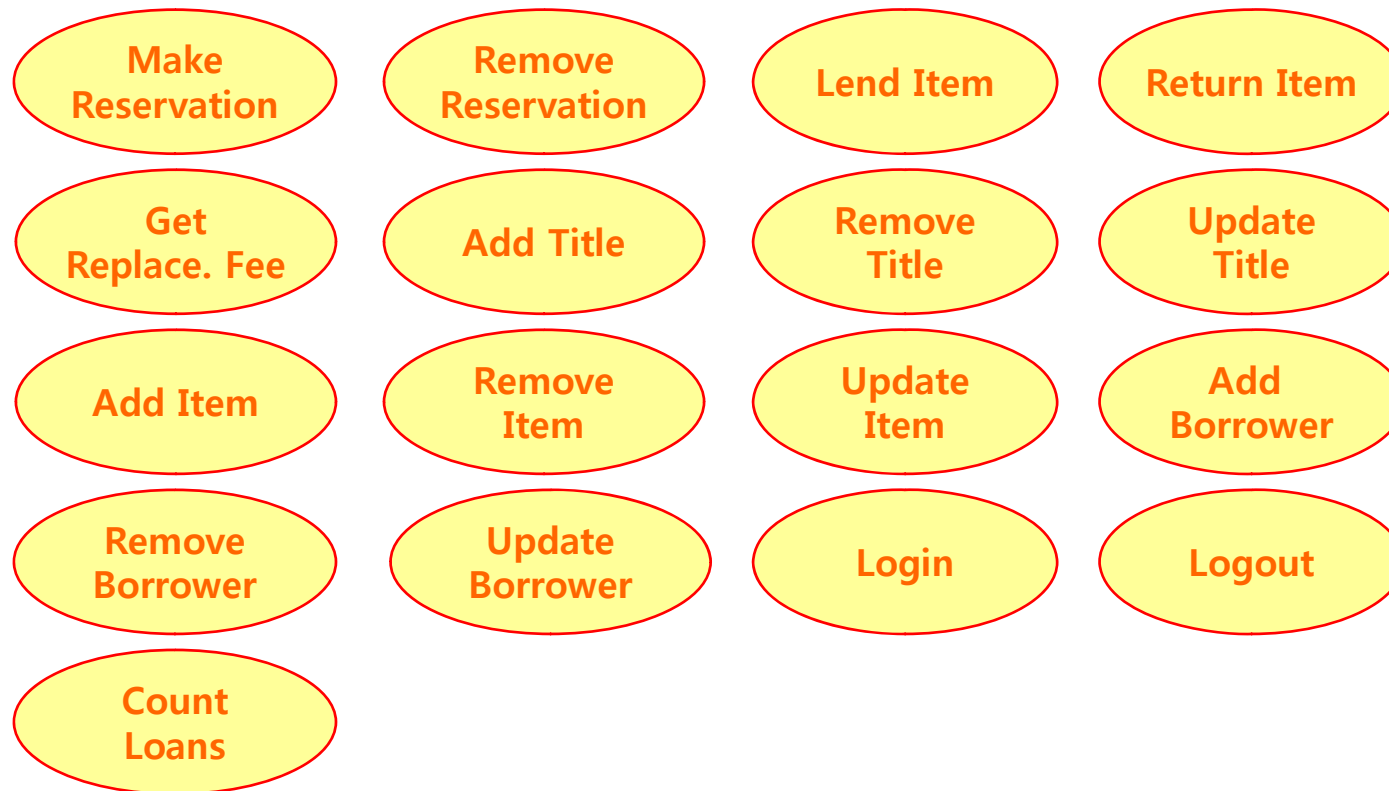
# Activity 1006. Define Business Use Case

- Step 2. Identify and describe actors.
  - Librarian : an employee of the library who interacts with the customers(borrowers) and whose work is supported by the system.



# Activity 1006. Define Business Use Case

- Step 3. Identify use cases
  - Use-cases by actor-based



# Activity 1006. Define Business Use Case

- Step 3. Identify use cases
  - Use-cases by event-based

**Calculate  
Late-Return-Fee**

**Modify  
Availability**

# Activity 1006. Define Business Use Case

- Step 4. Allocate system functions into related use cases

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

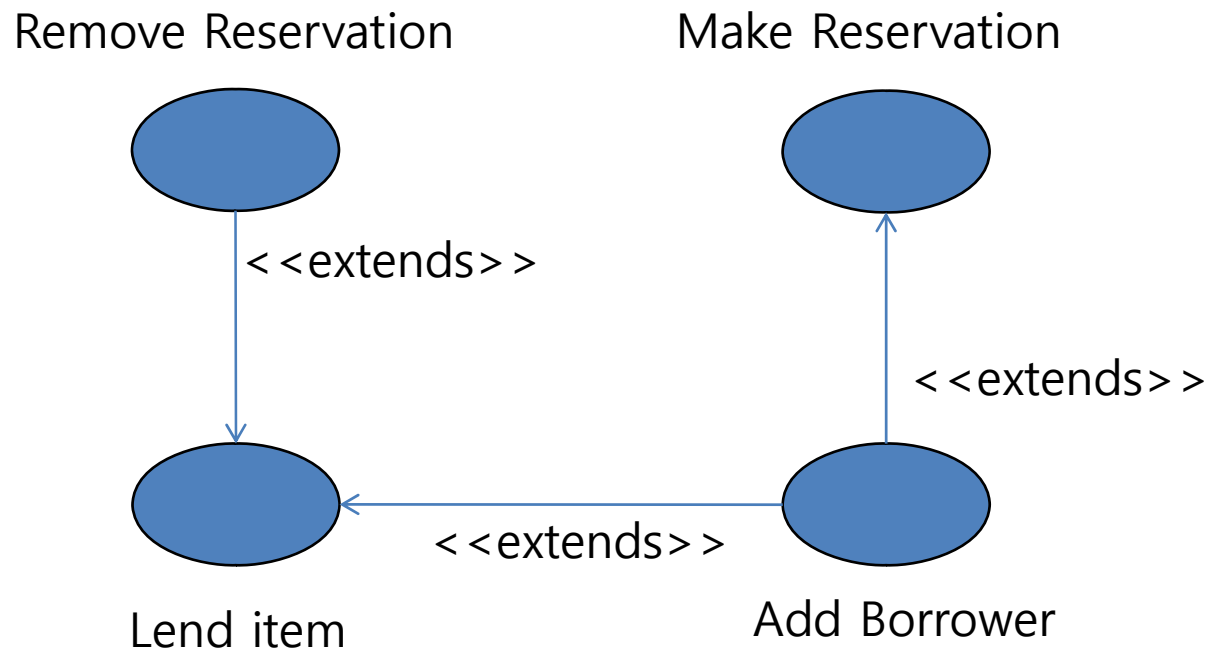
# Activity 1006. Define Business Use Case

- Step 5. Categorize use cases

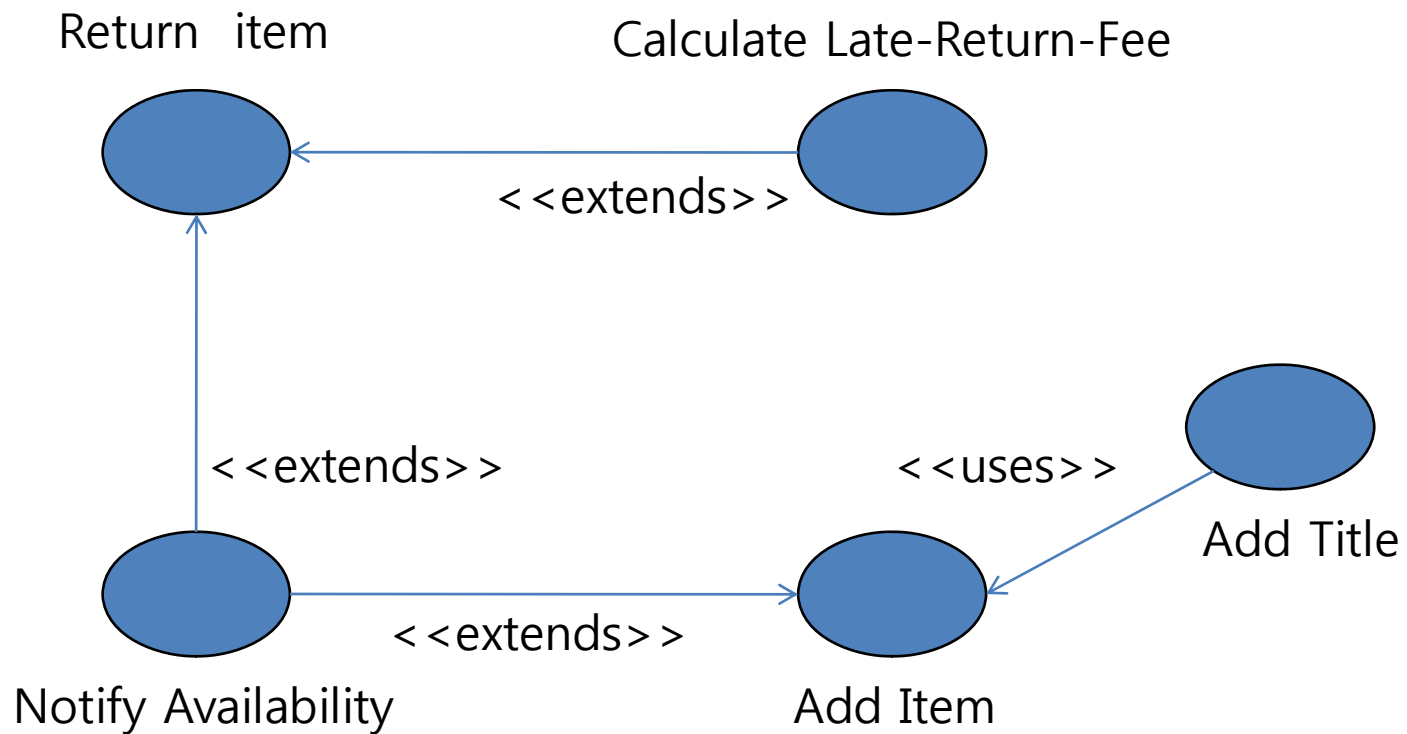
Ref. #	Function	Use Case Number & Name	Category
R1.1	Make reservation	1. Make Reservation	Primary
R1.2	Remove reservation	2. Remove Reservation	Primary
R1.3	Lend Item	3. Lend Item	Primary
R1.4.1	Return title	4. Return Title	Primary
R1.4.2	Calculate Late-Return-Fee	5. Calculate Late-Return-Fee	Primary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

# Activity 1006. Define Business Use Case

- Step 6. Identify relationships between use cases

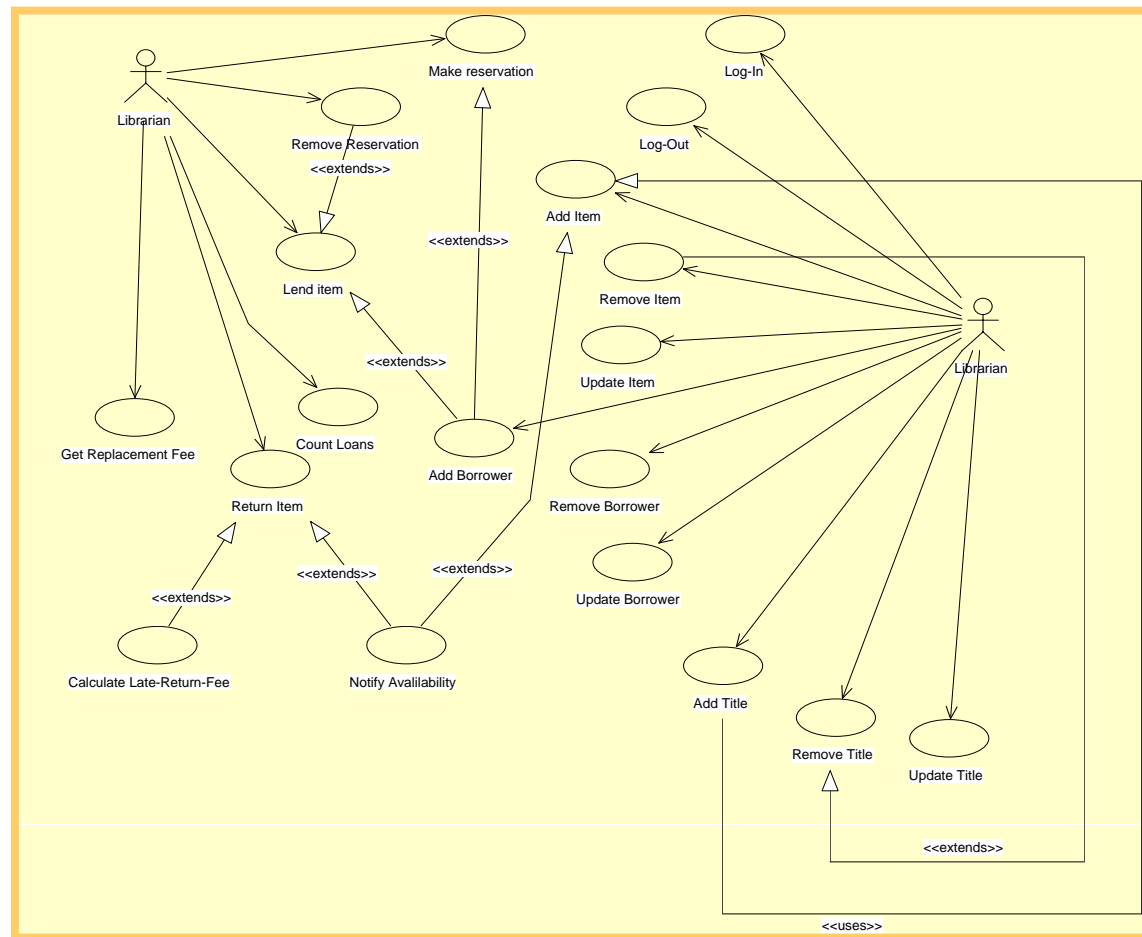


# Activity 1006. Define Business Use Case

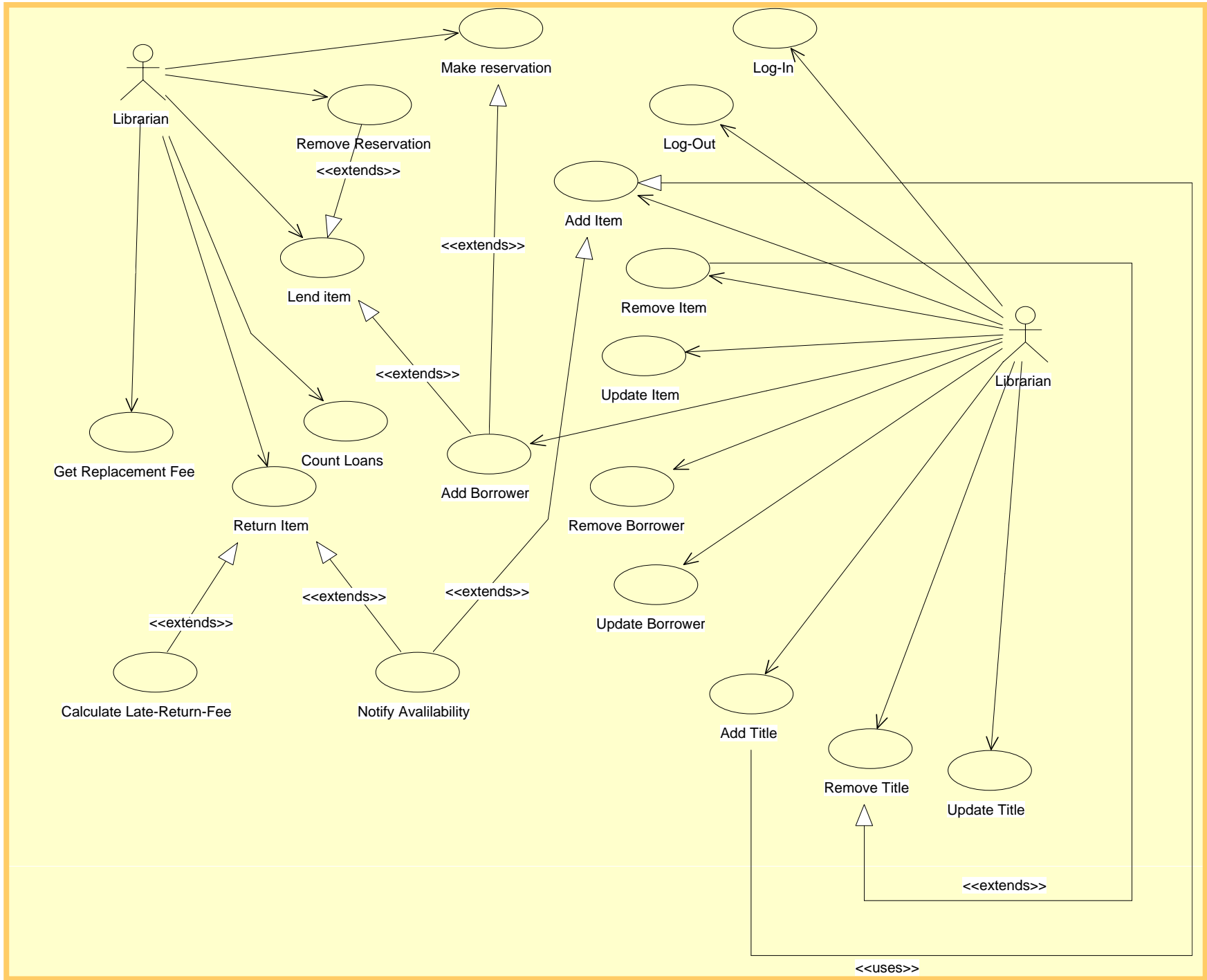


# Activity 1006. Define Business Use Case

- Step 7. Draw a use case diagram







# Activity 1006. Define Business Use Case

- Step 8. Describe use cases
  - High-level format

<b>Use Case</b>	Use case name
<b>Actors</b>	Actor name
<b>Description</b>	Overview of use case

# Activity 1006. Define Business Use Case

<b>Use Case</b>	1. Make Reservation
<b>Actors</b>	Librarian
<b>Description</b>	<ul style="list-style-type: none"><li>- This use case begins when a borrower arrives at the counter and then requests reservation.</li><li>- For a registered borrower, it makes a reservation slip (software-wise).</li><li>- For an unregistered borrower, the librarian registers the person and makes a reservation for the person.</li></ul>

<b>Use Case</b>	2. Remove Reservation
<b>Actors</b>	Librarian
<b>Description</b>	<ul style="list-style-type: none"><li>- A borrower who made a reservation can cancel his/her reservation.<ul style="list-style-type: none"><li>• Explicitly cancels the reservation. (Evident)</li></ul></li><li>- When a borrower checks out an item which he/she previously reserved, this use case is invoked automatically.<ul style="list-style-type: none"><li>• Hidden system function</li></ul></li></ul>

# Activity 1006. Define Business Use Case

<b>Use Case</b>	3. Lend Item
<b>Actors</b>	Librarian
<b>Description</b>	<ul style="list-style-type: none"><li>- This use case begins when the borrower arrives at the front desk with items to lend.</li><li>- If a borrower does not registered, register first his/her information in the system.</li><li>- This use case records the date, borrower ID, item ID and other relevant information for this loan.</li></ul>

<b>Use Case</b>	4. Return Item
<b>Actors</b>	Librarian
<b>Description</b>	<ul style="list-style-type: none"><li>- This use case begins when a borrower returns items at the counter.</li><li>- If the item is returned past due date, a late-return-fee is computed, so that the borrower should pay the penalty.</li></ul>

# Activity 1006. Define Business Use Case

<b>Use Case</b>	5. Calculate Late-Return-Fee
<b>Actors</b>	None
<b>Description</b>	<ul style="list-style-type: none"><li>- This use case computes the penalty amount for items returned late.</li><li>- It first computes the number of extra days held by the borrower, then multiplies it by a pre-determined daily rate for late returns.</li></ul>
<b>Use Case</b>	6. Get Replacement Fee
<b>Actors</b>	Librarian
<b>Description</b>	<ul style="list-style-type: none"><li>- This use case computes the cost for replacing the lost book.</li><li>- It first finds out the current price of the lost book, and add the handling cost to the book price.</li></ul>
<b>Use Case</b>	7. Notify Availability
<b>Actors</b>	None
<b>Description</b>	<ul style="list-style-type: none"><li>- This use case prints the book title that just became available, number of days held by the library, the name and address of the person who reserved on a post-card.</li><li>- The actual mailing will be done manually by the librarian.</li></ul>

# Activity 1006. Define Business Use Case

<b>Use Case</b>	8. Add Title
<b>Actors</b>	Librarian
<b>Description</b>	<ul style="list-style-type: none"><li>- Whenever a new kind of book is purchased, the book information is recorded into the system.</li><li>- Then, it invokes 'Add Item' use case to record the number of copies purchased.</li></ul>
<b>Use Case</b>	9. Remove Title
<b>Actors</b>	Librarian
<b>Description</b>	<ul style="list-style-type: none"><li>- Some old books are selected for removal by the librarians.</li><li>- This use case deletes the information of the book to be removed.</li><li>- And, it will be no longer available for loans.</li></ul>
<b>Use Case</b>	10. Update Title
<b>Actors</b>	Librarian
<b>Description</b>	<ul style="list-style-type: none"><li>- This use case will change the recorded information of the title.</li><li>- What actual kinds of information?</li></ul>

# Activity 1006. Define Business Use Case

<b>Use Case</b>	11. Add Item
<b>Actors</b>	Librarian
<b>Description</b>	<ul style="list-style-type: none"> <li>- When additional copies (of the currently available title) are purchases, this updates the total number of copies for each title.               <ul style="list-style-type: none"> <li>• Date, Price, Bookstore, Available, etc.</li> </ul> </li> <li>-When a reservation has been made for this title, this use case invokes 'notify availability' use case.</li> </ul>
<b>Use Case</b>	12. Remove Item
<b>Actors</b>	Librarian
<b>Description</b>	<ul style="list-style-type: none"> <li>- This use case will update the number of items for each title.</li> <li>- If no more item is remaining after removal, this use case will invoke 'Remove Title' use case.</li> </ul>
<b>Use Case</b>	13. Update Item
<b>Actors</b>	Librarian
<b>Description</b>	<ul style="list-style-type: none"> <li>- This use case updates the information of the items.</li> <li>- What actual kinds of information will be updated ?</li> </ul>

# Activity 1006. Define Business Use Case

<b>Use Case</b>	14. Add Borrower
<b>Actors</b>	Librarian
<b>Description</b>	- This use case will record the information of the new borrower such as name, address, phone, loan priority, etc.

<b>Use Case</b>	15. Remove Borrower
<b>Actors</b>	Librarian
<b>Description</b>	- This use case deletes the information of borrower from the system, so that the person can no longer check out titles. - This may happen if the borrower has a bad return history or has not been using the library longer than 2 years.

<b>Use Case</b>	16. Update Borrower
<b>Actors</b>	Librarian
<b>Description</b>	- This use case updates the information of the borrower such as new address and phone.



# Activity 1006. Define Business Use Case

<b>Use Case</b>	17. Log-In
<b>Actors</b>	Librarian
<b>Description</b>	<ul style="list-style-type: none"><li>- This use case reads the user ID and password of the librarian, and verifies.</li><li>- If an invalid information is entered, it will re-prompt and read the ID and password.</li><li>- After 3 successive failures of login, it records this 'attach' information and automatically returns to the initial menu.</li></ul>

<b>Use Case</b>	18. Log-Out
<b>Actors</b>	Librarian
<b>Description</b>	<ul style="list-style-type: none"><li>- This use case records the date and time of the current logout, and returns to the initial menu.</li></ul>

<b>Use Case</b>	19. Count Loans
<b>Actors</b>	Librarian
<b>Description</b>	<ul style="list-style-type: none"><li>- This use cases computes the total number of items checked out.</li></ul>

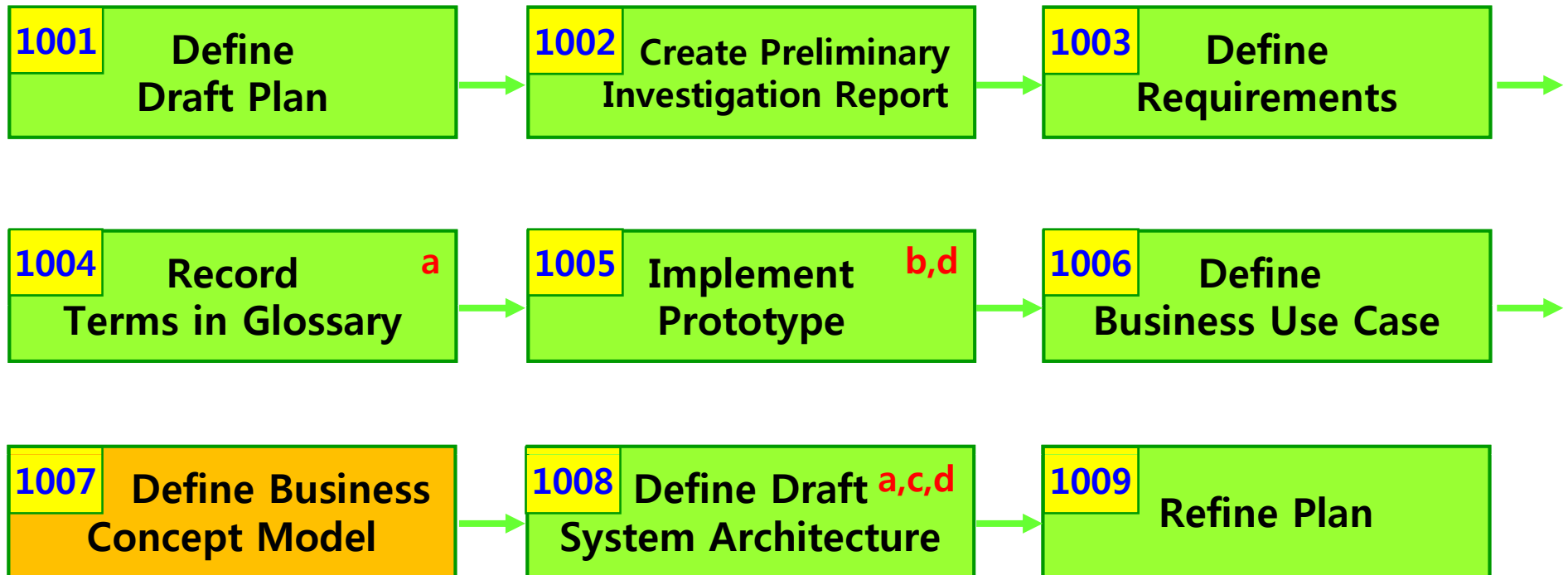
# Activity 1006. Define Business Use Case

- Step 9. Rank use cases

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

# Activity 1007. Define Business Concept Model

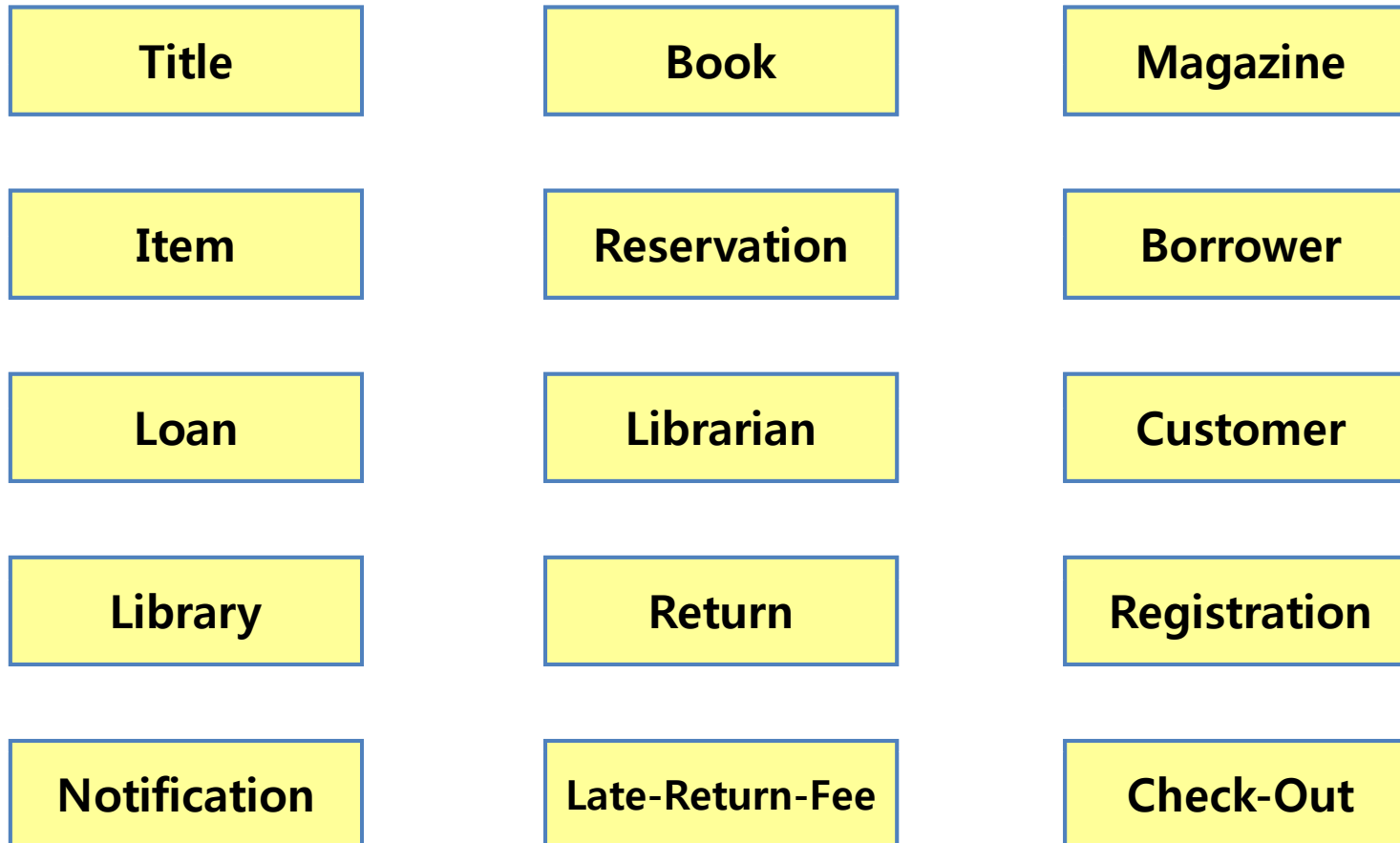
a. ongoing  
b. optional  
c. may defer  
d. varied order



# Activity 1007.

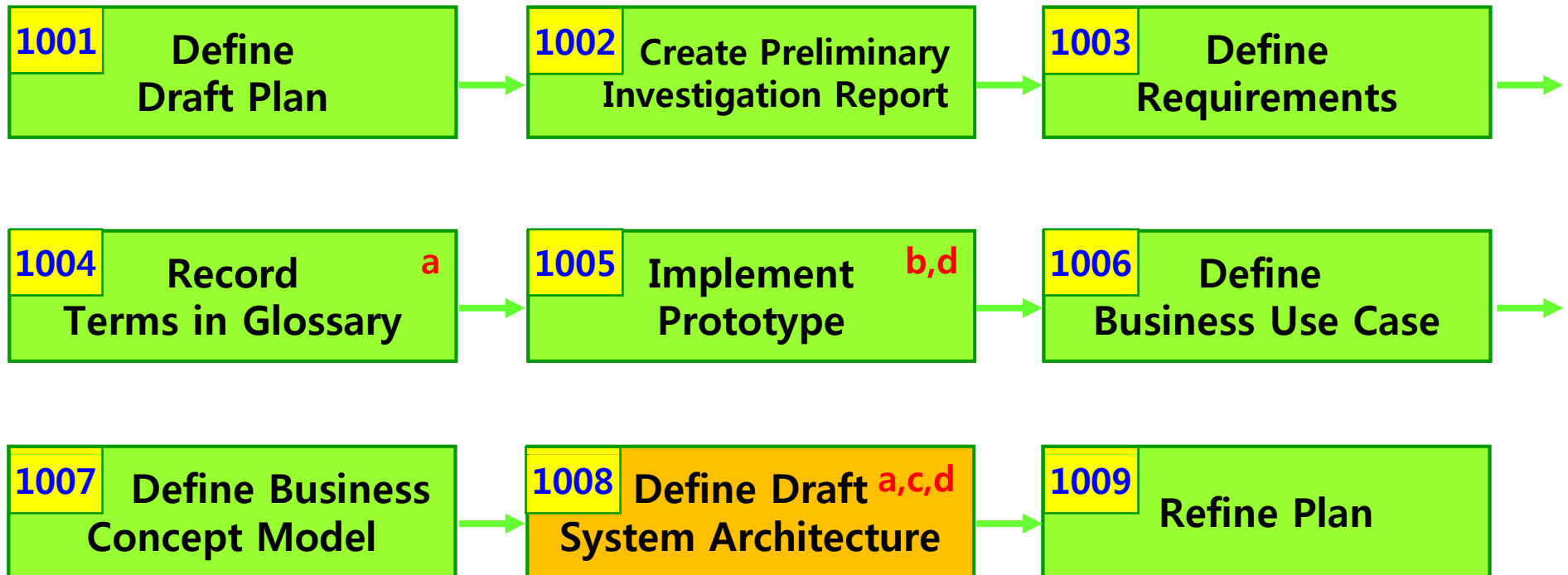
## Define Business Concept Model

- Identify 'Concepts' in the target domain.



# Activity 1008. Define Draft System Architecture

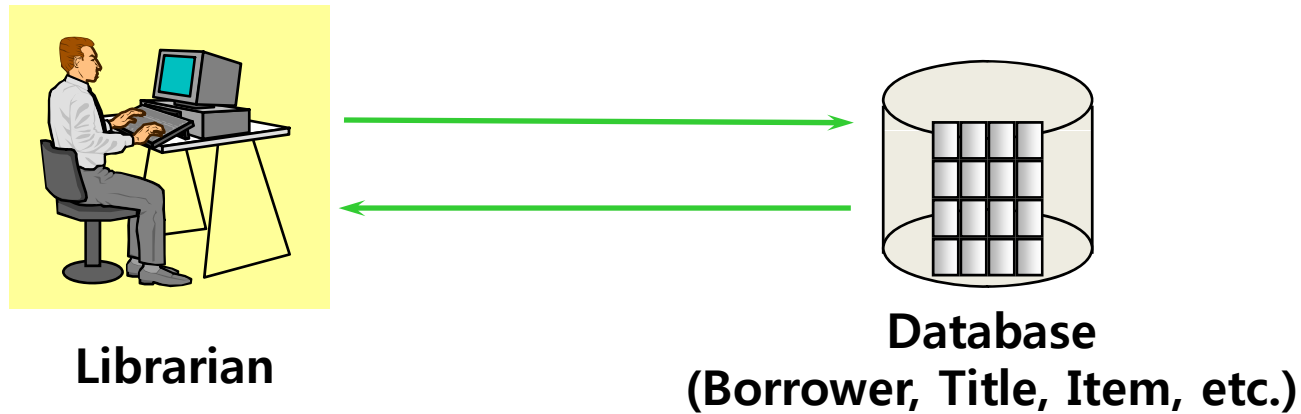
a. ongoing  
b. optional  
c. may defer  
d. varied order



# Activity 1008.

## Define Draft System Architecture

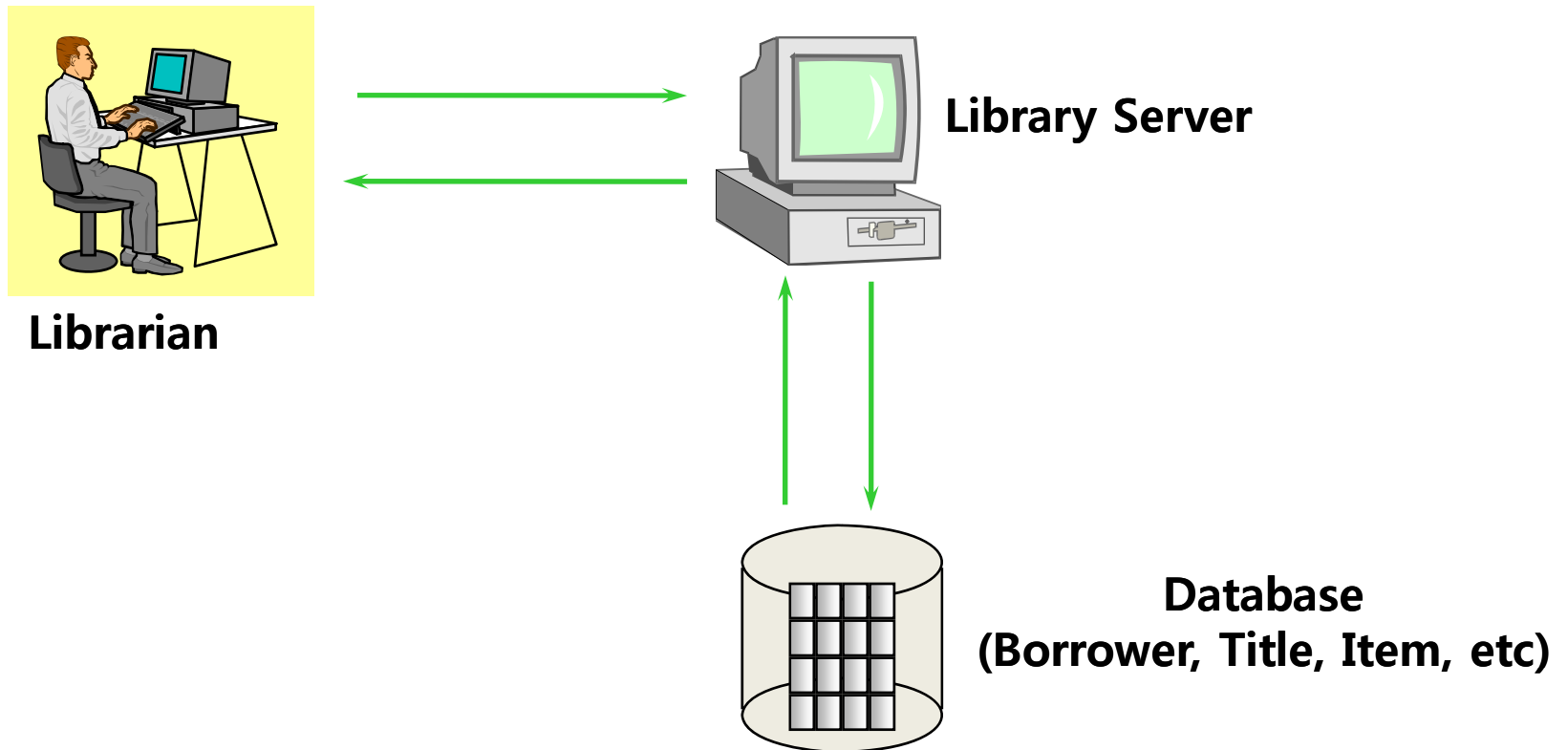
- Initial Version
  - Stand alone



# Activity 1008.

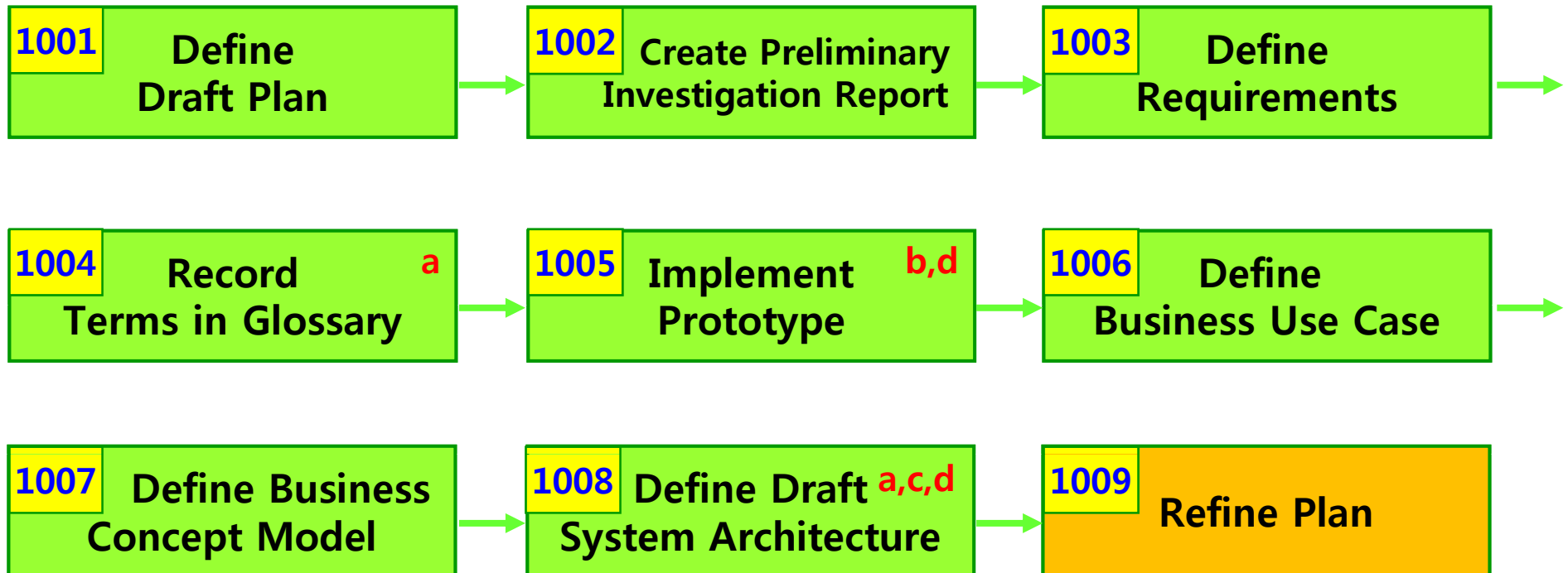
## Define Draft System Architecture

- Next Version
  - 3-Tier Client/Server



# Activity 1009. Refine Plan

a. ongoing  
b. optional  
c. may defer  
d. varied order





# Activity 1009. Refine Plan

- Project Scope
  - The library management software automates typical library operations; reservation, lending item, adding, removing, and updating the information of title, item, and borrower.
- Project Objectives
  - To develop a computerized library management software, that provides typical library operations such as:
    - Lend and return books, Reserve books, Maintaining Borrow information, and Purchasing new books.
  - The new software should be easy to learn and use, and efficient.

# Activity 1009. Refine Plan

- Functional Requirements

Ref. #	Function	Category
R1.1	Make reservation	Evident
R1.2	Remove reservation	Evident
R1.3	Lend Item	Evident
R1.4.1	Return title	Evident
R1.4.2	Calculate Late-Return-Fee	Hidden
R1.5	Calculate Replacement Fee	Evident
R1.6	Notify Availability	Hidden
R2.1	Add title	Evident
R2.2	Remove title	Evident
R2.3	Update title	Evident
R2.4	Add items	Evident
R2.5	Remove item	Evident
R2.6	Update item	Evident
R3.1	Add borrower	Evident
R3.2	Remove borrower	Evident
R3.3	Update borrower	Evident
R4.1	Validates system access	Evident
R5.1	Compute total # of items checked out	Evident

# Activity 1009. Refine Plan

- Performance Requirements
  - When making reservations, the information of reservation will appear within 5 seconds.
  - When lending items, the content of lending item will appear within 5 seconds.
  - When returning items, the content of returning item will appear within 5 seconds.
- Operating Environment
  - Microsoft Windows 95 and NT
- User Interface Requirements
  - Menu-driven approach
  - Should be designed for upgrading to 'Window-based' version.

# Activity 1009. Refine Plan

- Other Requirements
  - The content of database should be maintained reliably.
  - System should control the system access.
- Resources
  - Man Month : 6 Persons
    - A Team Leader
    - A Document Manager
    - 3-4 Engineers
  - Period : 5 Days (Around 40 Hours)
  - Hardware : Pentium PC
  - Software
    - OS : Windows 95/NT
    - Programming Language : C++
    - Case Tools : Rational Rose, Paradigm Plus

# Activity 1009. Refine Plan

- Scheduling

Stage	Phase(00x0)/Activity(000x)	Schedule(Day)				
		1	2	3	4	5
1000. Plan & Elaborate	1001. Define Draft Plan	█				
	1002. Create Preliminary Investigation Report	█				
	1003. Define Requirements	█				
	1004. Record Terms in Glossary	█				
	1005. Implement Prototype	█				
	1006. Define Use Cases	█				
	1007. Define Draft Conceptual Model	█				
	1008. Define Draft System Architecture	█				
	1009. Refine Plan	█				
	2000. Build	2010. Revise Plan				
	2020. Synchronize Artifacts					
	2030. Analyze					
	2031. Define Essential Use Case	█				
	2032. Refine Use Case Diagrams	█				
	2033. Refine Conceptual Model	█				
	2034. Refine Glossary	█				
	2035. Define System Sequence Diagrams	█				
	2036. Define Operation Contracts	█				
	2037. Define State Diagrams	█				
	2040. Design					
	2041. Define Real Use Cases	█				
	2042. Define Reports, UI and Storyboards	█				
	2043. Refine System Architecture	█				
	2044. Define Interaction Diagrams	█				
	2045. Define Design Class Diagrams	█				
	2046. Define Database Schema	█				
	2050. Construct					
	2051. Implement Class & Interface Definition					
	2052. Implement Methods.					
	2053. Implement Windows					
	2054. Implement Reports					
	2055. Implement DB Schema					
	2056. Write Test Code					
	2060. Test					
	2061. Unit Testing					
	2062. Integration Testing					
	2063. System Testing					
	2064. Performance Testing					
	2065. Acceptance Testing					
	2066. Documentation Testing					
3000. Deployment	3001. Complete Technical Documents					
	3002. Complete User Documents					
	3003. System Testing					
	3004. Acceptance Testing					
	3005. Documentation Testing					
	3006. Train					
	3007. Establish Parallel Runs and Crossover					
	3008. Establish Support					
	3009. Install					

Stage	Phase(00x0)/Activity(000x)	Schedule(Day)				
		1	2	3	4	5
1000. Plan & Elaborate	1001. Define Draft Plan 1002. Create Preliminary Investigation Report 1003. Define Requirements 1004. Record Terms in Glossary 1005. Implement Prototype 1006. Define Use Cases 1007. Define Draft Conceptual Model 1008. Define Draft System Architecture 1009. Refine Plan					
2000. Build	2010. Revise Plan 2020. Synchronize Artifacts 2030. Analyze 2031. Define Essential Use Case 2032. Refine Use Case Diagrams 2033. Refine Conceptual Model 2034. Refine Glossary 2035. Define System Sequence Diagrams 2036. Define Operation Contracts 2037. Define State Diagrams 2040. Design 2041. Define Real Use Cases 2042. Define Reports, UI and Storyboards 2043. Refine System Architecture 2044. Define Interaction Diagrams 2045. Define Design Class Diagrams 2046. Define Database Schema 2050. Construct 2051. Implement Class & Interface Definition 2052. Implement Methods. 2053. Implement Windows 2054. Implement Reports 2055. Implement DB Schema 2056. Write Test Code 2060. Test 2061. Unit Testing 2062. Integration Testing 2063. System Testing 2064. Performance Testing 2065. Acceptance Testing 2066. Documentation Testing					
3000. Deploy-ment	3001. Complete Technical Documents 3002. Complete User Documents 3003. System Testing 3004. Acceptance Testing 3005. Documentation Testing 3006. Train 3007. Establish Parallel Runs and Crossover 3008. Establish Support 3009. Install					

# Activity 1009. Refine Plan

- Configuration Management
  - Check-in and Check-Out Control
    - Change Request, Revision, Version Control
  - Directory Management
    - We will use 'UML-Project-Repository'
- Quality Assurance Plan
  - Formal Technical Review(FTR) : Inspection & Work-through
    - Should be applied to each activity.
  - Define Quality Assurance Metrics.

# Stage 1000. Plan and Elaboration

- Stage 1000 Activities

a. ongoing  
b. optional  
c. may defer  
d. varied order

