

## Project – Milestone #2

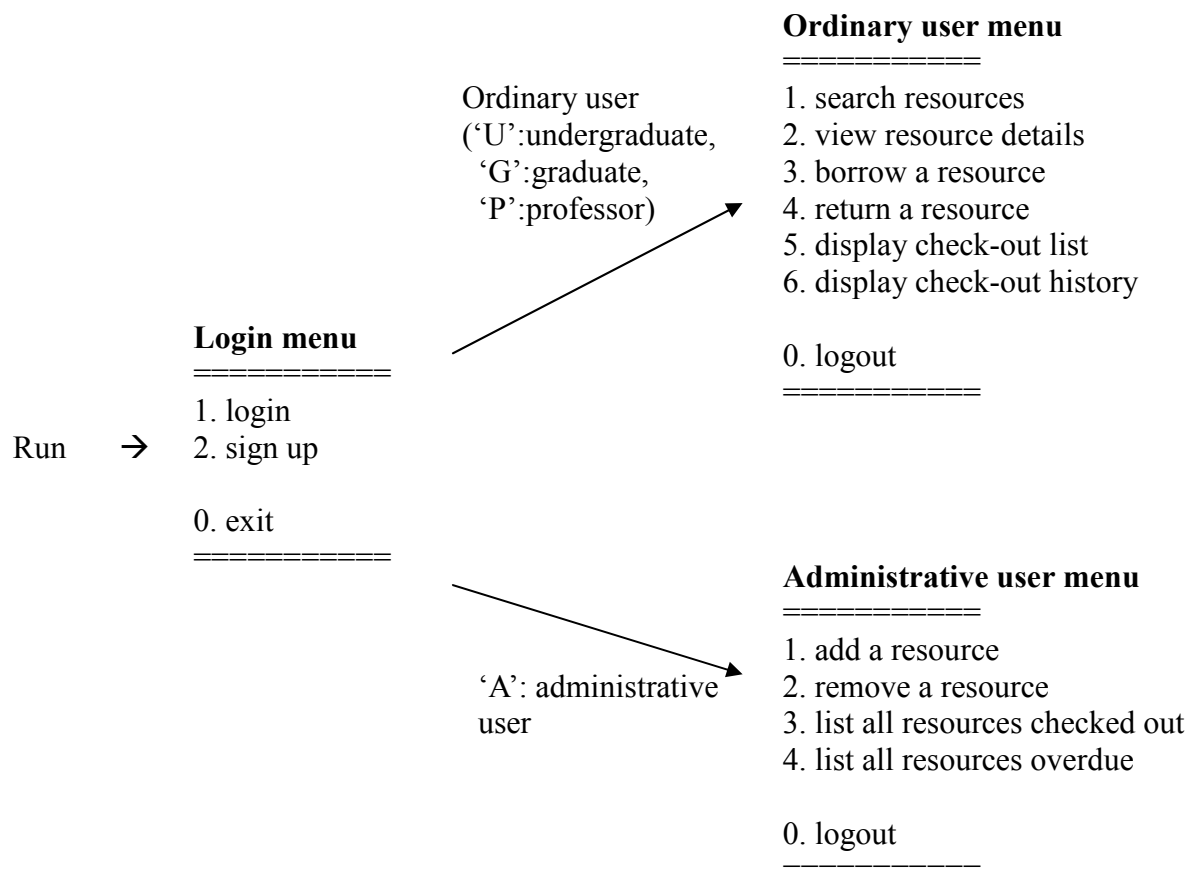
### Implementing a Database Application using JDBC for University Library Management System

#### Introduction

You will have a database schema for the university library management system at present. As the next step, you should implement a simple Java application program for library users with Tibero RDBMS. Through your program, ordinary users can search/borrow/return resources, and administrative users can add/remove resources. Your application should manage database tables by executing DML or query statements using the JDBC driver. You can modify your table schema if it is not enough to implement the following specifications.

#### Program Interfaces

Use the command line interfaces for the simplicity.



#### Program Details

##### Login menu

- 1. login
  - input: id(string) and password(string)
  - action: identify the user and show next menu according to the user's type
  - error: id does not exist or password is incorrect  
id or password is not specified

## 2. sign up: insert a new user

- input: id(string), password(string), name(string), and user type('U', 'G', 'P', 'A')
- action: add the user into the database
- error: the user type input is not in ('U', 'G', 'P', 'A')  
an input field is not specified

Ordinary user menu

## 1. search resources

- input: a set of keyword separated by whitespaces(string)
- action: search the resources which include all keywords in their title or author  
list their id, type('Book', 'Journal', 'Multimedia'), title, first author, publisher, year, availability('charged', 'not charged')
- error: no keyword is specified

## 2. view resource details

- input: resource id(string)
- action: list its type('Book', 'Journal', 'Multimedia'), title, all authors, publisher, year, pages, and additional information according to its type  
display copy#(number), availability('charged', 'not charged') and (most recent)  
return date of each copy of the resource
- error: there is no resource that has the specified id

## 3. borrow a resource

- input: resource id(string)
- action: check-out the resource that has the specified id to the current user  
loan period is different according to the user type  
( 'U': 14 days, 'G': 30 days, 'P': 90 days)  
display id, title, check-out date, return date of the resource
- error: the resource that has the specified id is not available  
there is no resource that has the specified id

## 4. return a resource

- input: resource id(string)
- action: return the resource that has the specified id to the current user  
return the copy with the earliest return date if the user borrowed multiple copies of the resource
- error: the resource that has the specified id is not available  
there is no resource that has the specified id

## 5. display check-out list

- input: no input
- action: list id, title, check-out date, return date, overdue days of resources which are borrowed and not returned by the current user yet

## 6. display check-out history

- input: no input
- action: list id, title, check-out date, return date of all resources which were borrowed by the current user

## 0. logout

- input: no input
- action: return to the login menu

Administrative user menu

## 1. add a resource

- input: id(string), type('B', 'J', 'M' – that means book, journal, and multimedia respectively), title(string), all authors(string separated by comma), publisher(string), year(number), pages(number), the number of copies(number), and additional information according to its type
- action: add the resource into the database
- error: the resource type input is not in ('B', 'J', 'M')  
an input field is not specified  
pages > 0, the number of copies > 0, year < current year

## 2. remove a resource

- input: resource id(string)
- action: delete the resource and all copies with the specified id
- error: there is no resource that has the specified id

## 3. list all resources checked out

- input: no input
- action: list id, copy#, title, check-out date, return date of resources which are borrowed by any user and not returned yet, along with id and name of their check-out user

## 4. list all resources overdue

- input: no input
- action: list id, copy#, title, check-out date, return date, overdue days of resources which are borrowed by any user and not returned by the return date, along with id and name of their check-out user

## 0. logout

- input: no input
- action: return to the login menu

**Submit**

## 1. Application program files

- source files, binary files, build files(makefile, ant, or batch script)
- A SQL file to create tables and integrity constraints in Tiberio RDBMS
- A SQL file to insert sample data for your test (about 5~10 rows for each table)

## 2. A document file containing the following:

- Description on your implementation and SQL statements
- Any assumption you made
- Development environment
- How to compile and run
- Test scenario and result
- Discussion or thoughts on your program

Please submit the files in .zip format with the filename [proj2\_student-id].zip via e-mail:

- To: twlee@idb.snu.ac.kr
- Title: [DB2013s\_proj2] your student-id, name
- Due date: 23:59pm, 21st May (late penalty: 10% per a day, no credit after a week)