January 2018 – created first instance of GUI window

-registered for a free mySQL server host

25 Jan 2018 – create a github repository and pushed existing project files to it

-created mySQL database including tables using phpMyAdmin

Data structure: ItemData -hold information about items currently in database: name, id, quantity, expiry date, last restocked

Orders: Order id, order date, item id, item quantity, standingorder?-restock the items

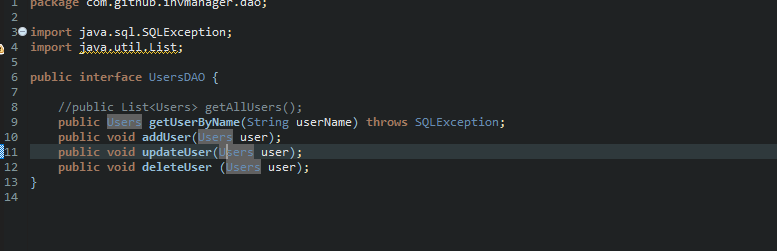
Users: User id, username, password, privilege

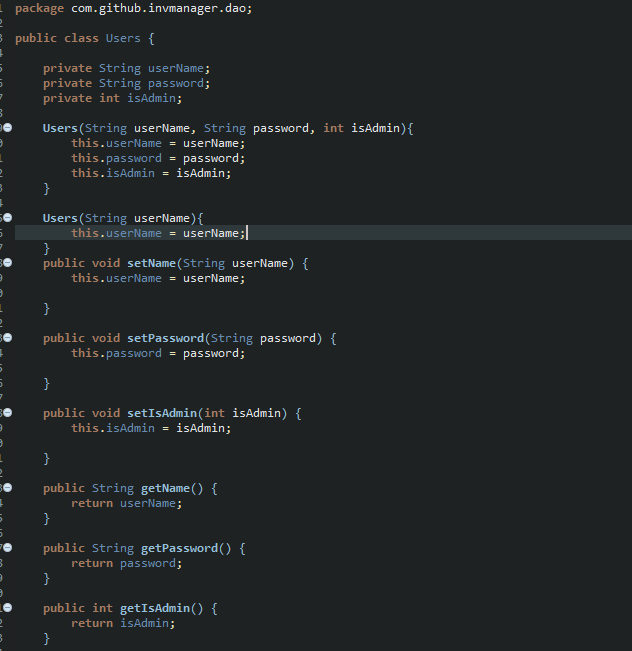
29/01/2018 start working on data access object implementation

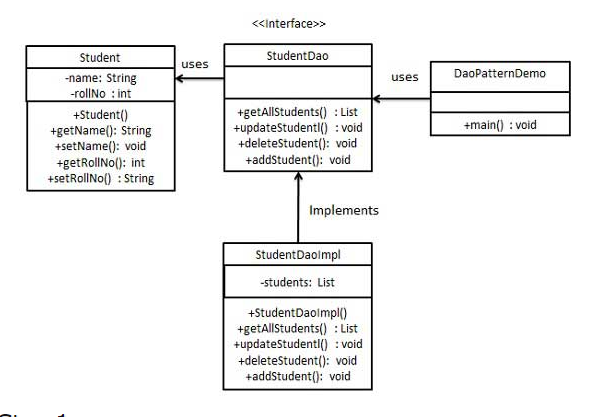
-first draft of the users DAO: interface, implementation and users class



Figure -Initial draft of the DAO Implementation generic class, has changed





Pattern example (change this diagram to correspond with actual implementation): 

**REMEMBER TO CREATE UML DIAGRAMS FOR THE PROGRAM…AT LEAST FOR THE DATA ACCESS OBJECT PATTERN AS EXAMPLE ABOVE**

01/02/18 – Users Data access layer takes shape, method to list all users is in place.

<https://ramj2ee.blogspot.co.uk/2013/08/data-access-object-design-pattern-or.html>

source for the DAL code

I am using the Apache DBCP library (3 libraries) to create connection pooling to save resources by not having to open new connections each time (not sure of explanation, google connection pooling)

Data access layer in place, all three tables in the database have their respective Data Access Objects created, code is roughly finished

Still need to add more update methods, as at the moment only Items and Orders have update methods, and only for one of the table rows

Code commented as well for the DAL

Testing is done using random data, as at the moment the actual item information is not available

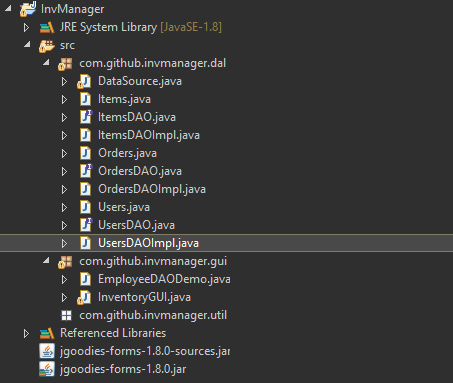


Figure - Screenshot of the Data Access Layer

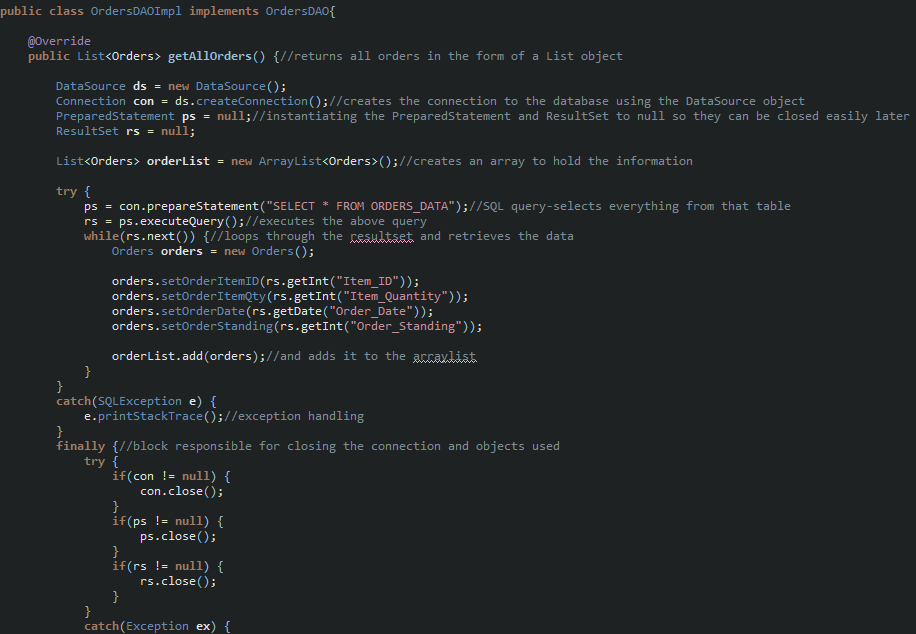


Figure -Actual implementation of the DAOImpl class

There is a bug—more of a nuisance, as I cant figure out how to insert a specific Date in the sql tables, all I get is the default 01-01-1970 values.

06/02/18 – linked DAL with the GUI, added ways of creating and retrieving the data objects on button presses. Still have the bug with the Date not being converted/inserted properly

-fixed date bug, code from IBM <https://www.ibm.com/support/knowledgecenter/en/SSEPGG_9.7.0/com.ibm.db2.luw.apdv.sample.doc/doc/java_jdbc/s-CreateEmployee-java.html>

Used a String to set the date value and converted to Date using Date.valueOf(String)

Works

Also had a small issue with retrieving an item by id or name, so I created array lists to hold the values, same as the getallitems method

07/02/18 – switched to a cardLayout and added another JPanel which contains the login data : username and password fields

Started work on a login method, implemented it successfully but without hashing the password

Ran into a problem related either to phpMyAdmin (the database manager) or the Data access layer of the program – its not possible to add more than 1 user to the database for some reason

Never mind the previous line-im just an idiot and wasn’t calling the line which was meant to add the actual data.

Login method works ¬¬ish not completely, as retrieving a single item from the database using a for loop is a bit confusing. The loop does not execute unless the parameters given to it are correct, which makes it hard to add error handling

<http://1bestcsharp.blogspot.co.uk/2015/06/java-create-login-form-window-mysql-database-in-java-netbeans.html>

source for a select statement which allows me to select a single row of data

08/02/18 – refined login method, with slightly better error handling and support for card scan login in the future

-created first draft of the barcode scanner app, atm it just scans a code and displays the data as text – next up I will have to make it so it sends the data to the database and also implement a new activity for scanning multiple items one after each other

-pushed the current project to github

-slightly adjusted gui, added a search bar and button to search the database for a specific item. The textfield also responds to enter so theres no need to press button to search

13/02/2018

-implemented sockets code to send a scan result from android application to main Java interface

-added sound to android project to beep on scan

-adjusted layouts for the android app, changed colour to light blue and switched from having all items in the main layout to using a content layout and using <include>

15/02/18

-switched from using google vision library for mobile scanning to using zxing mobile library

-source code : <https://github.com/rajdeol/android-barcode-scanner-bulk-scan-with-flash>

16/02/18

-imlemented run() method to automatically log in if the card is scanned, ran into problems

--the if statements to check if the input matches the database don’t get checked

--doesn’t work

-finally implemented Robot class to automatically press Enter to log in, but unfortunately only works with the card login, not username and password

-need to come up with something better

18/02/18

-creaeted a way of adding items to the database in a separate gui

-problem with the date format, it didn’t convert from string to sql date

-fixed by assigning the parameter from the textfield before adding the value to the item object

25/02/18

-looked up libraries with EAN data for barcode values

-tried outpan api, the service is down and therefore unable to give api key

-tried multiple services, unfortunately none worked

01/03/18

-found upcitemdb.com for product data api

-created JSON object retrieval class in android app to get the value from the api

-doesn’t work because the value returned by the API is stored in an JSON Array, and the class I created cant display the value

06/03/18

-started work on continuous scanning mode for app to add items to the database

-cannot get continuous mode from zxing to work

22/03/18

-fix bug where the scanner app crashes if theres no result returned from json by adding v.setLayerType(View.*LAYER\_TYPE\_SOFTWARE*, null);

To the get json method

Feature idea:

Scan a product, look it up in the database, and get past purchase values for it as well as weekly prediction for how much of it will sell using different algorithms

24/3/18

-added a second api access, to upcitemdb.com, to get better product information when scanning codes

25/03/18

-on Java program, when the password is scanned the system logs in automatically without the user having to press any key

-the password is hardcoded, could be hashed

Lesson learned:

\* Using a database driver to connect to an Internet server is bad practice, and should never be used, as I just found out

\* It causes latency and cannot deal with potential problems such as the server being down

\* Instead I should have opted for a SOAP or RESTful web service to communicate with the server

\* But I found this out late in the project and did not have time to redesign the whole DAL

\* Which is why the program is a lot slower than it should be