CPEN 312: OBJECT ORIENTED PROGRAMMING USING JAVA
Course Project

PROJECT 1: Automated Teller Machine

Using java swing design user interface for an Automated Teller Machine.
The application should have a screen that displays messages to the user, a keypad that receives numeric input from the user, a cash dispenser that dispenses cash to the user and a deposit slot that receives deposit envelopes from the user.
NOTE- You have been provided with sql statement (atm.sql) to populate the Accounts database for the GUI.

1. The screen displays a welcome message and prompts the user to enter an account number. Eg (Welcome Mr. Kwasi Adomako)
2. The user enters a five-digit account number using the keypad.
3. The screen prompts the user to enter the PIN (personal identification number) associated with the specified account number.
4. The user enters a five-digit PIN using the keypad.
5. If the user enters a valid account number and the correct PIN for that account, the screen displays the main menu. If the user enters an invalid account number or an incorrect PIN, the screen displays an appropriate message, then the ATM returns to Step 1 to restart the authentication process.
6. Use images of the GHC 5, 10, 20 and 50 notes. Simulate deposit and withdrawal using the images, that is when as user withdraws money there should be a simulation of GHC notes dispensed from the machine into a tray.

Your program should have the following menu and submenu:

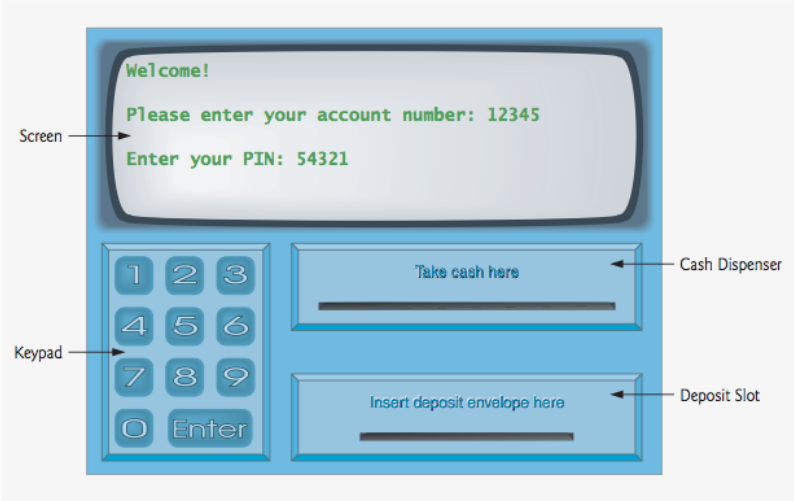
- **Cash deposit**
  - Place cash in deposit slot.
    - Cash has been accepted
    - Your new balance is…….
    - 0 – Back to Menu
- **Cash withdrawal**
  - 1 – GHC10 5 – GHC200
  - 2 – GHC20 6 – GHC300
  - 3 – GHC50 7 – GHC400
  - 4 – GHC100 8 - GHC500
  - 9 - Cancel transaction
    - Draw cash from dispenser
- **Balance Enquiry**
  - Your balance is ……..
  - 0- Back to Menu

The program should connect to a database (My SQL) to fulfill the above process using the following tables.

- Client table
- Accounts table
- Service type table

(Hint: Assume cash dispenser and deposit slot as buttons, a click on each will signify cash has been draw from dispenser and cash has been deposited in the deposit slot respectively)

Sample user interface



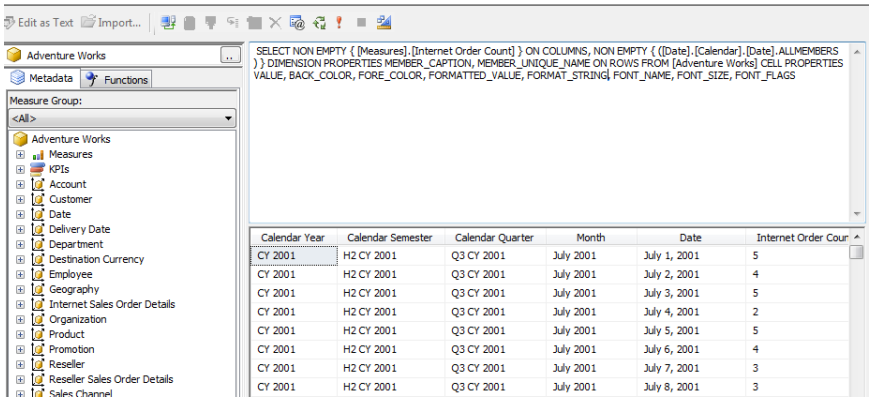
PROJECT 2. MySql Browser

You have given an existing database (MySQL server). Using java swing design a user interface for the MySql server**.** The application should have the following features:

- Mysql data driver upload
- Connect to mysql server
- Server login
- List all database in the server
- Execute Data Manipulation Language (DML) such as Insert, Delete, Merge, Select and Update on a selected database.
- Execute Data definition Language such (DDL) as Alter, Create, Disable, Truncate on the server.

NOTE- You have been provided with sql statement (atm.sql, reptut_film.sql and revenue.sql) to create and populate the sql server.

Sample user interface



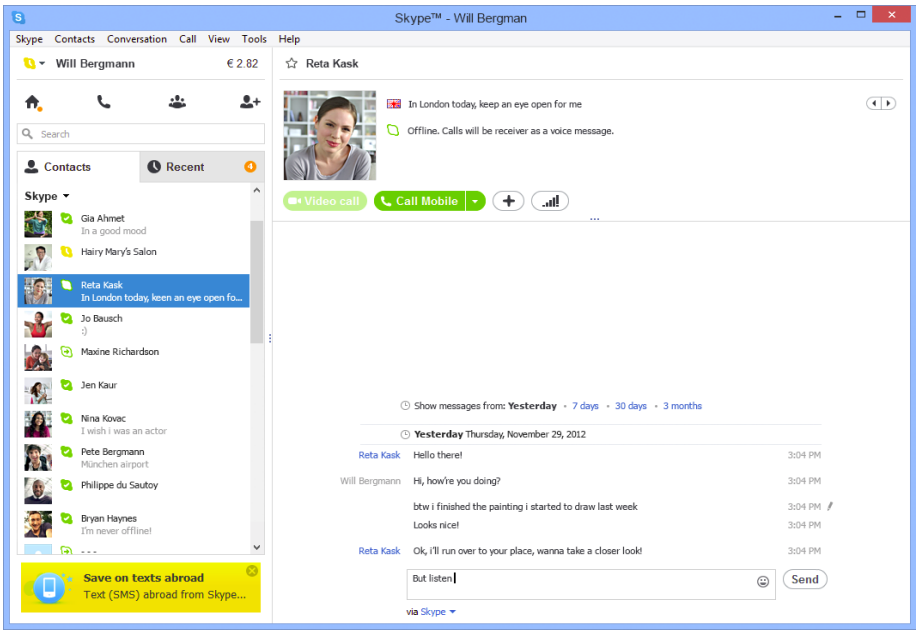
PROEJCT 3. Local Area Network (LAN) Chat System

Using java swing and sockets design user interface for Network Chat System. The application should have the following features:

- Login
- Server app
- Client app
- Send and receive smileys
- Notify with a sound when app is inactive and a message arrives
- Contact or friends list with avatar
- Conversation history

(Hint: Use sqllite for conversation history)

Sample user interface



PROJECT 4: A Game of Ludo

Using java swing design the graphics and logic of a Ludo game.
The application should have the following features:

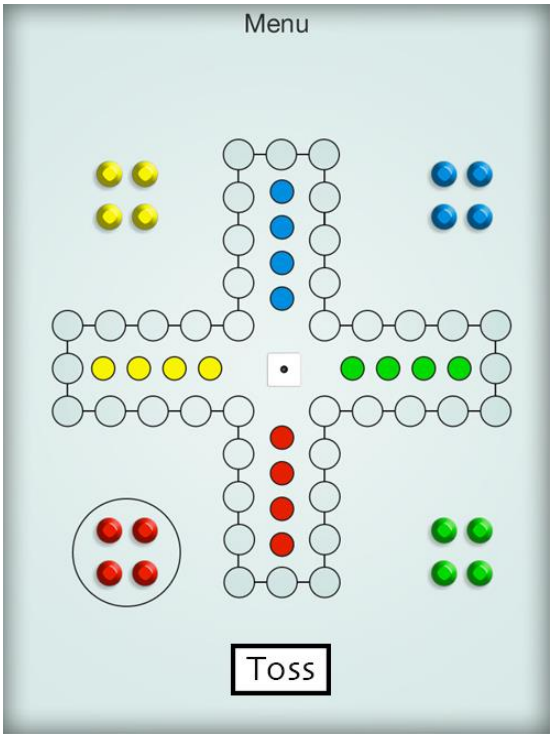
Implement the game play between a user and the CPU.
Four colors (Positions) from which a player can choose.
Four playing pieces for each color.
Implement a cell and a home for each color and link each cell to home with a path to complete the environment.
Implement a die using random number generator. The die should have six faces each representing numbers from 1~6.
Play audio tunes in the back ground.

For the basic rule, a user should be able to toss the die for the number 6 before he can have a chance to move a single playing piece into the environment, after which ever toss value of the die is used to count the piece through a path around the environment till it reaches home.
This is repeated on every toss of 6 for each playing piece until all pieces reaches home.

For every toss of die a user has to select which piece should move and in which direction assuming there are more than one piece on the playing environment.

A user and CPU play the game in turns. Who ever moves all four pieces from cell to home is the Winner for that game.

Sample user interface



PROJECT 5: An Elevator control System (Simulate) for Engineering Building

Using java swing design the graphics and logic of an Elevator control System.
The application should have the following features:

1. Simulate the motion of the elevator cat for a six-story building.
2. Functions of buttons, Opening and closing of elevator doors.

Use the following logic
If the elevator is on the floor 1 and the floor requested is the floor 1, then the elevator remains on the floor 1.
If the elevator is on the floor 1 and the floor requested is the floor 2, then the elevator is raised up 1 floor.
If the elevator is on the floor 1 and the floor requested is the floor 3, then the elevator is raised up 2 floors.
If the elevator is on the floor 2 and the floor requested is the floor 1, then the elevator is lowered down 1 floor.
If the elevator is on the floor 2 and the floor requested is the floor 2, then the elevator remains on the floor 2.
If the elevator is on the floor 2 and the floor requested is the floor 3, then the elevator is raised up 1 floor.
If the elevator is on the floor 3 and the floor requested is the floor 1, then the elevator is lowered down 2 floors.
If the elevator is on the floor 3 and the floor requested is the floor 2, then the elevator is lowered down 1 floor.
If the elevator is on the floor 3 and the floor requested is the floor 3, then the elevator remains on the floor 3…..
and so on.

Sample user interface

PROJECT 6: A game Oware

Using java swing design the graphics and logic of an Oware game.
The application should have the following features:

A graphical implementation of the board, seed and logic.
A two-player game between a user and CPU with the following rules and conditions.

Sowing

The game is starts by sowing, to sow take all the seeds of any of the holes opposite and close to the player and drop each seed out along the holes against the direction of the clockwise. If you reach the last hole of your ground you must continue in the land of the other player. Remember, you always have to drop out seeds in the direction against the clockwise.

Harvesting

If the last hole where you sow is in the land of the other player has two or three seeds take them out and of the hole (off the board) and keep them (harvest).

If the previous holes also contain two or three seeds also take them and remove all the seeds of your opponent that contains two or three seeds.

The player with the most harvested seed is the winner.

Conditions

1. To playing leaving the other player with no seeds to continue playing is not allowed. The user automatically loses the game
2. If the other player has only one seed in his field you will have to remove it in order to harvest and continue playing. This situation means that the other player will not be able keep playing.
   Players must provide in advance to avoid this situation. For example, having at least one seed in the last hole to harvest immediately to our opponent side and allow him to keep playing.
   If this is impossible, because we only have one seed in our land.
3. When there are few seeds left on the counter, the game may be perpetuating and hardly any of the 2 players can capture any new seed. By mutual agreement player can agree the end of the game. In this case every player is the owner of the seeds in his side. As always, who has gained more wins the match.

Sample user interface

PROJECT 7: Phone Simulator

Using java swing design a simulation of a smart phone.
The application should simulate the following features.

Home with icons
Time
Calendar
Phone call
SMS
Contacts
Call logs
Favorites
Contact search

(Hint: Use sqllite for Contacts, call logs and Favorites)

Sample user interface

PROJECT 8:Traffic Simulators

Using java swing design a simulation of a Crossed junction double road traffic situation.
The application should have the following features:

Logic of a crossed junction double road traffic situation with total of 10 traffic lights each with red, yellow and green lights.
For equal consideration each implement waiting and go time to have duration of 1min(For the simulation purposes).

1. All possible collusions should be avoided.
2. Implement graphical car with the ability to move and access all routs possible.
3. Generate a random motion for the cars to tread all possible routs.
4. Implement a real time transition of the traffic lights from green to yellow to red, which the cars will respond to.
5. With pedestrian crossing.

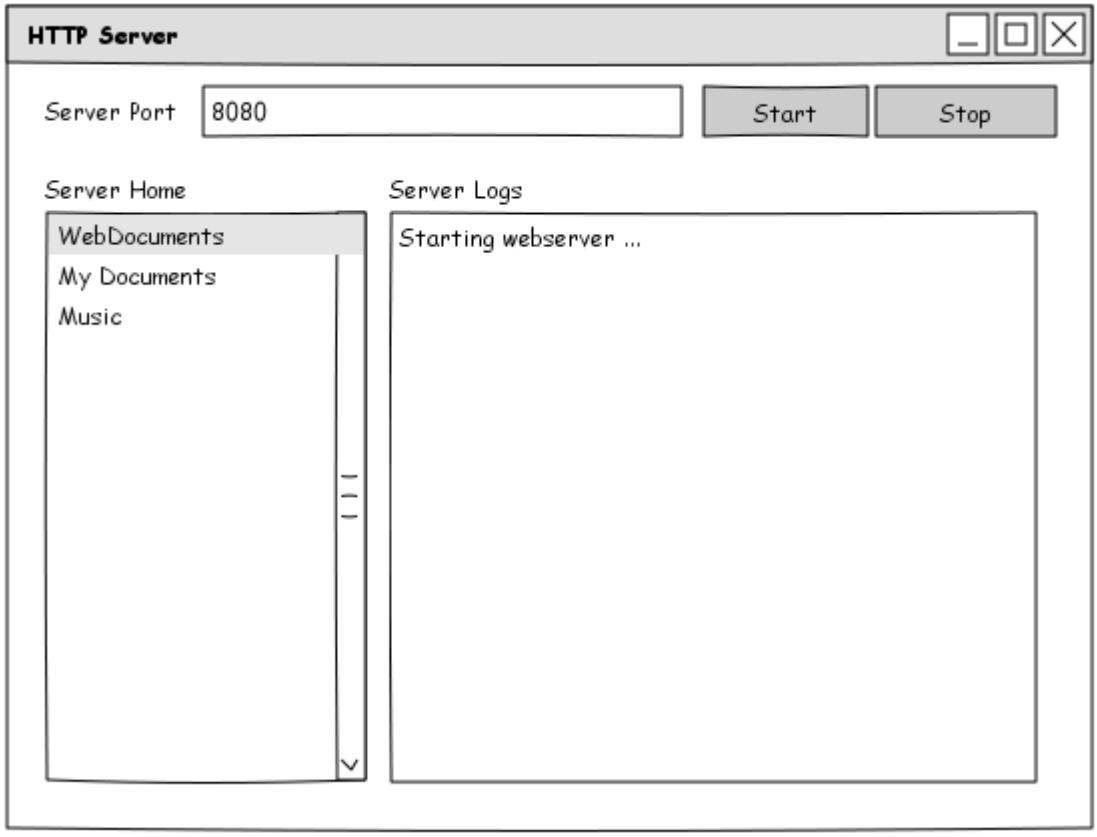
Sample user interface

PROJECT 9:HTTP Web Server

Using java swing and sockets, develop an HTTP web server for serving content to a web browser. The application should serve static content that is stored in a

directory. Your application should also have a user interface through which the end user can configure and control the web server:

- The root directory.
- The port.
- A button to start and stop the server.
- A read only text area to display the server status.



For simplicity sake, your server should only work with the HTTP GET verb and it should not generate static content. Your server implementation should not use any of Java's embedded web server classes.
Image