Assignments #4-5
CSCI 201 Fall 2018
8.5% of course grade

Title
Networked Hangman

Topics Covered
Databases and SQL
Networking
Multi-Threading
Synchronization

Introduction
This program will be an implementation of the popular game Hangman! We will be combining the last two assignments together, but you need to get started on the program as soon as possible since there are a lot of features and many other things happening throughout the remainder of the semester. There is a checkpoint about half-way through where there is certain functionality you should try to have completed by then.

Assignment
Hangman is a popular game where a user guesses letters that may be in a secret word. If the letter is in the secret word, the location of the letter is revealed. If it is not, the number of wrong guesses the user gets is decremented. If the user runs out of wrong guesses before guessing the secret word correctly, or if the user tries to guess the secret word and is incorrect, the user loses the game. If the user guesses the secret word before running out of wrong guesses, the user wins.

You will implement Hangman for this assignment in a multi-player and networked fashion. You could have from 1-4 players in the game. If there is more than one player, the game will end when one of the users guesses the secret word correctly or everyone has lost the game.

This assignment will be developed as a command line program. You will not need to have a GUI for this program.

Configuration File
When the program is executed, it will prompt the user to enter the name of a configuration file. If the configuration file cannot be located, you will display an error message stating:

```
Configuration file config.txt could not be found.
```

You will need to substitute in the proper configuration file name that could not be located.

Your program will then loop, prompting the user to enter another configuration file name. The configuration file will be formatted with one key and one value on each line, separated by an equal sign. In Java, look at the `Properties` class for an easy way to read and parse this file (though you could easily write a parser yourself as well). The configuration file will contain all the initialization parameters that you need to execute your program. Here is a sample configuration file.

```
ServerHostname=localhost
ServerPort=6789
DBConnection=jdbc:mysql://localhost:3309/hangman
DBUsername=root
DBPassword=root
SecretWordFile=hangmanwords.txt
```

We will use the same configuration file in both the client and server even though not all the parameters are required by both. The `ServerHostname` and `ServerPort` will be the only parameters necessary to read in the client. The `ServerPort`, `DBConnection`, `DBUsername`, `DBPassword`, and `SecretWordFile` will be read in the server. If any required parameter is missing, an error message should be displayed stating:

```
ServerHostname is a required parameter in the configuration
file.
```

You will need to substitute in the proper parameter that is missing. Once an appropriate configuration file is provided, print out all the parameters formatted as follows:

```
Server Hostname – localhost
Server Port – 6789
Database Connection String – jdbc:mysql://localhost:3309/hangman
Database Username – root
Database Password – root
Secret Word File – hangmanwords.txt
```

Initial Server Connection
After the configuration file has been read, your client needs to make a connection to the server. The server program will need to be run before your client can be run. If the client is unable to make a connection to the server, the client should terminate with an error message stating:

```
Trying to connect to server...Unable to connect to server
localhost on port 6789.
```

You will need to substitute in the proper hostname and port as passed in the configuration file.

Once a connection has been made to the server, you will display a message stating:

```
Trying to connect to server...Connected!
```

Server Execution
When the server is executed, you will prompt the user for the name of the configuration file, with similar behavior to what was explained in the "Configuration File" section above. The server will need to establish a connection with the database based on the configuration parameters. If the server is able to make a connection, the server will state:

```
Trying to connect to database...Connected!
```

If the server is unable to connect to the database, the server should terminate with an error message stating:

```
Trying to connect to database...Unable to connect to database
jdbc:mysql://localhost:3309/hangman with username root and
password root.
```

You will need to substitute in the proper database connection string, username, and password as passed in the configuration file.

The server will then output the status of everything that is happening in the game. Here is a list of all the output statements the server should have. Any parameters in angle brackets <> should be replaced by the appropriate variable when the program executes. There should also be a timestamp included at the beginning of each line in the format `HH:mm:ss.SSS`. For example, `18:20:55.478`.

1) `<username> - trying to log in with password <password>.`
2) `<username> - successfully logged in.`
3) `<username> - does not have an account so not successfully logged in.`
4) `<username> - has an account but not successfully logged in.`
5) `<username> - created an account with password <password>.`
6) `<username> - has record <numWins> wins and <numLosses> losses.`

7) `<username> - wants to start a game called <gameName>.`
8) `<username> - wants to join a game called <gameName>.`
9) `<username> - <gameName> already exists, so unable to start <gameName>.`
10) `<username> - successfully started game <gameName>.`
11) `<username> - successfully joined game <gameName>.`
12) `<username> - <gameName> exists, but <username> unable to join because maximum number of players have already joined <gameName>.`
13) `<username> - <gameName> needs <numPlayers> to start game.`
14) `<username> - <gameName> has <numPlayers> so starting game. Secret word is <secretWord>.`
15) `<gameName> <username> - guessed letter <letter>.`
16) `<gameName> <username> - <letter> is in <secretWord> in position(s) <positions>. Secret word now shows <secretWordDisplayed>.`
17) `<gameName> <username> - <letter> is not in <secretWord>. <gameName> now has <numGuessesRemaining> guesses remaining.`
18) `<gameName> <username> - guessed word <guessedWord>.`
19) `<gameName> <username> - <guessedWord> is incorrect. <username> has lost and is no longer in the game.`
20) `<gameName> <username> - <guessedWord> is correct. <username> wins game. <otherUsernames> have lost the game.`

Server Database

The database on the server will be used for user account information and statistics (i.e., number of wins and losses for a user). You can store additional information in the database if you want. The schema of the database is up to you, but you must provide a `database.sql` file in your project to define the database schema and populate it with any initial data that is needed. Place this file in the `src` directory of your program.

User Accounts
Once the client is connected to the server, the user will need to log in. The user will specify a username and password, which will be passed to the server. Each user who plays the game will need to have an account. Usernames and passwords are case-sensitive. If the username does not exist in the server's database, the user will be notified:

```
No account exists with those credentials.
```

The user will then be prompted to see if he would like to create that account. If so, the account will be created, and the user will automatically be logged into it. If the user does not want to have that account created, he will be prompted to enter another username and password.

Once the user is logged in, a message will be displayed stating:

```
Great! You are now logged in as Tommy!

Tommy's Record
--------------
Wins - 0
Losses - 0
```

You will need to substitute in the proper username, number of wins, and number of losses.

Starting and Joining Games
After logging in successfully, a user can start a new game or join an existing game. A menu with numbers will be provided to the user so he can specify what he wants to do:

```
1) Start a Game
2) Join a Game

Would you like to start a game or join a game?
```

Regardless which option is specified, the user will be prompted to enter the name of the game:

```
What is the name of the game?
```

Starting a Game
If the user is starting a new game, the name of the game cannot be in existence. Game names need to be unique at the time the game is being played. As soon as a game ends, the same game name can be reused. Game names have no restrictions on length or characters, so make sure to account for spaces and control characters. If the user is trying to start a game with a name that already exists (the name of the game is `OneGame` in this example), display a message stating:

```
OneGame already exists.
```

You will need to substitute in the proper game name. The program should then loop to allow the user to enter another game name.

After the user has specified an acceptable game name, he will be prompted to enter the number of users:

```
How many users will be playing (1-4)?
```

If the user enters a value other than 1, 2, 3, or 4, display a message stating:

```
A game can only have between 1-4 players.
```

The user should then be prompted to enter the number of users again.

Once an appropriate number of users is entered, the game will then wait for that many users to join. If there is more than one user in the game, display a message stating:

```
Waiting for 1 other user to join...
```

You will need to substitute in the proper number of additional users.

When a user joins, display a message with the name of the user and the user's record:

```
User Joe is in the game.

Joe's Record
-------------
Wins - 0
Losses - 15
```

You will need to substitute in the proper username and record.

Once all the required number of users have joined the game, display a message:

```
All users have joined.
```

The game will then move into the game play (see section below).

Joining a Game
If the user is joining a game, the name of the game must already exist, and there must be space for another user to join. If the name of the game does not exist, display a message stating:

```
There is no game with name OneGame.
```

You will need to substitute in the proper name of the game.

If the game does exist but there is no space for another user to join, display a message stating:

```
The game OneGame does not have space for another user to join.
```

You will need to substitute in the proper name of the game. Game names are all case-insensitive.

Once a user does join a game, the names and statistics of all the other users who are already in the game will be displayed:

```
User Tommy is in the game.

Tommy's Record
-------------
Wins - 1
Losses - 0
```

Once all the users have joined a game that are required, display a message:

```
All users have joined.
```

The game will then move into the game play (see section below).

Game Play
Once all the users that are required have joined the game, a message should be displayed stating:

```
Determining secret word...
```

The secret words will come from a file that is posted on the course web site called `hangmanwords.txt`. The location and name of the file for your program will be read from the configuration file, so you can place that file anywhere you want as long as your program can access it. The secret word should be randomly chosen from the file each time a new secret word needs to be determined. The file is formatted with one word on each line.

The program will then show the user the number of letters in the secret word, with each letter replaced by an underscore. The total number of incorrect guesses remaining will be shown, which initially starts at seven. The user will then be prompted to determine whether he wants to guess a letter or guess the word.

```
Secret Word _ _ _ _ _ _ _

You have 7 incorrect guesses remaining.
   1) Guess a letter.
   2) Guess the word.

What would you like to do?
```

If the user enters an option other than 1 or 2, display a message stating:

```
That is not a valid option.
```

Prompt him again to determine if he would like to guess a letter of guess the word. If the user wants to guess a letter, prompt him to enter a letter:

```
Letter to guess -
```

If a letter the user guesses is in the word, replace the proper underscores with the letter. For example, if the secret word was “traveler” and the user guessed ‘e’, the following message should be displayed:

```
The letter 'e' is in the secret word.

Secret Word _ _ _ _ E _ E _
```

All the letters in the secret word should be capitalized when displayed. However, all letters and words should be case-insensitive.

For single player games, this process continues until one of the following situations:

1. The user guesses seven incorrect letters, which would make him lose the game.
2. The user guesses the word correctly before guessing seven incorrect letters, making him win the game.
3. The user guesses the word incorrectly before guessing seven incorrect letters, making him lose the game.

For multi-player games, this process continues until one of the following situations:

1. The combination of all users guesses a total of seven incorrect letters, which would make all players lose the game.
2. Any user guesses the word correctly before the combination of all users guesses a total of seven incorrect letters, which make the user who guessed the word win and the other users lose.
3. If one user guesses the word incorrectly before the combination of all users guesses a total of seven incorrect letters, the user who guessed the word incorrectly loses and is out of the game. The other users of the game can continue playing until one of these three cases arises.
    a. If all of the users guess the word incorrectly before the combination of all users guesses a total of seven incorrect letters, all users will have lost and the game should terminate.

If the user wants to guess the word, prompt him to enter the secret word:

```
What is the secret word?
```

If he guesses the word correct, display a message letting him know that he wins the game:

```
That is correct! You win!
```

If he guesses the word incorrectly, display a message letting him know that he has lost the game, followed by the correct answer:

```
That is incorrect!  You lose!
The word was "traveler".
```

In either case, when the game ends, display the updated statistics for that user and all other users in the game. The current user's statistics should be displayed first.

```
Tommy's Record
--------------
Wins - 1
Losses - 0
```

Then always thank the user for playing your game before terminating the program.

```
Thank you for playing Hangman!
```

In multi-player games, one user at a time will guess a letter, starting with the host and proceeding in the order that users joined the game. When a user guesses a letter, the other users will be shown a message stating what that user guessed, whether the letter was in the word, and the updated secret word will be displayed.

When a player is waiting for another player to make a move, display a message:

```
Waiting for Joe to do something...
```

Once the user has made a move, display to the other players what he has done and the result. For example, if the user guessed 'r' and the letter 'r' is in the word, display:

```
Joe has guessed letter 'r'.

The letter 'r' is in the secret word.
```

Your program will continue in this manner for all of the users. Only the user whose turn it is will get to make a move. A small sample of this is below, with the output from two different users displayed.

```
Secret Word: _ _ A _ _ _ _ _

You have 7 incorrect guesses remaining.
Waiting for Joe to do something...



Joe has guessed letter 'r'.

The letter 'r' is in the secret word.

Secret Word: _ R A _ _ _ _ R

You have 7 incorrect guesses remaining.
   1) Guess a letter.
   2) Guess the word.

What would you like to do?
```

```
Secret Word: _ _ A _ _ _ _ _

You have 7 incorrect guesses remaining.
   1) Guess a letter.
   2) Guess the word.

What would you like to do? 1

Letter to guess - r

The letter 'r' is in the secret word.

Secret Word: _ R A _ _ _ _ R

You have 7 incorrect guesses remaining.
Waiting for Tommy to do something...
```

Single Player Sample Client Execution
Here is a sample execution of your Hangman program with one user and the user input bolded (though this will not be the case when you run your program).

```
What is the name of the configuration file? config.txt
Reading config file...
Server Hostname – localhost
Server Port – 6789
Database Connection String – jdbc:mysql://localhost:3309/hangman
Database Username – root
Database Password – root
Secret Word File – hangmanwords.txt

Trying to connect to server...Connected!

Username: Tommy
Password: Trojan

No account exists with those credentials.
Would you like to create a new account? Yes
Would you like to use the username and password above? Yes

Great! You are now logged in as Tommy.

Tommy's Record
--------------
Wins – 0
Losses – 0

   1) Start a Game
   2) Join a Game

Would you like to start a game or join a game? 1

What is the name of the game? OneGame

How many users will be playing (1-4)? 1

All users have joined.

Determining secret word...

Secret Word _ _ _ _ _ _ _

You have 7 incorrect guesses remaining.
   1) Guess a letter.
   2) Guess the word.

What would you like to do? 1

Letter to guess – a

The letter 'a' is in the secret word.

Secret Word: _ _ A _ _ _ _ _
```

```
You have 7 incorrect guesses remaining.
   1) Guess a letter.
   2) Guess the word.

What would you like to do? 1

Letter to guess - o

The letter 'o' is not in the secret word.

Secret Word: _ _ A _ _ _ _ _

You have 6 incorrect guesses remaining.
   1) Guess a letter.
   2) Guess the word.

What would you like to do? 1

Letter to guess - t

The letter 't' is in the secret word.

Secret Word: T _ A _ _ _ _ _

You have 6 incorrect guesses remaining.
   1) Guess a letter.
   2) Guess the word.

What would you like to do? 1

Letter to guess - e

The letter 'e' is in the secret word.

Secret Word: T _ A _ E _ E _

You have 6 incorrect guesses remaining.
   1) Guess a letter.
   2) Guess the word.

What would you like to do? 1

Letter to guess - m

The letter 'm' is not in the secret word.

Secret Word: T _ A _ E _ E _

You have 5 incorrect guesses remaining.
   1) Guess a letter.
   2) Guess the word.

What would you like to do? 1

Letter to guess - r

The letter 'r' is in the secret word.
```

```
Secret Word: T R A _ E _ E R

You have 5 incorrect guesses remaining.
   1) Guess a letter.
   2) Guess the word.

What would you like to do? 2

What is the secret word? traveler

That is correct! You win!

Tommy's Record
--------------
Wins - 1
Losses - 0

Thank you for playing Hangman!
```

Multi-Player Sample Client Execution

Here is a sample execution of your Hangman program with two users and the user input bolded.

```
What is the name of the configuration file?
config.txt
Reading config file...
Server Hostname - localhost
Server Port - 6789
Database Connection String -
jdbc:mysql://localhost:3309/hangman
Database Username - root
Database Password - root
Secret Word File - hangmanwords.txt

Trying to connect to server...Connected!

Username: Tommy
Password: Trojan

Great! You are now logged in as Tommy.

Tommy's Record
--------------
Wins - 1
Losses - 0

   1) Start a Game
   2) Join a Game

Would you like to start a game or join a
game? 1

What is the name of the game? MyGame

How many users will be playing (1-4)? 2

Waiting for 1 other user to join...



User Joe is in the game.

Joe's Record
--------------
Wins - 0
Losses - 15

All users have joined.

Determining secret word...

Secret Word _ _ _ _ _ _ _

You have 7 incorrect guesses remaining.
   1) Guess a letter.
   2) Guess the word.

What would you like to do? 1
```

```
What is the name of the configuration file?
config.txt
Reading config file...
Server Hostname - localhost
Server Port - 6789
Database Connection String -
jdbc:mysql://localhost:3309/hangman
Database Username - root
Database Password - root
Secret Word File - hangmanwords.txt

Trying to connect to server...Connected!

Username: Joe
Password: Bruin

Great! You are now logged in as Joe.

Joe's Record
--------------
Wins - 0
Losses - 15

   1) Start a Game
   2) Join a Game

Would you like to start a game or join a
game? 2









What is the name of the game? MyGame

User Tommy is in the game.

Tommy's Record
--------------
Wins - 1
Losses - 0

All users have joined.

Determining secret word...

Secret Word _ _ _ _ _ _ _

You have 7 incorrect guesses remaining.
Waiting for Tommy to do something...
```

```
Letter to guess – a

The letter 'a' is in the secret word.

Secret Word: _ _ A _ _ _ _ _

You have 7 incorrect guesses remaining.
Waiting for Joe to do something...



Joe has guessed letter 'r'.

The letter 'r' is in the secret word.

Secret Word: _ R A _ _ _ _ R

You have 7 incorrect guesses remaining.
   1) Guess a letter.
   2) Guess the word.

What would you like to do? 1

Letter to guess – o

The letter 'o' is not in the secret word.

Secret Word: _ R A _ _ _ _ R

You have 6 incorrect guesses remaining.
Waiting for Joe to do something...



Joe has guessed letter 'z'.

The letter 'z' is not in the secret word.

Secret Word: _ R A _ _ _ _ R

You have 5 incorrect guesses remaining.
   1) Guess a letter.
   2) Guess the word.

What would you like to do? 2

What is the secret word? traveler

That is correct! You win!
```

```
Tommy has guessed letter 'a'.

The letter 'a' is in the secret word.

Secret Word: _ _ A _ _ _ _ _

You have 7 incorrect guesses remaining.
   1) Guess a letter.
   2) Guess the word.

What would you like to do? 1

Letter to guess – r

The letter 'r' is in the secret word.

Secret Word: _ R A _ _ _ _ R

You have 7 incorrect guesses remaining.
Waiting for Tommy to do something...



Tommy has guessed letter 'o'.

The letter 'o' is not in the secret word.

Secret Word: _ R A _ _ _ _ R

You have 6 incorrect guesses remaining.
   1) Guess a letter.
   2) Guess the word.

What would you like to do? 1

Letter to guess – z

The letter 'z' is not in the secret word.

Secret Word: _ R A _ _ _ _ R

You have 5 incorrect guesses remaining.
Waiting for Tommy to do something...



Tommy has guessed the word 'traveler'.

Tommy guessed the word correctly. You lose!
```

```
Tommy's Record
--------------
Wins - 2
Losses - 0

Joe's Record
--------------
Wins - 0
Losses - 16

Thank you for playing Hangman!
```

```
Joe's Record
--------------
Wins - 0
Losses - 16

Tommy's Record
--------------
Wins - 2
Losses - 0

Thank you for playing Hangman!
```

Grading Criteria

Configuration File (0.5%)

Single Player Game Play (1.5%)

Multi-Player Game Play (2.0%)

Server Output (1.5%)

Database (1.0%)

Secret Word File (0.5%)

Starting a Game (0.5%)

Joining a Game (0.5%)

User Accounts (0.5%)