TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN/ĐỒ ÁN CUỐI KÌ MÔN KIẾN TẬP CÔNG NGHIỆP**

**TÌM HIỂU VỀ JAVA VÀ THIẾT KẾ HỆ THỐNG ĐẶT VÉ XEM PHIM**

*Người hướng dẫn*: **THẦY DUNG CẨM QUANG**

**CB NGUYỄN HỮU NHẬT TÂN**

*Người thực hiện*: **VŨ MẠNH ĐỨC – 51703062**

Lớp **: 17050301**

Khoá  **: 21**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2020**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN/ĐỒ ÁN CUỐI KÌ MÔN KIẾN TẬP CÔNG NGHIỆP**

**TÌM HIỂU VỀ JAVA VÀ THIẾT KẾ HỆ THỐNG ĐẶT VÉ XEM PHIM**

Người hướng dẫn: **DUNG CẨM QUANG**

**CB NGUYỄN HỮU NHẬT TÂN**

Người thực hiện: **VŨ MẠNH ĐỨC**

Lớp **: 17050301**

Khoá  **: 21**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2020**

LỜI CẢM ƠN

Khoảng thời gian thực tập em được tiếp cận nhiều với những kiến thức thực tế, cán bộ hướng dẫn tại công ty nhiệt tình giúp đỡ em trong suốt quá trình thực tập. Em xin cảm ơn cán bộ hướng dẫn đã chỉ bảo em và thầy Dung Cẩm Quang đã hỗ trợ, chỉ bảo em trong quá trình thực tập và báo cáo.l

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi / chúng tôi và được sự hướng dẫn của TS Nguyễn Văn A;. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày tháng năm*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Vũ Mạnh Đức*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

Quá trình thực tập em được training kĩ càng về Java và CSDL MariaDB và cách kết nối hệ CSDL đó với ứng dụng Java bằng JDBC và có một sản phẩm demo nhỏ về những gì đã được học tập tại công ty.

Bài báo cáo gồm 3 chương:

Chương 1: Ngôn ngữ Java

Chương 2: Cơ sở dữ liệu MariaDB

Chương 3: Kết nối CSDL MariaDB với Java JDBC

Chương 4: Nghiên cứu thực nghiệm

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc46738812)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#_Toc46738813)

[TÓM TẮT iv](#_Toc46738814)

[MỤC LỤC 1](#_Toc46738815)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 3](#_Toc46738816)

[CHƯƠNG 1 – NGÔN NGỮ JAVA 4](#_Toc46738817)

[1.1 Thuật ngữ Java 4](#_Toc46738818)

[1.2 Các tính năng của Java 6](#_Toc46738819)

[CHƯƠNG 2 – CƠ SỞ DỮ LIỆU MARIADB 8](#_Toc46738820)

[2.1 Sơ lược về MariaDB 8](#_Toc46738821)

[2.2 Ưu điểm 8](#_Toc46738822)

[2.3 Cài đặt MariaDB 9](#_Toc46738823)

[CHƯƠNG 3 – KẾT NỐI CSDL MARIADB VỚI JAVA JDBC 14](#_Toc46738824)

[3.1 JDBC là gì 14](#_Toc46738825)

[3.2 Các thành phần của JDBC 15](#_Toc46738826)

[3.3 Kết nối Database với Java JDBC 16](#_Toc46738827)

**DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT**

**CÁC KÝ HIỆU**

*f Tần số của dòng điện và điện áp (Hz)*

*p Mật độ điện tích khối (C/m3)*

**CÁC CHỮ VIẾT TẮT**

CSDL Cơ sở dữ liệu

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

**DANH MỤC HÌNH**

[Hình 1: Chọn hệ điều hành windows 64-bit 10](#_Toc46738828)

[Hình 2: Click vào ứng dụng vừa tải về và tiến hành cài đặt 10](#_Toc46738829)

[Hình 3: Tiến hành cài đặt 11](#_Toc46738830)

[Hình 4: Nhập mật khẩu cho database 11](#_Toc46738831)

[Hình 5: Đặt tên cho Services và port sẽ là 3306 12](#_Toc46738832)

[Hình 6: Cài đặt kết nối tới MariaDB 12](#_Toc46738833)

[Hình 7: Giao diện HeidiSQL dùng để kết nối và truy vấn trong MariaDB. 13](#_Toc46738834)

[Hình 8: Mô hình kết nối ứng dụng Java với CSDL 14](#_Toc46738835)

[Hình 9: Download driver 16](#_Toc46738836)

[Hình 10: Configure Build Path 17](#_Toc46738837)

[Hình 11: Thêm driver vừa tải về 17](#_Toc46738838)

[Hình 12: Cấu hình kết nối tới database 19](#_Toc46738839)

[Hình 13: tạo kết nối 19](#_Toc46738840)

[Hình 14: Tạo câu lệnh truy vấn 19](#_Toc46738841)

[Hình 15: Cấu hình kết nối tới database 21](#_Toc46738842)

[Hình 16: Lớp CustomerEntity (chỉ hiển thị thuộc tính) 21](#_Toc46738843)

[Hình 17: Class CustomerRepo 22](#_Toc46738844)

[Hình 18: Class Main 23](#_Toc46738845)

CHƯƠNG 1 – NGÔN NGỮ JAVA

Java được phát triển bởi James Gosling và Patrick Naughton ở Sun Mircosystems Inc vào năm 1991 và sau đó được tập đoàn Oracle mua lại. Là một ngôn ngữ lập trình đơn giản, viết, biên dịch và gỡ lỗi một chương trình rất dễ dàng.

Java được tạo ra với tiêu chí “Viết một lần, thực thi khắp nơi” (“Write Once, Run Anywhere”). Chương trình phần mềm viết bằng Java có thể chạy trên mọi nền tảng khác nhau thông qua một môi trường thực thi với điều kiện có môi trường thực thi thích hợp trên nền tảng đó.

1.1 Thuật ngữ Java

Trước hết ta sẽ làm quen với các khái niệm trong Java:

1. *Máy ảo JVM*

Máy ảo Java (JVM): Bao gồm các giai đoạn viết chương trình, biên dịch và chạy chương trình.

1. Viết chương trình: Được thực hiện bởi lập trình viên Java
2. Quá trình biên dịch: Chương trình này được thực hiện bởi trình biên dịch javac, javac là trình biên dịch java chính có trong bộ công cụ phát triển Java (JDK). Nó lấy chương trình java làm đầu vào và tạo mã byte java làm đầu ra.
3. Chạy chương trình: JVM thực thi mã byte được tạo bởi trình biên dịch. Đây được gọi là giai đoạn chạy chương trình.

Chức năng chính của JVM là thực thi mã byte do trình biên dịch tạo ra. Mỗi hệ điều hành có JVM khác nhau, tuy nhiên đầu ra mà chúng tạo ra sau khi thực thi mã byte là giống nhau trên tất cả các hệ điều hành. Đó là lý do tại sao gọi java là ngôn ngữ độc lập nền tảng.

1. *Mã byte*

Trình biên dịch javac của JDK biên dịch mã nguồn java thành mã byte để nó có thể được thực thi bởi JVM. Mã byte được lưu trong tệp. Class bởi trình biên dịch.

1. *Java development kit (JDK)*

Như tên cho thấy đây là bộ công cụ phát triển java hoàn chỉnh bao gồm JRE (Java Runtime Environment), trình biên dịch và các công cụ khác nhau như JavaDoc, trình gỡ lỗi Java,… Để tạo, biên dịch và chạy chương trình Java, sẽ cần phải đặt JDK trên máy tính của mình.

1. *Java runtime environment (JRE)*

Môi trường thời gian thực thi java(JRE) là một phần của JDK, có nghĩa là JDK bao gồm cả JRE. Khi đã cài đặt JRE trên hệ thống của mình, có thể chạy một chương trình java tuy nhiên lại không thể biên dịch được. JRE bao gồm JVM, plugin trình duyệt và hỗ trợ applet. Khi chỉ cần chạy một chương trình java trên máy tính của mình chỉ cần JRE.

1.2 Các tính năng của Java

1. *Ngôn ngữ độc lập nền tảng*

Java là ngôn ngữ độc lập với nên tảng trình biên dịch (javac) chuyển đổi mã nguồn (.java) thành mã byte (. class). Như đã đề cập ở trên, JVM thực thi mã byte do trình biên dịch tạo ra. Mã byte này có thề chạy trên bất kỳ nền tảng nào như Windows, Linux, Mac OS,…Điều đó có nghĩa là một chương trình được biên dịch trên windows có thể chạy trên linux và ngược lại. Mỗi hệ điều hành có JVM khác nhau, tuy nhiên đầu ra mà chúng tạo ra sau khi thực thi mã byte là giống nhau trên tất cả các hệ điều hành. Đó là lý do tại sao gọi java là ngôn ngữ độc lập nền tảng.

1. *Ngôn ngữ hướng đối tượng*

Java là ngôn ngữ hướng đối tượng lập trình hướng đối tượng là một cách tổ chức các chương trình dước dạng tập hợp các đối tượng, mỗi đối tượng đại diện cho một thể hiện của một lớp.

Bốn khái niệm chính của lập trình hướng đối tượng là:

* Abstraction
* Encapsulation
* Inheritance
* Polymorphism

1. *Đơn giản*

Java được coi là một trong những ngôn ngữ đơn giản vì nó không có các tính năng phức tạp như nạp chồng toàn tử, nhiều kế thừa, con trỏ và cấp phát bộ nhớ rõ ràng.

1. *Mạnh mẽ*

Java được phát triển theo cách tập trung vào việc kiểm tra sớm các lỗi có thể xảy ra, đó là lý do tại sao trình biên dịch Java có thể phát hiện trong các ngôn ngữ lập trình khác. Các tính năng chính của Java làm cho nó mạnh mẽ là thu gom rác, xử lý ngoại lệ và cấp phát bộ nhớ.

1. *Bảo mật*

Không có con trỏ và không thể truy cập vào các mảng bị ràng buộc (xảy ra lỗi ngoại lệ ArrayIndexOutOfBoundException nếu ta cố gắng làm như vậy) trong java. Đó là lý do tại sao mộ số lỗi bảo mật như lạm dụng ngăn xếp hoặc tràn bộ nhớ đệm là không thể khai thác trong Java.

1. *Phân tán*

Chúng ta có thể tạo các ứng dụng phân tán. RMI (Remote Method Invocation) và EJB (Enterprise Java Bean) được sử dụng để tạo các ứng dụng phân tán trong java. Nói một cách đơn giản: các chương trình java có thể được phân phối trên nhiều hệ thống được kết nối với nhau bằng kết nối Internet. Các đối tượng trên một JVM (máy ảo java)có thể thực hiện các thủ tục trên một JVM từ xa

1. *Đa luồng*

Java hỗ trợ đa luồng. Đa luồng là một tính năng java cho phép thực thi đồng thời hai hoặc nhiều phần của chương trình để sử dụng tối đa CPU.

1. *Khả chuyển*

Mã java được viết trên một máy có thể chạy trên máy khác. Mã byte độc lập nền tảng có thể được mang đến bất kỳ nền tảng nào để thực thi làm cho mã java linh hoạt.

CHƯƠNG 2 – CƠ SỞ DỮ LIỆU MARIADB

2.1 Sơ lược về MariaDB

[MariaDB](https://mariadb.org/) là một hệ quản trị cơ sở dữ liệu mã nguồn mở (DBMS) được xây dựng bởi một số tác giả sáng lập MySQL và hiện tại đang được đông đảo cộng đồng các nhà phát triển sử dụng.

MariaDB là một hệ thống quản lý cơ sở dữ liệu quan hệ mã nguồn mở , một sự thay thế tương thích cho công nghệ cơ sở dữ liệu MySQL được sử dụng rộng rãi.

MariaDB có sẵn để cài đặt trên[Linux](https://blog.tinohost.com/he-dieu-hanh-linux-la-gi/) [CentOS](https://blog.tinohost.com/centos-la-gi-tim-hieu-ve-he-dieu-hanh-centos/), [Ubuntu](https://blog.tinohost.com/he-dieu-hanh-ubuntu/) và [Window](https://blog.tinohost.com/he-dieu-hanh-windows-la-gi-uu-va-khuyet-diem/). MariaDB được phát triển nhằm thay thế công nghệ cơ sở dữ liệu MySQL, vì thế nó tương thích và cho một hiệu suất cao hơn so với MySQL.

MariaDB có các phiên bản cho các hệ điều hành khác nhau: Windows, Linux,.. với các gói cài đặt tar, zip, MSI, rpm cho cả 32bit và 64bit. Hiện tại phiên bản mới nhất của MariaDB là 10.4.7.

2.2 Ưu điểm

Đây là các ưu điểm khi sử dụng hệ CSDL MariaDB

Mã nguồn mở: Trái ngược với MySQL, MariaDB hoàn toàn mã nguồn mở được vận hành bởi cộng đồng thực sự (MariaDB Foundation), đứng đằng sau là công ty Monty Program, mặc dù cũng có bản thương mại nhưng không ảnh hưởng đến phiên bản mã nguồn mở.

Được bảo trì bởi người tạo ra MySQL: Sau khi bị Oracle mua lại, một số thành viên chủ lực của MySQL đã ra đi, trong đó bao gồm những người tạo ra MySQL. Trong khi đó, MariaDB tiếp tục phát triển, đến bây giờ đã phát triển đến thế hệ 10.x. Do đó, tất cả những nhược điểm của MySQL đều được khắc phục ở MariaDB.

#### Hoàn toàn tương thích với MySQL: Do “sinh cùng cha” nên MariaDB có hệ thống hoàn toàn tương thích như với MySQL từ phiên bản 5.1 -> 5.5. Do đó, nếu Website của bạn đang dùng MySQL 5.1 -> 5.5 thì việc chuyển qua MariaDB là cực kì dễ dàng.Sự sự tương thích giữa MariaDB và MySQL nên trong hầu hết trường hợp chúng ta có thể xóa bỏ MySQL và cài đặt MariaDB để thay thế mà hệ thống vẫn hoạt động bình thường.

Nhiều Engine lưu trữ: Ngoài việc hỗ trợ các storage engines cơ bản như MyISAM, BLACKHOLE, CSV, MEMORY, ARCHIVE, and MERGE thì trên MariaDB còn bổ sung thêm các storage engines sau: Aria (được xem như một phiên bản cập nhập của MyISAM), XtraDB (thay thế cho InnoDB), FederatedX, OQGRAPH, SphinxSE, IBMDB2I, Spider, PBXT, ….

Kết hợp 2 thế giới SQL và NoSQL: Cả 2 loại cơ sở dữ liệu này đều có điểm lợi riêng, việc MariaDB cho phép kết nối sẽ tận dụng được lợi thế của 2 thế giới này: Dynamic Column và Cassandra Store Engine.

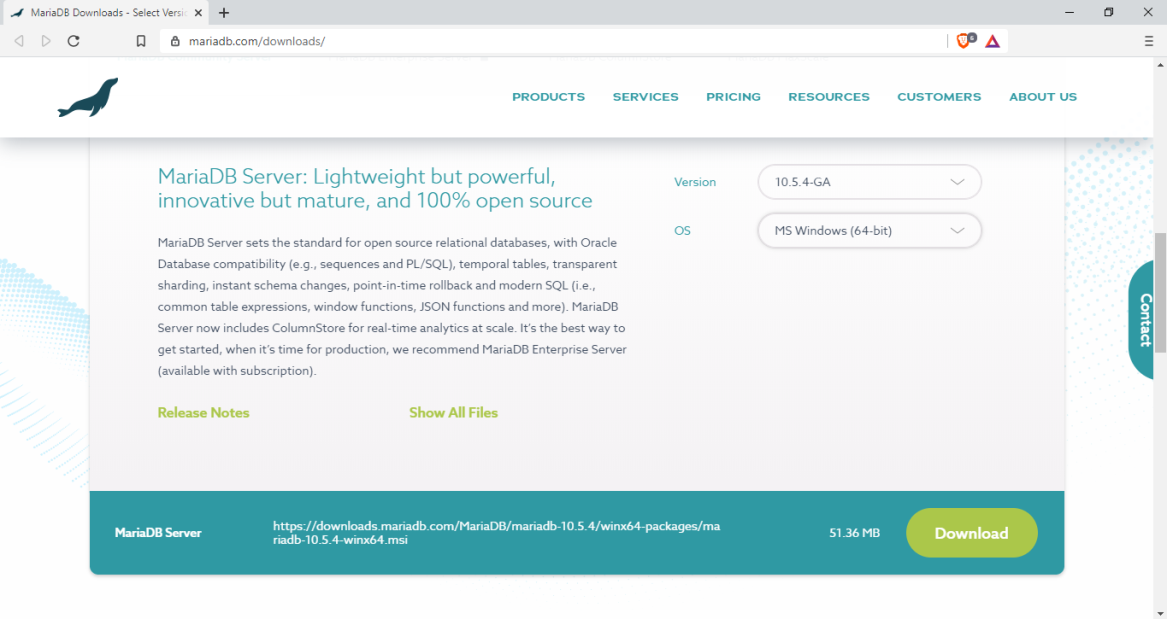
Tối ưu tốc độ: Trong cùng một điều kiện như nhau thì MariaDB nhanh hơn MySQL từ 3-5% tốc độ.

Các Website lớn đã sử dụng MariaDB: Một trong những website lớn nhất thế giới là [Wikipedia](https://www.wikipedia.org/) đã chuyển từ MySQL qua MariaDB. Ngay cả Fullstack-Station cũng đang dùng MariaDB.

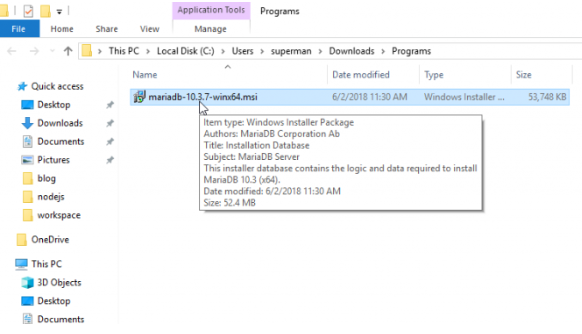
2.3 Cài đặt MariaDB

Có khá nhiều phiên bản tương thích với các hệ điều hành khác nhau. Trong bài báo cáo này sẽ hướng dẫn cài đặt ở hệ điều hành Windows 64bit.

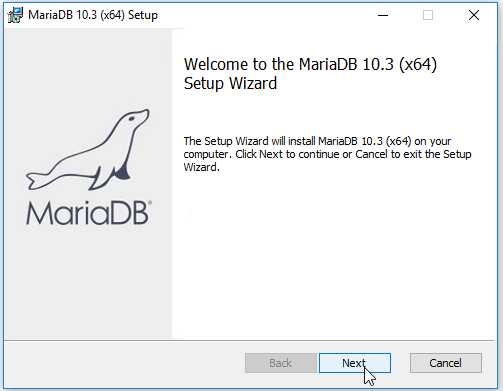
Truy cập vào trang web [*https://mariadb.com/downloads/*](https://mariadb.com/downloads/) và tiến hành tải về



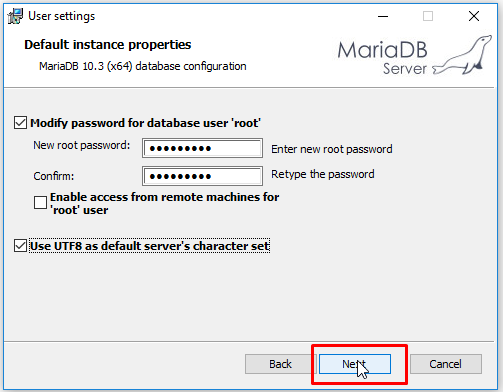
Hình 1: Chọn hệ điều hành windows 64-bit



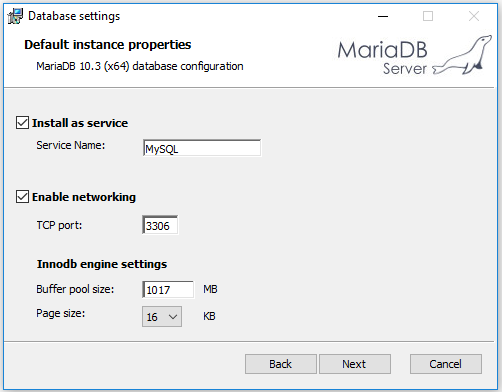
Hình 2: Click vào ứng dụng vừa tải về và tiến hành cài đặt



Hình 3: Tiến hành cài đặt

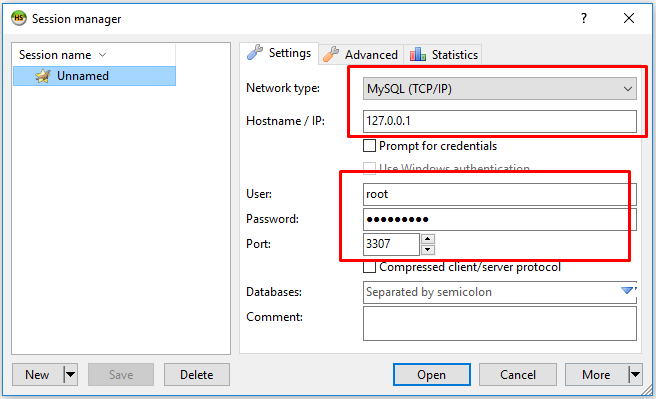


Hình 4: Nhập mật khẩu cho database



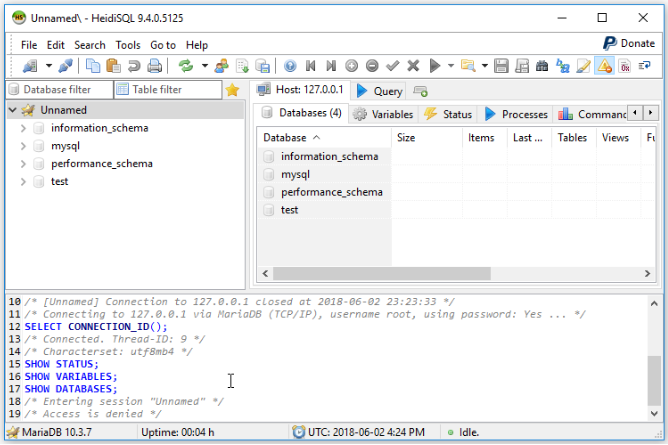
Hình 5: Đặt tên cho Services và port sẽ là 3306

Sau đó bấm next, tới khi cài đặt xong ta mở phần mềm HeidiSQL để kết nối tới MariaDB.



Hình 6: Cài đặt kết nối tới MariaDB

Sử dụng tài khoản root, mật khẩu tạo ở trên hình 5 và cổng 3306 cài đặt ở hình để kết nối tới CSDL

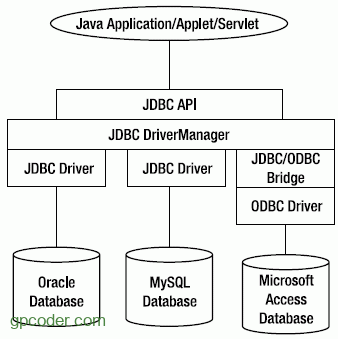


Hình 7: Giao diện HeidiSQL dùng để kết nối và truy vấn trong MariaDB.

CHƯƠNG 3 – KẾT NỐI CSDL MARIADB VỚI JAVA JDBC

3.1 JDBC là gì

**JDBC (Java Database Connectivity)**là một API chuẩn dùng để tương tác với các loại cơ sở dữ liệu quan hệ (database relationship). JDBC bao gồm một tập hợp các class và các interface dùng cho ứng dụng Java có thể giao tiếp với các cơ sở dữ liệu (database) khác nhau.

[](https://gpcoder.com/wp-content/uploads/2019/09/java-jdbc-components.gif)

Hình 8: Mô hình kết nối ứng dụng Java với CSDL

JDBC có thể làm việc với bất kỳ cơ sở dữ liệu ([MySQL](https://www.mysql.com/), [PostgreSQL](https://www.postgresql.org/), [Oracle](https://www.oracle.com/), [SQL Server](https://www.microsoft.com/en-us/sql-server/sql-server-downloads), …) dựa vào một **Driver** được cung cấp.

JDBC API bao gồm hai package chính:

* **java.sql** : là một phần của Java standard.
* **javax.sql** : là một phần của Java enterprise.

3.2 Các thành phần của JDBC

* **DriverManager :** là một class quản lý danh sách các Driver (database drivers). Các yêu cầu kết nối từ ứng dụng Java sẽ được class này tìm kiếm Driver phù hợp đầu tiên để thiết lập kết nối với cơ sở dữ liệu.
* **Driver:** là một interface dùng để xử lý các giao tiếp với cơ sở dữ liệu. Thông thường ứng dụng Java sẽ không giao tiếp trực tiếp với class này mà thông qua DriverManager.
* **Connection :** là một interface cung cấp tất cả các method cần thiết cho việc giao tiếp với database. Interface này chứa nhiều phương thức đa dạng để tạo kết nối với một Database. Tất cả các thông tin giao tiếp với cơ sở dữ liệu chỉ có thể thông qua đối tượng Connection. Một Connection đại diện cho một phiên (session) làm việc với cơ sở dữ liệu.
* **Statement :** là một interface cho phép gửi các câu lệnh SQL tới Database. Ngoài ra, một số Interface kết thừa từ nó cung thêm các tham số để thực thi các thủ tục đã được lưu trữ (stored procedure).
* **ResultSet :** đại diện cho tập hợp các bản ghi (record) có được sau khi thực hiện truy vấn (query).
* **SQLException :** class này xử lý bất cứ lỗi nào xuất hiện trong khi làm việc với Database.

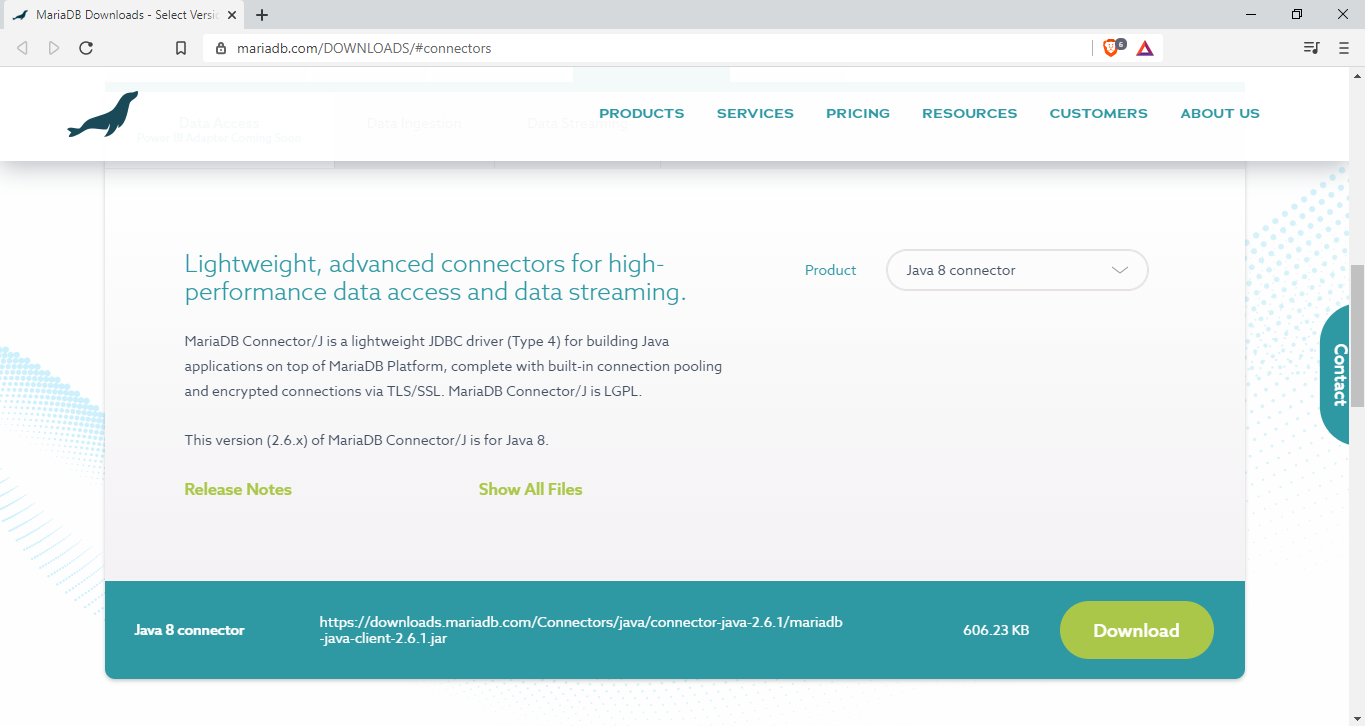
3.3 Kết nối Database với Java JDBC

Để làm việc với JDBC, bạn cần cài đặt Java và một cơ sở dữ liệu (database). Bạn có thể sử dụng bất kỳ database nào, chẳng hạn MySQL.

Như đã giới thiệu ở trên, để làm việc với cơ sở dữ liệu từ Java cần phải có Driver. Trong JDBC API, chúng ta có **java.sql.Driver,** đây là một interface có sẵn trong JDK. Mỗi hệ quản trị cần cài đặt một Driver riêng tương ứng với cơ sở dữ liệu đó. Như vậy công việc cần phải làm là download thư viện Driver ứng với loại Database mong muốn sử dụng.

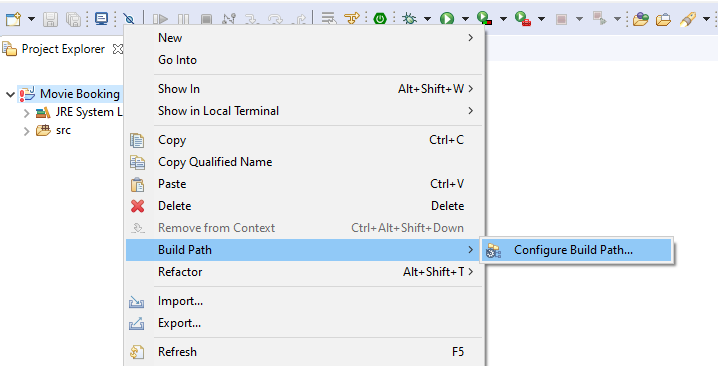
**3.3.1 Download Driver**

Vào trang web <https://mariadb.com/DOWNLOADS/#connectors> chọn phiên bản dành cho Java8 và tải về

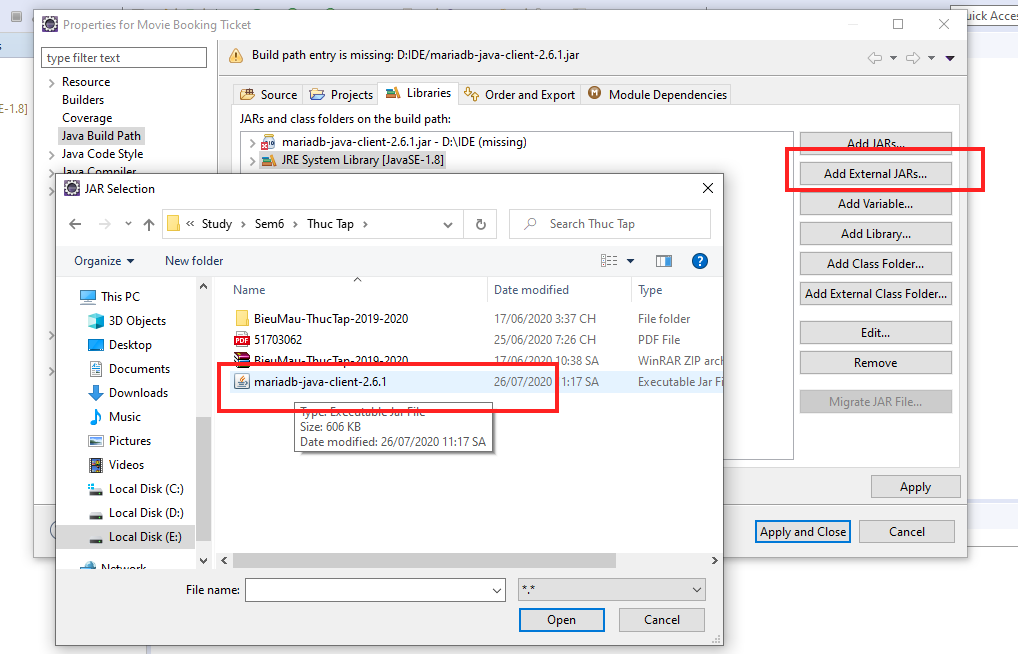
****

Hình 9: Download driver

Mở project bằng Eclipse, click chuột phải vào project, chọn Configure Build Path như hình bên dưới:

****

Hình 10: Configure Build Path

****

Hình 11: Thêm driver vừa tải về

Chọn sang tab Libraries, Chọn Add External JARs sau đó tìm driver vừa tải về, Open và Apply and Close

**3.3.2 Kết nối database từ Java**

Các bước kết nối Java với Database

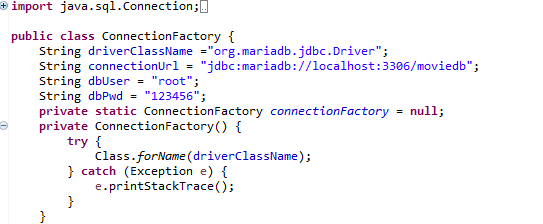
* Load Driver
* Tạo kết nối (Open Connection)
* Tạo câu lệnh truy vấn SQL (Statement)
* Thực thi câu lệnh truy vấn SQL (Execute Query)
* Đóng kết nối (Close Connection)

a) Load Driver

Để kết nối với database, chúng ta cần load driver và register nó với ứng dụng.

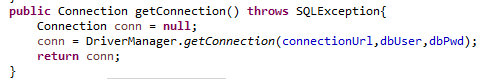
Có 2 cách để thực hiện:

* Class.forName() : load class Driver trong memory tại thời điểm runtime. Để đăng ký gọi phương thức: **Class.forName(“driverName”);**
* DriverManager.registerDriver() : DriverManager là class có sẵn trong Java. Để đăng ký gọi phương thức: **DriverManager.registerDriver(new DriverName());**



Hình 12: Cấu hình kết nối tới database

b) Tạo kết nối (Open Connection)



Hình 13: tạo kết nối

Sau khi load Driver, ta tạo kết nối như hình trên, với connectionUrl, dbUser và dbPwd đã được khởi tạo như hình trước đó.

c) Tạo câu lệnh truy vấn(Statement)

Sau khi một kết nối được thiết lập, ta có thể tương tác với CSDL

Các interface JDBCStatement, CallableStatement và PreparedStatement xác định các phương thức cho phép gửi các lệnh SQL và nhận dữ liệu từ cơ sở dữ liệu.



Hình 14: Tạo câu lệnh truy vấn

d) Thực thi câu lệnh truy vấn SQL(Execute Query)

Statement cung cấp một số phương thức để truy vấn SQL như sau:

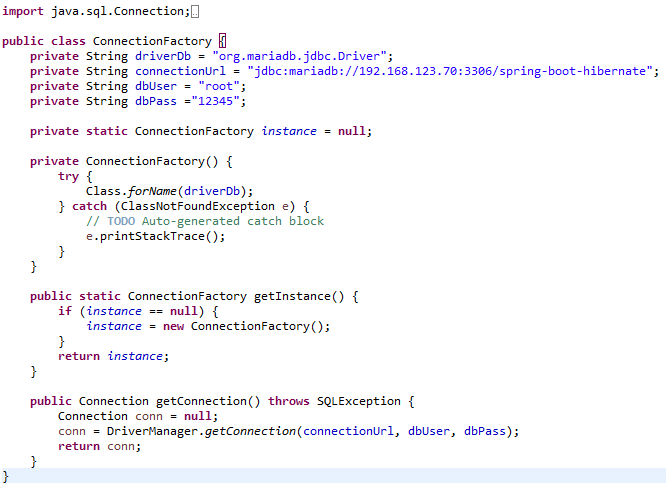
* Phương thức ***executeQuery()*** : được sử dụng để thực hiện các truy vấn truy xuất giá trị từ cơ sở dữ liệu (select). Phương thức này trả về đối tượng ResultSet có thể được sử dụng để lấy tất cả các dữ liệu (record) của bảng.
* Phương thức ***executeUpdate()*** : được sử dụng để thực hiện các truy vấn insert/ update/ delete.
* Phương thức ***execute()*** : có thể thực thi cả 2 trường hợp trên. Nếu phương thức statement.getUpdateCount() trả về số lượng record bị affect.
* Nếu giá trị **> 0**, có nghĩa là thực thi các câu lệnh insert/ update/ delete.
* Nếu giá trị **= 0**, có nghĩa là thực thi các câu lệnh insert/ update/ delete không có dòng nào bị ảnh hưởng hoặc thực thi câu lệnh cập nhật data structure.
* Nếu giá trị **= -1**, có nghĩa là thực thi câu lệnh select. Khi đó, có thể gọi tiếp lệnh statement.getResultSet() để lấy ResultSet.

e) Đóng kết nối (Close Connection)

Cần phải gọi phương thức close() để đóng kết nối sau khi sử dụng để giải phóng tài nguyên. Khi Statement được đóng thì ResultSet của Statement cũng được đóng theo.

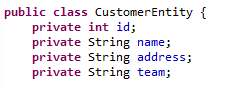
**3.4 Chương trình Java kết nối cơ sở dữ liệu sử dụng JDBC**

Trước tiên ta sẽ tạo class ConnectionFactory để thực hiện việc kết nối tới database



Hình 15: Cấu hình kết nối tới database

Sau đó ta tạo một Entity có tên là CustomerEntity bao gồm constructor và các phương thức get() set() như sau:

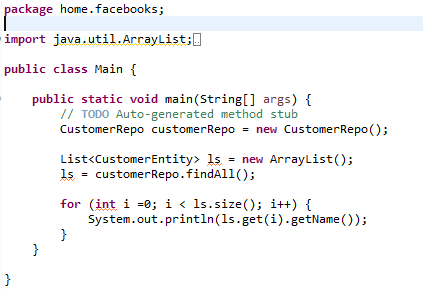


Hình 16: Lớp CustomerEntity (chỉ hiển thị thuộc tính)



Hình 17: Class CustomerRepo

Ở class này ta sẽ thực hiện các câu truy vấn như select, insert, update, delete (ví dụ ở trên thực hiện câu lệnh select)



Hình 18: Class Main

Cuối cùng, ta khởi tạo đối tượng customerRepo tại hàm main, sau đó thực thi phương thức findAll() đã định nghĩa ở hình để tìm ra danh sách các đối tượng CustomerEntity.

**CHƯƠNG 4 – NGHIÊN CỨU THỰC NGHIỆM**

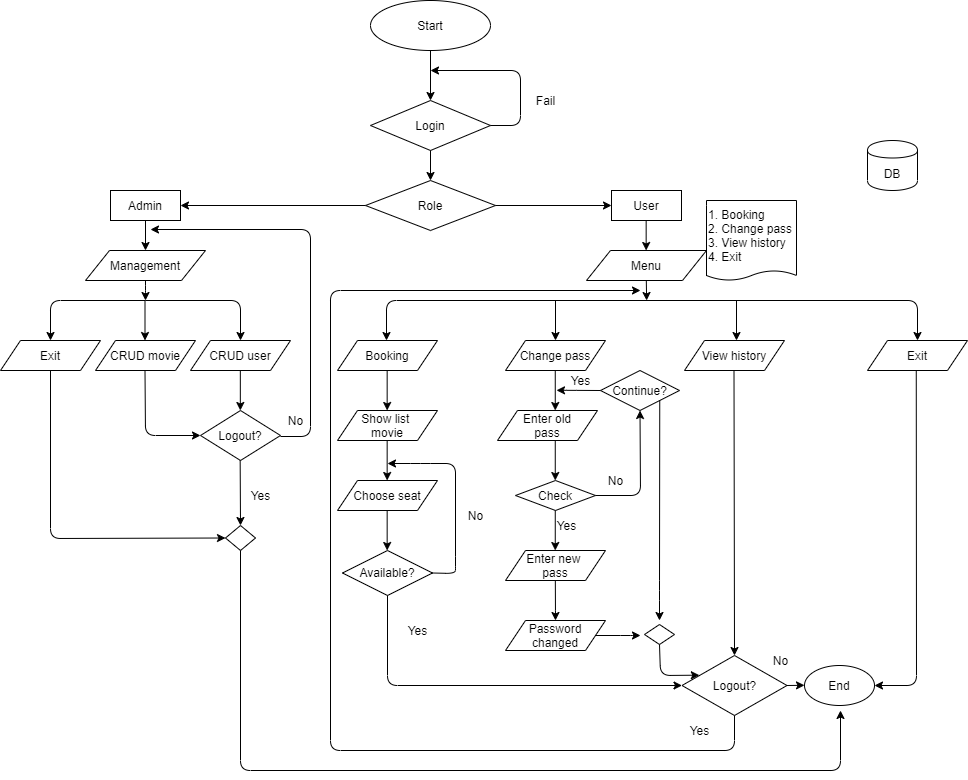
**4.1 Giới thiệu đề tài**

Để đề tài nghiên cứu được mang tính thực tế, em chọn một ứng dụng đặt vé xem phim để demo. Giúp hiểu rõ hơn những gì mình đã học được.

Đề tài sẽ sử dụng những gì đã nêu ở các chương trước. Do chỉ tập trung xử lí về các luồng sự kiện và tương tác với database nên ứng dụng sẽ không có giao diện và chạy bằng màn hình console của Eclipse.

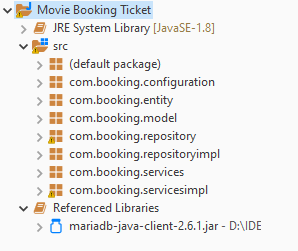
**4.2 Thực hành**

***4.2.1 Flowchart của ứng dụng***

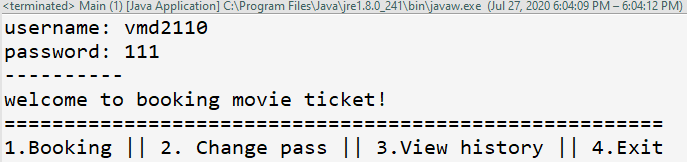


Hình 19: Flowchart

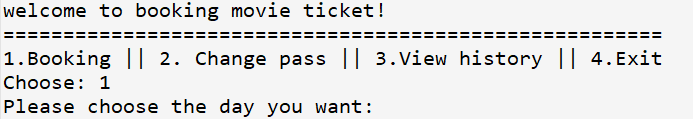
***4.2.2 Ứng dụng***

****

Hình 20: Các package của project

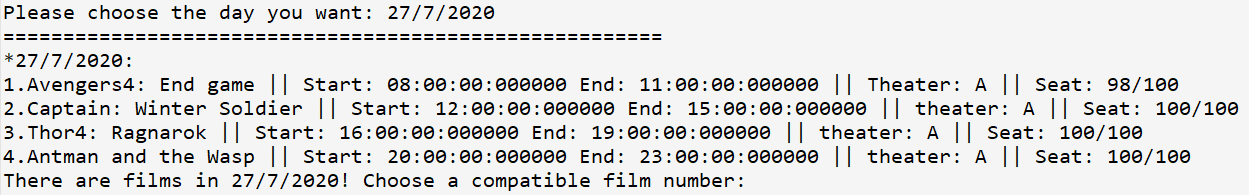


Hình 21: Màn hình đăng nhập



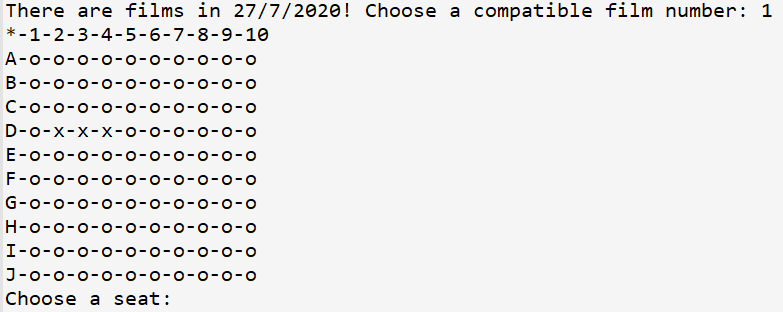
Hình 22: Màn hình chọn chức năng thực hiện

Với quyền hạn của user, ta có thể thực hiện 1 trong 4 chức năng được liệt kê ở màn hình trên. Ta sẽ thực hiện từng chức năng. Với chức năng đầu tiên, khi muốn đặt vé hệ thống sẽ cho nhập ngày mà ta muốn coi phim vào ngày đó.

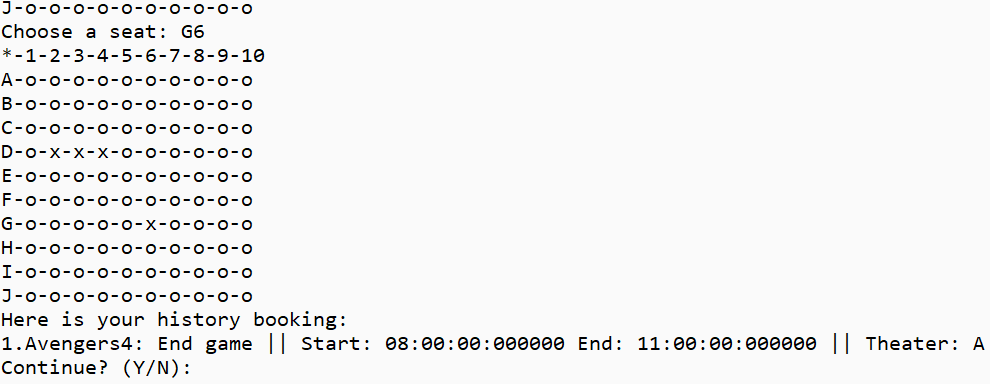


Hình 23: Màn hình hiển thị danh sách chiếu phim vào ngày 27/7/2020

Hệ thống sẽ đưa ra danh sách các phim được chiếu vào ngày hôm đó. Người dung chọn phim bằng cách nhập số thứ tự ở đầu mỗi phim.

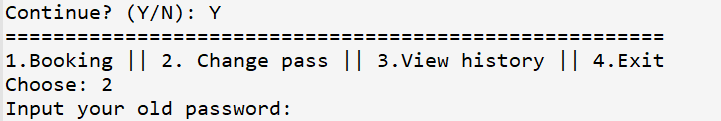


Hình 24: Màn hình hiển thị chỗ ngồi

Đây là danh sách chỗ ngồi của phim vào khung giờ tương ứng. Với vị trí x đã có chỗ ngồi.

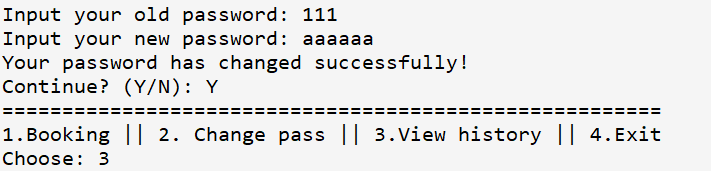
Hình 25: Màn hình thông báo chọn thành công

Sau khi đã chọn chỗ thành công hệ thống sẽ cập nhật lại ví trí chỗ ngồi. Người dung đã chọn chỗ ngồi là G6 vậy nên vị trí tương ứng đã đổi thành dấu x. Hệ thống sẽ đưa ra câu hỏi cho người dung có muốn tiếp tục thực hiện chức năng nào đó không, nếu có thì nhập vào Y.



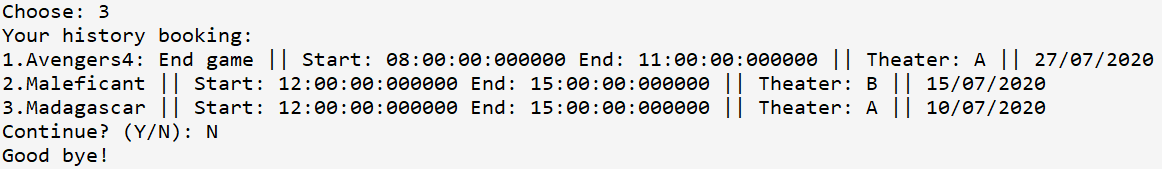
Hình 26: Màn hình chọn chức năng

Kế đến ta sẽ thực hiện chức năng đổi mật khẩu cho người dùng



Hình 27: Màn hình thông báo thay đổi mật khẩu thành công

Sau khi thay đổi mật khẩu thành công hệ thống sẽ tiếp tục hỏi rằng người dùng có muốn thực hiện tiếp hay không. Ta nhập Y và tiếp tục thực hiện chức năng thứ ba.



Hình 28: Màn hình hiển thị lịch sử đặt vé

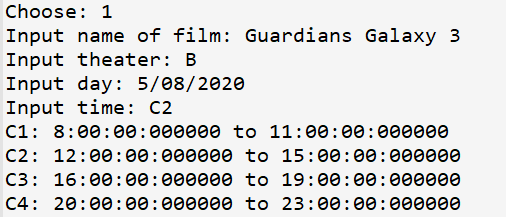
Sau khi yêu cầu hệ thống đã hiển thị ra lịch sử mua vé của người dùng. Sau đó người ùng đã nhập N. Hệ thống đã đóng lại.

Tiếp theo ta sẽ thực hiện các chức năng hệ thống cung cấp cho admin. Đầu tiên ta sẽ login vào hệ thống với tài khoản được phân quyền là admin



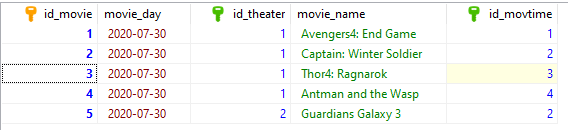
Hình 29: Màn hình đăng nhập admin

Đây là các chức năng mà admin có thể thực hiện được. Ta sẽ bắt đầu với chức năng thứ nhất.



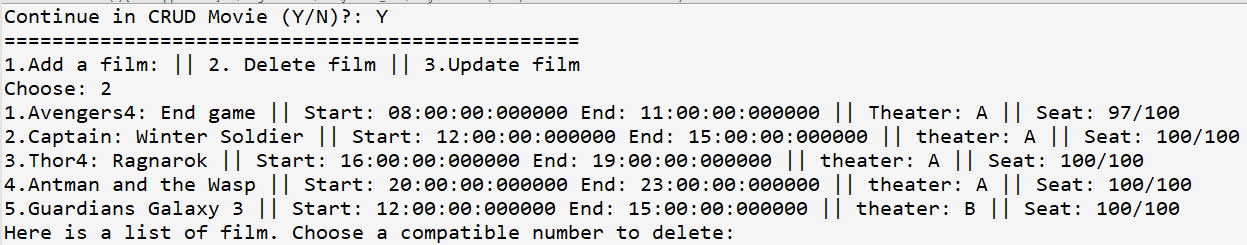
Hình 30: Màn hình thêm

Thực hiện chức năng them một bộ phim đối với chức năng thứ nhất, ta sẽ nhập các thông tin tương ứng vào. Đối với thông tin nhập khung giờ chiếu, sẽ có sẵn các khung giờ cho admin chọn.



Hình 31: Bản ghi của bảng Movie

Sau khi thêm một bộ phim thành công thì tại CSDL đã hiển thị lên được dữ liệu vừa them vào. Sau đó hệ thống sẽ hỏi admin có muốn tiếp tục không (như hình dưới) và ta sẽ nhập Y để tiếp tục.



**TÀI LIỆU THAM KHẢO**

**Tiếng Việt**

<https://vi.wikipedia.org/wiki/Java_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)>

<https://blog.tinohost.com/mariadb-la-gi-gioi-thieu-va-cai-dat-mariadb/>

<https://stackjava.com/install/mariadb-la-gi-cai-dat-mariadb-tren-windows-10.html>

<https://gpcoder.com/5188-huong-dan-ket-noi-co-so-du-lieu-voi-java-jdbc/>

**Tiếng Anh**

<https://beginnersbook.com/2013/05/java-introduction/>