Medical School LIC Scheduler

Developer Manual

Version 1.4

Revision History

Date	Version	Description	Author
05/Mar/2019	1.0	Initial Draft	Katie Ortstadt
24/Mar/2019	1.1	Continued Work	Katie Ortstadt
24/Apr/2019	1.2	Edits	Justin Herold
24/Apr/2019	1.3	Deployment Info	Alexander Parris
27/Apr/2019	1.4	Final review/added small info	Zach Alaniz

Table of Contents

Developer's Manual

1. Introduction

1.1 Purpose

The purpose of the *Developer's Manual* is to provide a comprehensive guide for developing our project. It describes how to install the software, how to access the database, and how to edit the code. Anyone wishing to perform maintenance or add new features to the project should reference this manual for help.

1.2 Project Overview

The new TCU and UNTHSC School of Medicine is taking a progressive approach to curriculum for their students. The standard for medical clerkships is for a medical student to focus on one practice, then move on to the next practice. This leaves a gap of time between learning and implementing a medical practice in the real world. The Longitudinal Integrated Clerkship (LIC) will engage students in a variety of medical practices in 2-week cycles, so students will constantly be maintaining their grasp on import skills and practices. It is our job to provide the scheduling application that will best match each student and doctor, at the best times. We solve this problem with two solutions. The first allows students to built their own schedules based on location and time preferences. This involves an interactive user interface sent to each student. The second, referred to as "brute force", automatically generates schedules with no student input. The "brute force" algorithm serves as a backup, in the event that the first solution fails. The end goal is to provide student and doctor schedules in an appropriate format so that the LIC administrator can easily view and distribute them.

2. Development Environment Setup

2.1 Java

The application is developed in the Spring Boot framework, which uses Java. You will need both JRE (Java Runtime) and JDK (Java Development Kit). Since JDK comes with JRE, we will focus on the installation of JDK.

2.1.1 Version Check

To see if JDK is already installed on your machine, enter the following command into the terminal:

```
$ javac -v
```

This will print the current version of JDK you have, if any. You will need JDK v1.8 or higher.

2.1.2 Downloading Java

If you do not have Java on your machine, you will need to install it. First, navigate to the official Oracle download page: http://www.oracle.com/technetwork/java/javase/downloads/index.html. Select the installation button for "Oracle JDK". On the next page, select the JDK download. For Linux, this should be a .tar file.

2.1.3 Installing Java

Navigate to the directory location where you would like to install the JDK, and move the .tar file to this directory. Next, unpack the java tarball (use the correct file name for your installation):

```
$ sudo tar xzvf jdk-8uversion-linux-i586.tar.gz
```

At this point you can delete the .tar file if you would like. The installation is complete.

2.2 Apache Maven

You will need Apache Maven installed to build the project.

2.1.1 Version Check

To see if Maven is already installed on your machine, enter the following command into the terminal:

```
$ mvn -v
```

This will print the current version of Apache Maven you have, if any. You will need Maven 3.2.5 or higher.

2.1.2 Downloading Maven

If you do not have Maven on your machine, you will need to install it. First, navigate to the official Apache download page: https://maven.apache.org/download.cgi. Select the binary file for your operating system. For Linux, this should be a .tar file.

2.1.3 Installing Maven

Navigate to the directory location where you would like to install Maven, and move the .tar file to this directory. Next, unpack the maven tarball (use the correct file name for your installation):

```
$ sudo tar xzvf apache-maven-3.6.0-bin.tar.gz
```

Next you will need to set the path variable so the system can find your maven installation. Make sure you are using the correct file name for your installation:

```
$ export PATH=/opt/apache-maven-3.6.0/bin:$PATH
```

Now Maven should be installed on your machine. You can delete the .tar file if you would like.

2.2 IDE (Integrated Development Environment)

For this project, any IDE can be used as long as it supports Spring development. We have chosen to use IntelliJ; however, the full version must be purchased, as the Community Version does not include support for Spring. Free licenses are available for students and open source developers.

2.1.1 IntelliJ

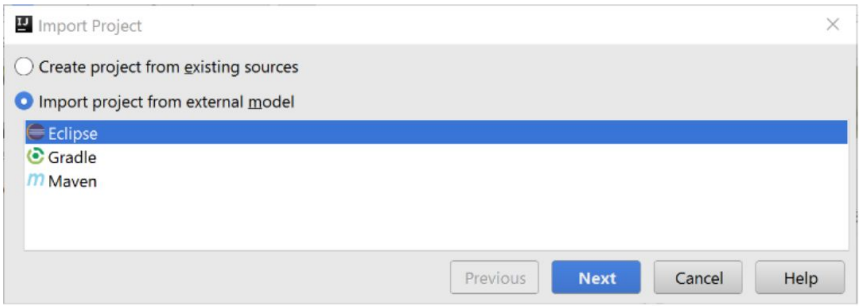
To download IntelliJ, navigate the official download page: https://www.jetbrains.com/idea/download. Select the Ultimate Edition for your operating system. Unpack the downloaded file and run IntelliJ. Follow the on-screen prompts to set up the IDE.

For the remainder of this manual, we will assume you are using IntelliJ. If you are using a different IDE, the steps to import and run the project may vary slightly. Be sure to consult your IDE's official documentation for assistance.

3. Project Setup

3.1 Loading the Project

To load the LIC code, open up IntelliJ and select “Import Project” on the welcome screen. Navigate to the project and select the top-level folder. The Import Project screen will appear. Chose the “Import Model From External Model” radio button. Select “Maven” from the list and click “Next”.



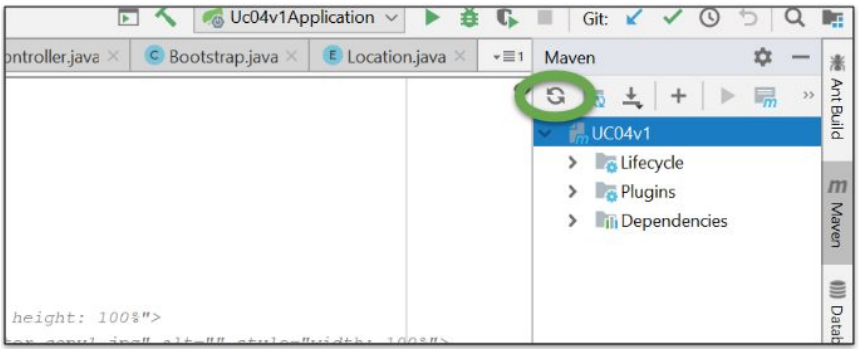
On the following screen, select “Import Maven projects automatically” and leave the rest of the values at their default. Select “Next”. Now the maven project should be selected for import. Press “Next” again. Finally, select the correct java version, and finish importing the project.

3.2 Importing Dependencies

As with most projects, the LIC scheduling software relies on external dependencies. This is handled by Maven, but you will need to import the dependencies initially. Any time dependencies are added or changed, you will need to re-import them.

Dependencies are stored in the maven pom file, called “pom.xml” in our project. It can be found in the primary project folder.

To update the maven pom file, navigate to the tabs on the right-hand side of IntelliJ. Select the tab labeled “Maven.” When the tab opens up, select the refresh icon on the top, as indicated below:



After a few moments, the project dependencies will be imported. At this point, the project should be fully loaded and ready to run.

4. Running the Program

4.1 Compiling and Running on Localhost

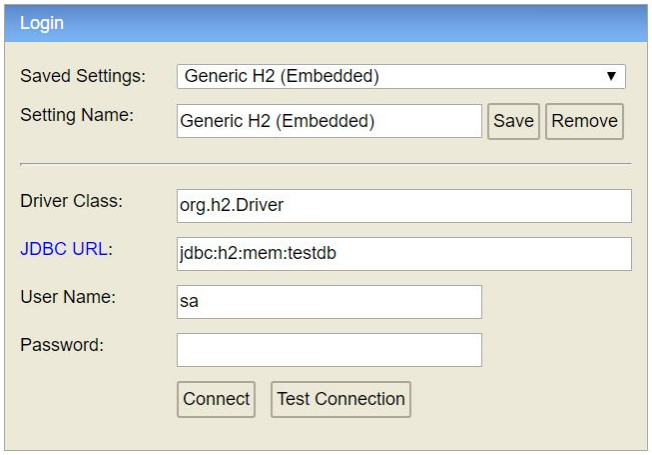
When you are ready to run the program, enter the following line into the IntelliJ command terminal:

```
$ mvn spring-boot:run
```

After a few moments, the spring project will build. In the IntelliJ terminal window, you will see the localhost address which the project is hosted on. Typically, this will be localhost:8080. Open up a web browser (other than Internet Explorer, which is not supported) and navigate to this address.

4.2 Viewing the Database (Local H2 Database)

To view the database, navigate to http://localhost:8080/h2-console/. You will need log on to view the database. Make sure the User Name field reads “sa”, as shown below:



After you have filled out the form, press “Connect.” You can now view the database. It should update in real time as you use the program.

5. Deployment and AWS Configuration

5.1 Creating AWS Server

Before setting up the server create an Amazon AWS account. You will be sent to the Amazon AWS dashboard where you need to select “EC2” > “Instances”. Select “Launch Instance”, select “Red Hat Linux” as the operating system, and select a t2.medium server. We use a t2.medium server for increased speed and reliability, a lesser server will crash because SQL won’t have enough memory to run on. Click “review and Launch” and add the following security groups (Take note of which groups are inbound versus outbound).

Description | Inbound | **Outbound** | Tags

Edit

Type	Protocol	Port Range	Destination	Description
All traffic	All	All	0.0.0.0/0	
MYSQL/Aurora	TCP	3306	0.0.0.0/0	
MYSQL/Aurora	TCP	3306	::/0	

Description | **Inbound** | Outbound | Tags

Edit

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	
HTTP	TCP	80	::/0	
Custom TCP Rule	TCP	8080	0.0.0.0/0	
Custom TCP Rule	TCP	8080	::/0	
SSH	TCP	22	0.0.0.0/0	
MYSQL/Aurora	TCP	3306	0.0.0.0/0	
MYSQL/Aurora	TCP	3306	::/0	

When you click “Launch” you will be able to create a new key pair. Select “Create new key pair” and give it a key pair name, i.e. “LIC_KEY”. Download the new key pair “LIC_KEY.pem” and store it in a secure location on your device. Then click launch instances. It will take some time for Amazon to create this instance anywhere from 2 - 20 minutes.

If these options have changed in the future here are the exact configuration settings that we used.

AMI Details

RHEL-7.6_HVM_GA-20181017-x86_64-0-Hourly2-GP2 - ami-0b500ef59d8335eee
Provided by Red Hat, Inc.
Root Device Type: ebs Virtualization type: hvm

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.medium	Variable	2	4	EBS only	-	Low to Moderate

Security Groups

Security Group ID	Name	Description
sg-095d447a195d69deb	launch-wizard-5	launch-wizard-5 created 2019-03-18T08:09:17.843-05:00

All selected security groups inbound rules

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	
HTTP	TCP	80	::/0	
Custom TCP Rule	TCP	8080	0.0.0.0/0	
Custom TCP Rule	TCP	8080	::/0	
SSH	TCP	22	0.0.0.0/0	
MYSQL/Aurora	TCP	3306	0.0.0.0/0	
MYSQL/Aurora	TCP	3306	::/0	

Instance Details

Number of instances 1
Network vpc-8b0712e3
Subnet subnet-088ee572
EBS-optimized No
Monitoring No
Termination protection No
Shutdown behavior Stop
Stop - Hibernate behavior Disabled
Capacity Reservation
IAM role None
Tenancy default
T2/T3 Unlimited Disabled
Host ID
Affinity
Kernel ID Use default
RAM disk ID Use default
User data
Assign Public IP Yes
Assign IPv6 IP No

Purchasing option On demand

While you are looking at your new EC2 instance, rename it “LIC Application”.

Next you will need to set up a static IP address, incase the server is shut down and restarted the domain will not change. To do this go to the EC2 console under the “Network & Security” tab click “Elastic IPs”. Select “Allocate New Address”, keep “From Amazon Pool” checked, and click “Allocate”. Once your elastic IP address is allocated you will need to associate it with the EC2 server that we just created. To do this select the address in the control panel and click “Actions” > “Associate Address”. Then select the EC2 server in the “Instance dropdown, and select the available IP from the “Private IP” dropdown. Finally click “Associate”, and you are ready to start setting up the processes that need to run on the server.

5.2 Setting up software on the AWS Server

Now it's time to connect to the server. Go to the EC2 instances console and select the server. Click "Actions" > "Connect". Copy and paste the example ssh command into the terminal application on your computer and hit enter. You may need to change the file path of the .pem file, depending on where you stored it on your computer.

```
Alexanders-MacBook-Pro-3:~ abparris$ ssh -i "/Users/abparris/Downloads/dev_ops_c
ourse.pem" ec2-user@ec2-3-19-86-213.us-east-2.compute.amazonaws.com
```

Once in the server run the following lines of code:

- sudo su
- yum update
- yum install java-1.8.0-openjdk
- yum install wget
- yum install docker-engine
- systemctl enable docker-engine
- systemctl enable docker.service
- systemctl start docker
- docker run -d --name prod_mysql mysql/mysql-server:latest
- docker run mysql -u root
- docker run mysql -u root MYSQL_ALLOW_EMPTY_PASSWORD
- docker run mysql -u root MYSQL_ALLOW_EMPTY_PASSWORD=true
- docker run mysql -u root MYSQL_ALLOW_EMPTY_PASSWORD=yes
- sudo docker run -d --name prod_mysql -p 3306:3306 -v /var/lib/mysql:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=tiger mysql/mysql-server:latest
- sudo docker exec -it prod_mysql bash
- mysql -p
- tiger

  The following are the sql commands needed to run.

- Create database springguru;
- Grant all on * to 'spring_guru_owner' @ '*';
- Use springguru;
- Grant all on * to 'spring_guru_owner' @ '*';
- Grant all privileges on springguru.* to 'spring_guru_owner' @ '%';
- GRANT ALL PRIVILEGES ON lic_dev.* TO'spring_guru_owner'@'*';
- GRANT ALL PRIVILEGES ON lic_dev.* TO'spring_guru_owner'@'%';
- quit
- exit
- cd /etc/systemd/system
- vi springboot.service

At this point hit the 'i' key and copy the following into the file (not including quotes).
"

[Unit]
Description=Spring Boot Service
After=syslog.target

[Service]
User=ec2-user
# set dir to location of application.properties and springboot jar
WorkingDirectory=/home/ec2-user
ExecStart=/bin/java -jar UC04v1-4.4.1.jar
SuccessExitStatus=143

[Install]
WantedBy=multi-user.target
“

Now hit the ‘esc” key followed by “:”, “wq”, and the “enter” key.

- run systemctl daemon-reload
- systemctl daemon-reload
- systemctl enable springboot.service
- cd /home/ec2-user

Now you are going to use the wget command to retrieve the .jar of the project from a foreign machine. I will include my command, but yours will look different since you are using a different machine.

- wget http://18.220.6.104:8081/artifactory/libs-release-local/LIC/UC04v1/4.2.1/UC04v1-4.2.1.jar
- systemctl start springboot

5.3 Code Changes Needed (if using new server)

If you are setting up this application on a server other than the one we have deployed on as of May 2, 2019, then you will need to alter the IP addresses in the source code to whatever the static IP you chose in AWS. I will now include photos of said source code changes.

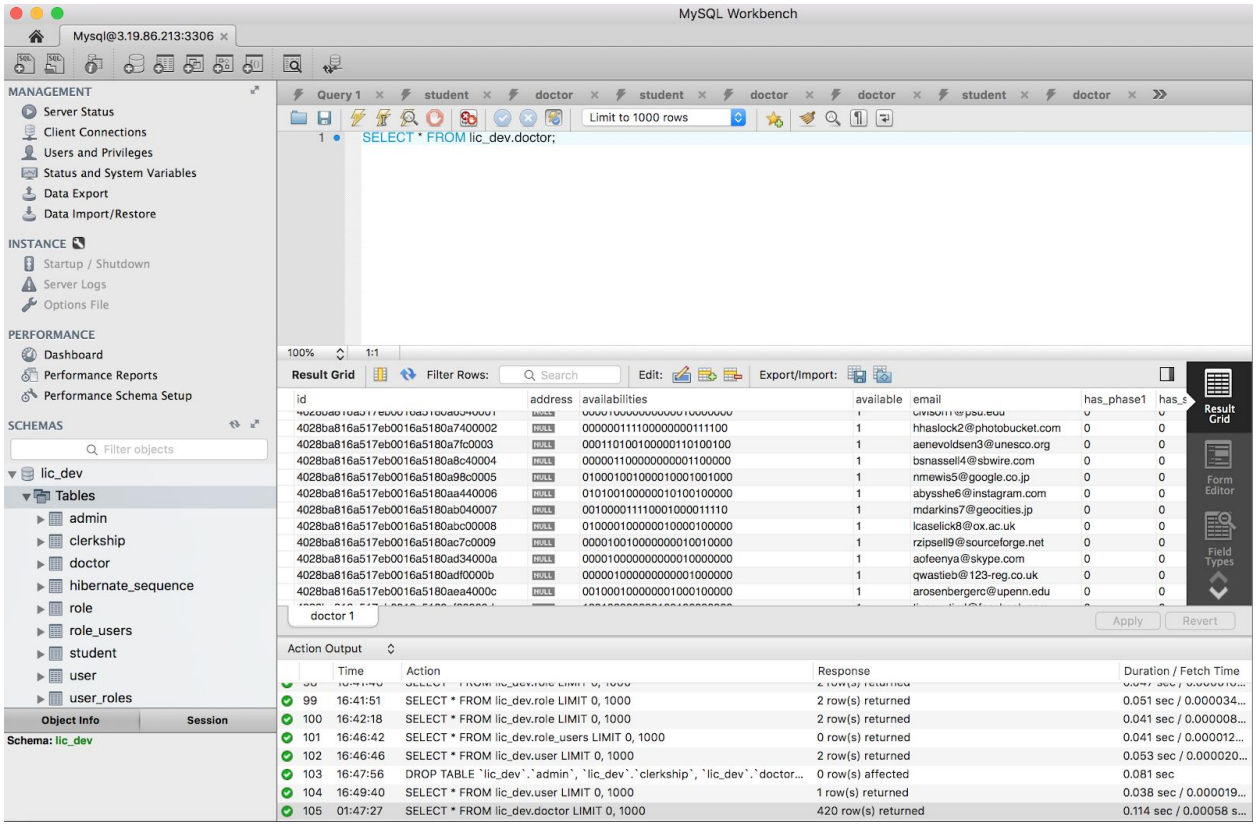
```
spring.datasource.url=jdbc:mysql://ec2-3-19-86-213.us-east-2.compute.amazonaws.com:3306/lic_dev?&serverTimezone=UTC
```

This line in Application.properties

5.4 Viewing the Database (Deployed SQL Database)

To view the database we are using MySQLWorkbench, which can be downloaded for free at: https://dev.mysql.com/downloads/workbench/

Start MySQLWorkbench and go to “Database” > “Connect to Database...” Type in your static ip for “Hostname”, the username is “spring_guru_owner”, and the password is “GuruPassword”. From here you should be able to view tables within the database.



6. Further Resources

Here are some further resources to aid in development:

Spring Udemy Tutorial: https://www.udemy.com/spring-framework-5-beginner-to-guru/

Spring Reference Documentation: https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/

IntelliJ Website: https://www.jetbrains.com/idea/download/#section=windows