Adam Kedra, Aalap Parekh, Andrew Resaul, Nelly Laevsky

Professor Kehoe

IT490

**Group: Rotten Spaghetti**

***General VM Instructions:***

● After installing Ubuntu, open terminal

● Sudo apt-get install git

○ Gitk

○ Php

○ Vim

○ aptitude

● Mkdir git

● Cd git

● Git clone <https://github.com/keddrxd/IT490Spaghetti.git>

● Sudo aptitude

○ Once loaded press "/"

○ Type "php amqp" and hit enter

○ Press "n" until "php amqp" is highlighted (it might be highlighted already)

○ Press "+"

○ Press "g" and then "g" again

○ When prompted, press "q" and then hit enter to quit aptitude

● Sudo apt-get install rabbitmq-server

● Make startup servers

○ Follow this guide to create a small server using PHP- <https://medium.com/@benmorel/creating-a-linux-service-with-systemd-611b5c8b91d6>

***Backend VM Machine Instructions:***

* Sudo apt-get install sendmail
* Sudo rabbitmq-plugins enable rabbitmq\_management
* Open firefox, go to localhost:15672

o Login with username and password guest

o Create a new vhost “testHost”

o Add new user “admin” and make password

o Give permissions for root “/” and “testHost”

o Create another user “test” with pass “test”

o Create a new exchange testExchange” for “testHost” settings

o Create a new queueu “testQueue” for “testHost” routing key

o Click on testQueue, bind with exchange “testExchange”

* Sudo apt-get install mysql-server
* Sudo apt-get install python3
* Sudo apt-get install python3-pip
* Sudo pip install pika
* Sudo pip install mysql-connector
* Install a firewall with UFW between back end and front end

***Database VM Instructions:***

● Sudo apt-get install mysql-server

● Create root password

● Login with the command “sudo mysql –u root p”

● Type in commands:

○ CREATE USER ‘admin’@’localhost’ IDENTIFIED BY ‘pass’;

○ GRANT ALL PRIVILEGES on \*.\* TO ‘admin’@’localhost’ WITH GRANT OPTION;

○ FLUSH PRIVILEGES;

● Sudo apt-get install php-mysql

● Follow this guide to set up database replication- <https://www.cloudjojo.com/how-to-setup-mysql-replication-on-ubuntu-16-04-master-slave/>

***FrontEnd VM Instructions:***

● Install Apache

○ Sudo apt update

○ Sudo apt install apache2

○ Sudo ufw app list list

● Cd /var/www/html

***DMZ VM Instructions:***

● Sudo apt-get install python3

● Sudo apt-get install python3-pip

● Sudo pip install pika

● Sudo pip install mysql-connector

***Deployment VM Instructions:***

* Follow General VM instructions
* Create deployServer.service
* Create database versiondb;
* Create versionControl table
* Follow instructions on how to setup RabbitMQ for the deployment server.

***Setting up the database instructions:***

sudo mysql -u root

GRANT ALL PRIVILEGES ON \*.\* TO 'admin'@'localhost' IDENTIFIED BY 'adminPwd';

mysql -u admin -p

CREATE DATABASE usersDB;

USE usersDB;

CREATE TABLE users (firstName varchar(255), lastName varchar(255), username varchar(100), password varchar(100), email varchar(255), zip varchar(20));

CREATE TABLE session (username varchar(100), sessionKey varchar(255));

CREATE TABLE category (username varchar(100), comedy varchar(100), horror varchar(100), action varchar(100), scifi varchar(100), romance varchar(100), animation varchar(100));

CREATE TABLE movieRec (username varchar(255), comedy varchar(255), rd1 varchar(255), horror varchar(255), rd2 varchar(255), action varchar(255), rd3 varchar(255), scifi varchar(255), rd4 varchar(255), romance varchar(255), rd5 varchar(255), animation varchar(255), rd6 varchar(255));

CREATE TABLE action (action varchar(255), rd varchar(255));

CREATE TABLE animation (animation varchar(255), rd varchar(255));

CREATE TABLE comedy (comedy varchar(255), rd varchar(255));

CREATE TABLE horror (horror varchar(255), rd varchar(255));

CREATE TABLE romance (action romance(255), rd varchar(255));

CREATE TABLE scifi (scifi varchar(255), rd varchar(255));

CREATE TABLE friends (username varchar(255), friend1 varchar(255), friend2 varchar(255), friend3 varchar(255), friend4 varchar(255));

---------------------------------------------------------------------------------------------------------------------

CREATE DATABASE versionControl;

GRANT ALL PRIVILEGES ON \*.\* TO 'adam'@'localhost' IDENTIFIED BY 'deployPass';

USE versionControl;

CREATE TABLE version(dateVer varchar(255), verName varchar(255), status varchar(255));

**Change Log**

2/11/19

Virtual Machine

- Changed Network adapter in Virtual Box settings from NAT to Bridged Network to allow connection between other VMS.

2/11/19

RabbitMQ

- Edit the testRabbitMQ.ini changed the BROKER\_HOST = 192.168.1.4

- After changing that we are able to connect multiple VMs using the testRabbitMQ example

3/2/19

Database

- Created a new table in MySQL we were able to connect to it.

3/5/19

Authentication Fixes

- Updated authentication function and registration function

3/9/19

Website

- Addded sytling to index.php (Main Page)

- Created Redirect Page to SuccessPage for testing authentication

Database

- Added sessionKeys for user registration and user login

3/12/19

Website

- Added movie selection website where users are able to select movies they like

Database

- Created function to add user movie preferences to database

3/24/19

Website

- Fixed styling issues with Movie Selection Page

- Added google maps API function to Main Page to show movie theater locations

Database

- Test to add user preferences to the database (needs work)

4/1/19

MovieAPI

- Added python script to grab upcoming movies from the API, stored into JSON file

4/8/19

MovieAPI

- Moved data from JSON file through RabbitMQ to store data into the database for use in movie reccomender

4/15/19

Website

- Added movie recommendations and username to main page

4/22/19

Deployment

- Created sendToDeploy to package up FE, BE, DMZ server files to the deployment server using SCP and TAR

Error Logging

- Added error logging through the RabbitMQ server, still need to add the Date to the error message

4/23/19

Deployment

- Added script to mark package status as good or bad.

- Need work on installing packages on VM to testRabbitMQ

Database/Frontend

- Added release dates

- Error logging goes through the RabbitMQ/need to add other server errors to finish error logging

4/25/19

Database

- Database replication works on test machine, need to implement on production machines

5/2/19

System Administration

- Created services for each server to run on startup / .services scripts

- Created cronjob for api call, once a week at 11:30am to fill database

- Friends list testing works with inputting into database, just need a way to show to html page

5/3/19

System Administration

- Database replication works on Backend production machines(regular and failover)

- Symbolic link in /var/hmtl folder completed

5/8/19

Frontend

- Movie page and login page has bootstrap/need to test the functionality

5/9/19

Frontend

- All pages now have bootstrap just need to get the functionality working now

5/15/19

Backend

- Merged the RabbitMQ server and the database to one machine