

3RD YEAR PROJECT

TUTOR QUEUING

By Sean Carroll, 12517683 and Justinas Ulevicius, 12367951

0 TABLE OF CONTENTS

1 Introduction

1.1 Overview

1.2 Glossary

2 System Architecture

3 High-Level Design

3.1 Overall context diagram of the system

3.2 Registering a user

3.3 User log in

3.4 Acquiring student's IP address

4 Problems and Resolutions

4.1 Acquiring IP

4.2 IP length

4.3 JSON / Asynchronous problem

5 Installation Guide

5.1 Operating System

5.2 Web Browser

5.3 JavaScript

6 User Manual

6.1 Registering a user

6.2 Logging in

6.3 For tutors only

6.4 For students only

1. INTRODUCTION

1.1 Overview

Tutor Queuing is a platform allowing lab tutors and lecturers to effectively manage their lab sessions, while allowing students to request assistance with the click of a button on a website, QueueDCU.com.

In order to use the website, users are required to first register and log in. There are two types of users, lecturers / tutors and students. Depending on the type of the account, users are given different permissions and are redirected and given access to different sections of the website upon successful logging in.

Tutors and lecturers in DCU have email addresses ending in @dcu.ie and in order to register as a lecturer, you must use one of these email addresses. Once a lecturer logs in, they are given access to lab pages which allow them to begin a lab session in the specified lab location. Students on the other hand, are given lab pages which allow them to request assistance by entering the queue.

The aforementioned lab pages display the layout of each computer lab in the School of Computing. Once a student is stuck on a task during a lab session, they have the ability to join the queue and wait until their turn to get help from the tutor. In the meantime, a lecturer is able to see where on the lab layout the student requiring help is situated.

1.2 Motivation

The main motive for choosing to implement this project was to solve a problem many students in Computer Applications face multiple times per week, especially since most CA classes have been quite large in numbers the last several years. This often leads to quite chaotic lab sessions, with approximately one hundred students in a lab and only a small number of tutors and lecturers to assist these students with their queries.

This project will aim to solve these issues, as students will receive help in order that they request it. This will make lab sessions more fair towards the students and less chaotic for lecturers.

1.3 Wordpress blogs

Throughout the development of the project, Wordpress blogs were kept to record the progress of the development cycle.

Sean's blog: blogs.computing.dcu.ie/wordpress/tutorqueuing

Justin's blog: blogs.computing.dcu.ie/wordpress/qdcu

1.4 Glossary

HTML

HyperText Markup Language, is the standard markup language used to create web pages. It is written in the form of HTML elements consisting of tags enclosed in angle brackets, for example, <html>.

CSS

Cascading Style Sheets is a style sheet language used for defining the look and formatting of a webpage written in a markup language.

JavaScript

JavaScript is a dynamic computer programming language, widely used in websites, whose implementations allow client-side scripts to interact with the user, control the browser, and alter the document content that is displayed to the user.

PHP

PHP is a server-side scripting language designed for web development, but sometimes also used as a general-purpose programming language.

MySQL

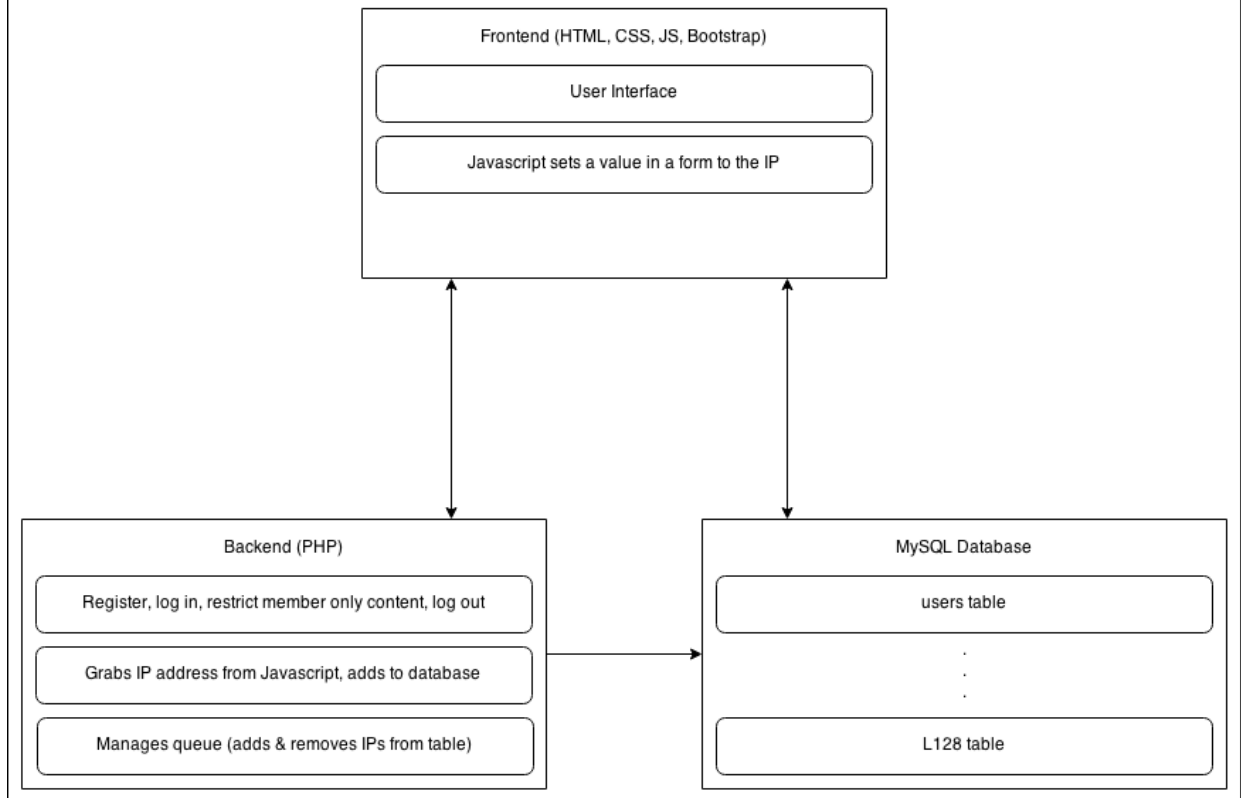
MySQL is a relational database management system (RDBMS) and most widely used open-source RDBMS.

2. SYSTEM ARCHITECTURE

There are three main aspects to the system architecture.

- The front end, which acts as a GUI for the user.
 - Written in HTML, CSS, and JavaScript using Twitter Bootstrap.
 - Contains buttons, which when clicked carry out the functionality by communicating with the back end.
 - Script written in JavaScript sets a value in a HTML form to the IP of the computer being used when a user tries to join the queue.
- The back end, which gives the web app its functionality.
 - It is written in PHP.
 - The register and log in system is written entirely in PHP and is connected to a MySQL database (described below). User information is sent to and retrieved from the database.
 - PHP script grabs the IP from the HTML form (which was set by the JavaScript earlier).
 - The IP is added to the appropriate lab table in the database.
 - The IP is removed from the table once the user's allocated time is up or the lecturer moves onto the next student in the queue.
- The MySQL database, which stores user information and IP addresses of the lab computers.
 - Database contains a table called *users* which is responsible for storing user information. Information such as username, email, password (hashed) and salt (used for even more secure encryption) are added when a user registers and is accessed when a user attempts to log in.
 - Tables for each lab (Lab128, Lab25, etc.) are created for each lab location. IP addresses of computers are added to the appropriate lab table when a student clicks the button to enter the queue.

OVERALL SYSTEM ARCHITECTURE OF QUEUEDCU.COM



3. HIGH-LEVEL DESIGN

The system is made up of several components, each of which carry out certain functionality. For easier understanding of the aforementioned components, context diagrams are provided.

3.1 Overall context diagram of the system

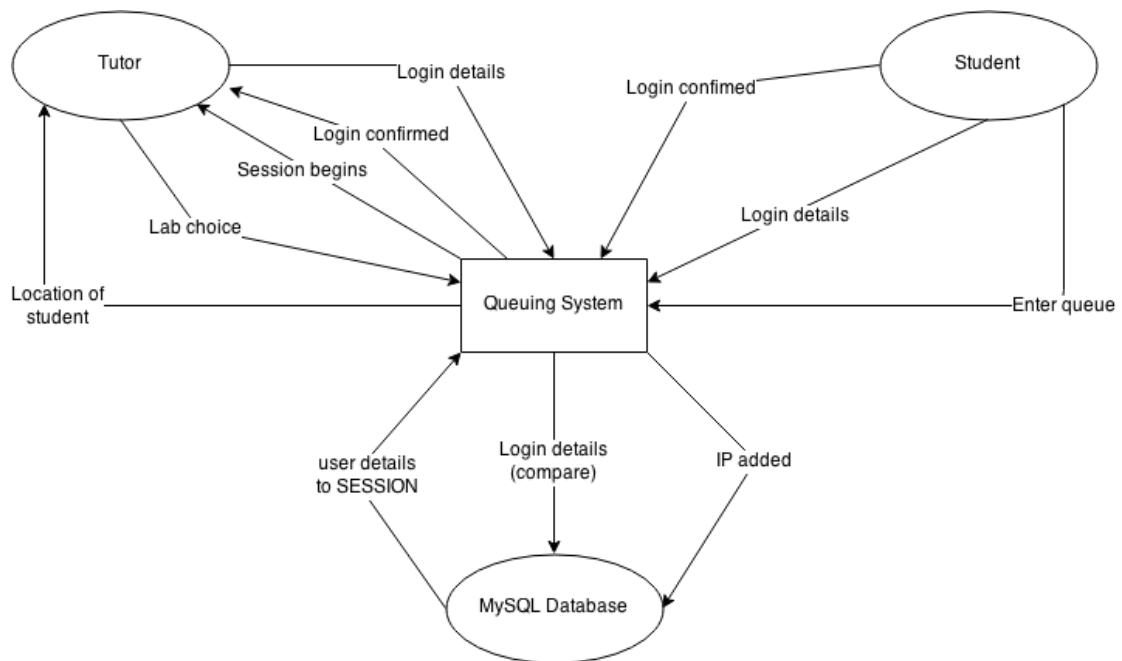


Fig. 3.1.1

From Fig. 3.1.1 we can see a brief overview of how the system functions as a whole. The two types of users, a tutor or a student must first log in, having previously registered, before they can carry out any of the tasks. After logging in, a tutor has the ability to begin a lab session in a specified location, and from that point onwards, a student can request assistance by entering the queue. By doing this, the local IP address of the student's computer is taken and added to the MySQL table dedicated for that lab and the location is sent to the tutor. Once the student has received help, their IP is removed from the table.

3.2 Registering a user

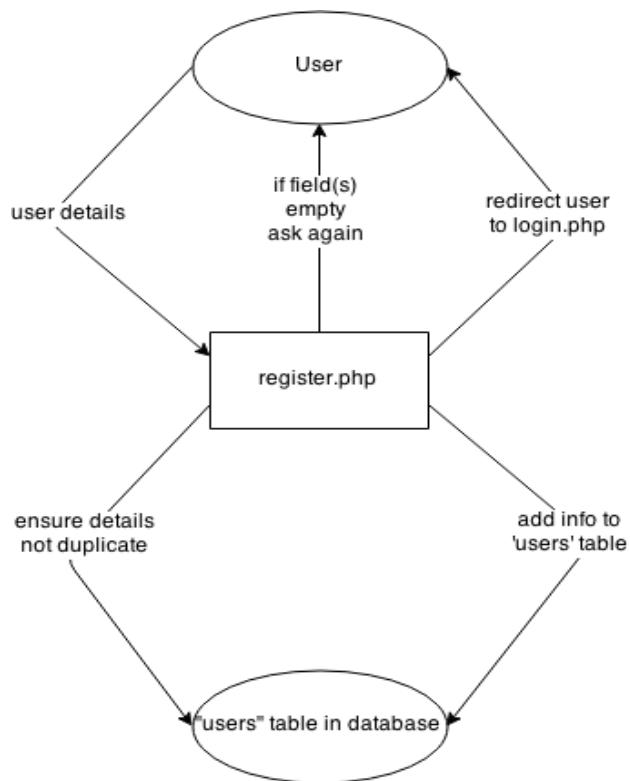


Fig. 3.1.2

The user registration script is written in PHP. The script is responsible for ensuring that the form is not empty and performs MySQL checks for each field (username, password, email) to ensure that these fields are not already present in the users table. It returns appropriate messages if any of the checks do not pass. If the checks do pass, the PHP script contains a MySQL query to create a new user row and adds all the new information into the table. The script uses salt and PHP's hash function along with the sha256 algorithm to hash the password. The hashed password and salt are placed alongside other user details in the table.

3.3 User log in

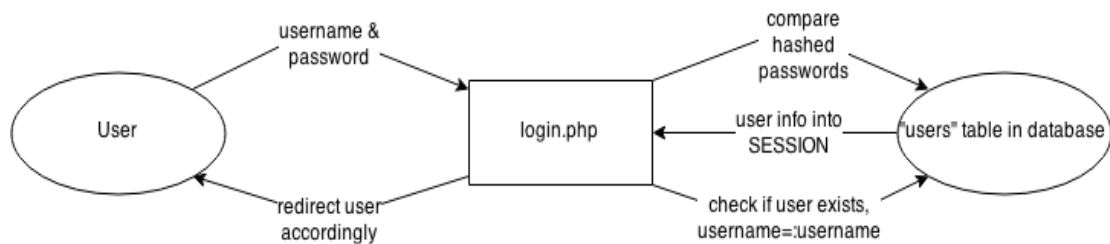


Fig. 3.1.3

Once a user enters their username and password, the user information is retrieved from the table where the entered username is equal to the username in the table. The entered password is then hashed and compared to the hashed password stored in the user's row. If the login is successful, the user's details are stored in the SESSION array and the user is redirected.

3.4 Acquiring student's IP address

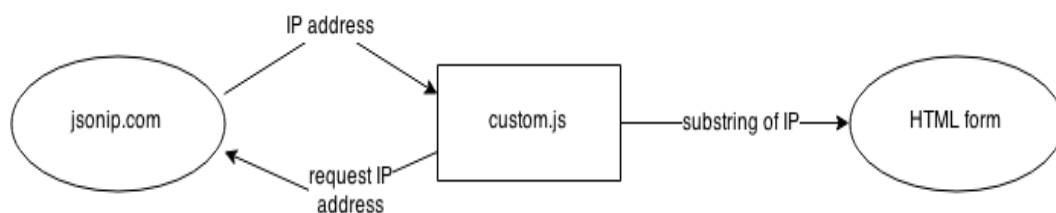


Fig. 3.1.4

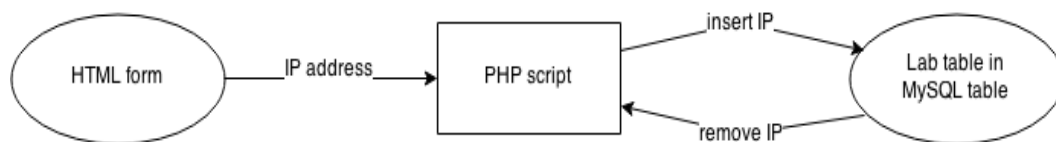


Fig. 3.1.5

The IP address of the computer at which the student is situated is collected using jsonip.com, however, this third party website returns some unnecessary information, therefore PHP's substring function is used to acquire only the IP address from the returned information. This is done using a JavaScript script and since PHP can't use information from a JavaScript variable, the IP address must first be placed into a HTML form, which is kept hidden and not shown to the user. The IP address is then inserted into the appropriate table in the database. Each lab has a dedicated table in the database. The table, which acts as the queue, is ordered by the time the IP address has been entered into it. Once a student has been assisted, the IP address is removed from the queue (a.k.a the lab's table).

4. PROBLEMS AND RESOLUTIONS

4.1 Acquiring IP

As mentioned above in the High-Level Design section, problems were encountered when attempting to grab a user's IP address. Initially, some third party websites would not return the local IP address of the computer, possibly cause by security restrictions. Once a suitable website was discovered, another problem was encountered. Since PHP is a server-side language, it did not communicate well with JavaScript in the sense that there was no way to get the IP address from a JavaScript variable. This issue was resolved by getting JavaScript to place the IP address into a HTML form, from which PHP could acquire the IP address from.

4.2 IP Length

The third party jsonip.com website which gets the local IP address of the student's computer would return the substring of the 13 characters of an IP address. If the last octet of the address happens to be a double digit, no problems are encountered, however, if it is a single digit, the substring grabs the next character after it, which is a comma (,). When this address and comma are added to the database table and compared against the IP address of the image ID in the HTML lab layout, they are not equal, when in reality they represent the same IP address. To solve this issue, a comma was appended to the end of the HTML ID.

4.3 JSON / Asynchronous problem

Json is asynchronous and the variables underneath it would execute before JSON had finished acquiring the IP address. To solve this, an AJAX function was used to make it synchronised so that the IP would be grabbed with JSON before passing the variable to the HTML form from which the PHP grabs the IP.

5. INSTALLATION GUIDE

Seeing as QueueDCU is a web application, its installation is very minimalistic; however, the system still holds some technical requirements in order to function as designed.

5.1 Operating System

The operating system on the machine which the web application is run on is not important so long as it is recent enough to allow the installation of an up to date web browser.

An internet connection is essential to carry out any kind of functionality on QueueDCU.com.

5.2 Web Browser

QueueDCU.com is fully functional with any modern browser. More specifically,

All versions of Google Chrome.

Internet Explorer 9 and above.

Mac OS X Safari 4.1.3.

Firefox 3.6.28.

Recommended browsers: Google Chrome.

Note: QueueDCU.com has not been tested on browsers older than listed above.

5.3 JavaScript

Seeing as a large portion of the website's functionality is written in JavaScript, it is extremely important to ensure that JavaScript is enabled on whatever browser is being used. Steps outlining the enabling of JavaScript on your browser are listed below:

Internet Explorer - Windows

Tools -> Internet Options.

Click on the Security tab.

Click on Custom Level button.

Select Scripting -> Active Scripting -> Enable.

Opera - Windows

Click on Opera button -> Settings.

Select Websites.

Check Allow all sites to run JavaScript.

Safari - Windows

Click on the Gear button -> Preferences.

Security tab -> check Enable JavaScript.

Safari - Mac

Safari -> Preferences -> Security.

In Web content, check Enable JavaScript.

Chrome - Windows and Mac

Customize and control Google Chrome -> Settings

Show advanced settings...

Under Privacy, click on the Content settings button.

Under the JavaScript section, Allow all sites to run JavaScript (recommended) must be selected.

Firefox - Windows and Mac

JavaScript is enabled by default.

Once the above steps have been carried out (if required), QueueDCU.com should be fully accessible and functional for everyone, on all browsers listed above.

6. User Manual

6.1 Registering a user

Before a user can use the web app, they must first register an account. This can be done by navigating to QueueDCU.com and clicking the Register link at the top of the page, on the navigation bar. They will be brought to a form, where they will be required to provide a username, email and password. To register as a lecturer, the user must use a @dcu.ie email address, whereas students should use @mail.dcu.ie email addresses.

6.2 Log In

Once the user has registered, they will be redirected to the log in page, where they must provide their username and password they used when registering.

6.3 For tutors only

When a tutor is ready, they may begin the lab session. On the lab page they have several options to choose from. They can click the "Attendance List" to view the students attending the lab.

They can click "Attending", which will start a three minute countdown allocated for the current student.

They can click "Next Student" to indicate that the current student has been helped. The next student in queue will be displayed on the lab layout.

6.4 For students only

In order to request help from a tutor, a student can click the "Enter" button, which adds the student to the end of the queue. The student must then wait for their turn to be helped. If a student no longer requires assistance, they may click the "Leave" button, which will remove them from the queue.