ROBERT GORDON UNIVERSITY ABERDEEN

SCHOOL OF COMPUTING SCIENCE AND DIGITAL MEDIA

GROUP PROJECT TEAM MANAGER APPLICATION SCALABILITY REPORT

BY

AFOLABI SAMAD

1910611

INTRANET SYSTEMS DEVELOPMENT (CMM007)

DATE: 01/05/2020

TABLE OF CONTENTS

1 ABSTRACT

The Group Project Team Manager web application is a prototype web application developed for purpose of aiding academic staff and their students for group activities. The application has potential which is why I have been commissioned to write a report of recommendation on how best to scale the web application to handle multiple users. The outcome of my research is this report which outlines my analysis and my recommendation for the Group Project Team Manager Web Application. The report looks at shortcomings of the prototype application as well as the technologies used to implement it. After the analysis comes proposed recommendations for scaling the application.

2 INTRODUCTION

The group project team manager web application is an app that allows the creation and viewing of groups assigned to students. The web application has two main users: Admin and Students. The admins are the users that create groups and students to those groups while the students can view the groups they have been assigned to. The application purpose is to help mainly academic staffs in the creation of groups for students in their classes. This report is an analysis of the prototype application already built and how to get this application to production level. One requirement of a production level application is the ability to scale to accommodate new users making use of the application. In this report, the technologies used for the prototype application are analysed and reported with an emphasis on their strengths and weaknesses. This report provides an analysis of the application in different areas but also offers recommendations about technologies that could improve the application.

3 ANALYSIS OF CURRENT LIMITATIONS

This section analyses the limitations of the technologies used in the frontend as well as the backend of the web application.

3.1 FRONT END/CLIENT-SIDE

The front end of a website is the part that users interact with. What is seen when navigating around the Internet, from fonts and colours to dropdown menus and sliders, is a combo of HTML, CSS, and Bootstrap being controlled by a computer’s browser.

3.1.1 HTML5 AND CSS

HTML5 is a hypertext mark-up language for defining structure for web pages. HTML5 provides a semantic view of structuring web pages.

Strengths:

- It supports media such as video, audio out of the box without need for third party software such as adobe flash player.
- Most browsers have support for HTML5.

Weaknesses:

- Users with older browsers would have problem accessing websites running HTML5.
- Example is Internet Explorer (IE) which is still being used by users with older Operating systems such as Microsoft windows 7 or XP.
- Another example is the use of older models of smartphones by users.
- Also, not all devices have very good responsiveness to HTML5.

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML. Cascading Style Sheets describes how elements should be rendered.

Strengths:

- Easy to implement.

Weaknesses:

- CSS cannot perform any logical operations.
- CSS cannot read or process any files.
- CSS cannot interact with databases which hinders user specific customization options.
- CSS is not compatible across all browsers making the task of development tedious.

3.1.2 BOOTSTRAP

A free front-end framework for faster and easier web development. Bootstrap includes HTML and CSS based design templates for forms, buttons, tables, navigation, image carousels and many others. Bootstrap also provides the ability to easily create responsive web pages.

Strengths:

- Easy to setup and implement.
- Very detailed documentation provided on how to style webpages.
- Reduces the amount of CSS to be written.

Limitations:

- Bootstrap is tied to jQuery which increases the size of files being loaded thereby slowing downloading of web pages.

3.2 BACK END

The back end of a website consists of a server, an application and a database. The back end is where building and maintenance of the technology that powers those components which, together, enable the user-facing side of the website to exist in the first place. The technologies used in the prototype application include PHP as the programming language, MySQL as the database of choice and Apache as the web server.

3.2.1 APACHE HTTP SERVER

An HTTP server is a piece of software that understands Uniform Resource Locators (web addresses) and Hypertext Transfer Protocol (the protocol browsers use to view webpages). It can be accessed through domain names (like intranet.com) of websites it stores and delivers the content to the user’s device. The Apache HTTP Server is an open-source HTTP server used in operating systems. The purpose of the Apache HTTP Server is to provide a secure, efficient and extensible server that provides HTTP services.

Strengths:

- It is easy to install and set up.
- Large community support reduces time looking for solutions to problems if an error is encountered.

Weaknesses:

- Large consumption of processing power especially when there are a lot of files to serve such as images.

3.2.2 PHP

PHP is a server-side scripting language used to develop web applications, static websites and Dynamic websites. PHP is fully represented as Hypertext Pre-processor. PHP scripts can only be interpreted on a server that has PHP installed. Client computers accessing the PHP scripts only require a web browser.

Strengths:

- Quick to install and setup.
- Large community support makes finding solutions to problems relatively easy.
- Its fast and performs operations quickly.
- Allows embedding of code into the webpages allowing for a lot of customization.

Weaknesses:

- It does not offer modularity like other object-oriented languages such as python or java.
- The code is hard to maintain when embedded a lot into web pages leading to bugs.
- Type conversion (which is the conversion of objects from one data type to another) in the language is confusing.
- Error handling is poor as it does not provide detailed assessments of what went wrong making it hard to fix.

3.2.3 MYSQL

MySQL is a popular relational database management system. It stores data in tables with columns and rows. Data is retrieved from the database using Structured query language (SQL). The implementation of MySQL used for the prototype application is not very relational. While each table consists of its own primary key. This primary key is not synchronized across other tables making each table a singular data storage that does not relate to other tables. This makes the table inefficient making queries performed are specific to one table and not across multiple tables.

Strengths:

- Its very easy to install and set up.
- PhpMyAdmin makes creating a database a relatively easy task.
- Querying the database is a small feat due to the Structured query language.
- Its very fast for low-intensive applications.
- It has a strong community support and lots of tutorials available which made finding solutions to problems encountered easy to find.

Weaknesses:

- It got slower as more data was sent to it.
- The database structure got more confusing as additional data was inserted and more tables created.
- The core debug program is not very effective at clarifying problems.

3.3 ATTACK VECTORS AND POTENTIAL VULNERABILITIES

HTML FORMS

- The input sections of the html forms are not properly secured against attacks such as SQLInjection. An attacker could provide the forms with commands to trick the PHP interpreter into providing access to sensitive information.
- Special characters are mostly allowed in the form inputs for the website making it less secure against injection attacks.

EXPOSED ACCESS CREDENTIALS

- The database connection php file is in the root folder which makes it easy to become exposed.

4 RECOMMENDED TECHNOLOGIES

This section discusses the different alternative technologies that can be used by the prototype application which may provide better scalability options for development.

4.1 FRONT END / CLIENT-SIDE RECOMMENDATIONS

4.1.1 AJAX

Asynchronous JavaScript and XML (AJAX) is not essentially a technology rather a term which describes an approach to using several existing technologies together including HTML, CSS, JavaScript, XML and XMLHttpRequest. When these technologies are combined in the AJAX model, web applications can make incremental updates to the user web interface without reloading the entire user web application page. This makes the application faster and more responsive to user actions. Although the X in AJAX stands for XML, JSON is used more often than XML nowadays because of its many advantages such as being lighter and a part of JavaScript. JSON and XML are both used for packaging information in the AJAX model. An outline of what AJAX accomplishes with development include:

- Update a web application page without reloading the web page.
- Requesting data from a server after the web page has loaded.
- Receiving data from a server after the web page has loaded.
- Sending data to a server in the background.

The use of AJAX would reduce the load time between pages in the application. The time it takes to add new student and generate the list of students would greatly reduce or the time it takes to randomly generate a group. AJAX would provide a near real-time effect for the application.

4.1.2 ANGULAR JS

AngularJS is a JavaScript-based open-source front-end web framework to resolve many of the challenges encountered in developing single-page applications. AngularJS extends html functionality to provide dynamic content in the web application. This would greatly reduce the amount of php code on web pages. AngularJS would also increase the load time of the web application pages since an extension of html functionality means less external linked files.

4.2 BACKEND / SERVER-SIDE

4.2.1 LARAVEL WEB FRAMEWORK

The Laravel web framework is a web framework used for php web development. It is like Django in Python, Spring boot in Java for developing web applications. Laravel framework operates using the Model-View-Controller pattern in developing its web applications. A web framework is a collection of tools to make development of web applications easier. The use of Laravel would greatly reduce the time it would take to develop a web application. The framework would also provide libraries for functions such as User authentication and authorization, generating admin dashboard to monitor the application, testing and automation of some of the web application functionalities, search function as well as other third-party libraries for different aspects of the web application.

4.2.2 MONGODB

MongoDB is a database program. It is categorised as a NOSQL database program that uses documents that are JSON-like instead of tables like MySQL. MongoDB is a relatively new database program compared to MySQL. The documents in MongoDB have varied structures. MongoDB uses its own query language called MongoDB query language. MongoDB does not require the developer to define the structure of the document first as seen in MySQL. The documents can be easily changed by adding and removing fields. One major advantage of MongoDB is that documents are allowed to have their own unique structure. Fields can be added to the document at any period and allowed to store any type of value. This type of functionality in a relational database would require the database to be restructured. In MongoDB, keys in a document must be unique but the same keys are allowed in other documents. MongoDB document format offers an improved performance over relational databases such as MySQL. MongoDB offers better security against SQL injection attacks due to it not having a language to parse. MongoDB also offers a very detailed documentation as well as an active community. Enterprise Grade support is also available for large scale database storage.

5 SCALABILITY SOLUTIONS

The prototype application currently being used operates on a 1-TIER architecture. The architecture has 3 layers namely:

- Client
- Server
- Database

The 1-TIER architecture is when all three layers are kept on the same machine. This is only used in prototyping of the applications. To achieve a scalable solution that can cater to multiple users, the application will use 3-tier architecture then N-tier architecture.

5.1 3-TIER ARCHITECTURE

In the 3-TIER architecture, the layers are do not exist on the same machine. The Server acts as an intermediary between the client and the Data access logic. Any request made to the database is sent to the server first and the server queries the database. The components are much more maintainable while operating using 3-TIER architecture and the components can be used for other purposes. This is the model used by a lot of modern websites. It can handle a lot of users but also has limits.

Limitations of 3-TIER architecture:

- It creates new queries for every request that is sent to the server.
- The speed at which the requests are replied depends on the amount of data to be searched through, the number of queries in the queue.

5.2 N-TIER ARCHITECTURE

To scale the web application from the 3-tier architecture to the N-tier architecture, the following requirements will be considered:

- Availability: The group project manager web application must be available all the time to cater to both administrators and students who wish to use the application. Students may wish to view groups at any time and administrators may wish to answer student requests at any time.
- Performance: The group project manager web application must process requests at a fast pace. The lag time between page loads must be minimal. Administrators must be able to view any changes made as soon as possible. Example is if any group change occurred, the updated groups list must be available at a fast pace even if there are many connections to the server.
- Reliability: The services offered by the group project team manger web application must be reliable which means any similar request made at any point in time must have similar answers when processed. Example is that the groups list must be the same every time its viewed unless explicitly updated by the administrator.
- Scalability: The services offered by the group project team manager web application must be easily upgradeable to accommodate new content. An example is if the number of images uploaded by the users for changing their profile pictures or the

number of student entries increases, or the number of groups created by different admins increases. The data storage used to store all these data must be easy to increase to accommodate the new data.

- Manageability: The group project team manager application must be easily manageable. The user accounts must be easily manageable, permissions can be used to limit the amount of control users have in the web application.
- Cost: The cost to create and implement any of the services offered by the group project team manager application should be efficient. Cost can be in monetary value and time.

In the 3-TIER architecture, any requests made by the users (Admin and student) of the group project team manager application will be sent to the server. An example of requests that will be made could be creation of new groups by admins and addition of new students by admin. These two requests would be handled by the server as one function.

A better option would be to separate the function into two services. One is to handle creation of new students while the other is to handle the creation of new groups. Splitting the different functions into their own services would lag time between requests and reply.

To deal with an increase in the data being sent to the database, to add redundancy for dealing with loss of data in cases of emergency, a backup database storage will be implemented for the group project team manger application. This backup storage will exist in a different geographical location. If the requests and dataset get too large to deal with, scaling the server will be implemented. There are two types of scaling:

- Horizontal scaling: Involves adding more storage and processing power to the server.
- Vertical scaling: Involves adding more nodes to the server to make use of the same services.

The group project team manager application will make implement vertical scaling of the server at the initial stage. For each node, one will be active while the rest go dormant to add redundancy. The nodes will become active only when needed.

5.3 SCALING METHODS

- Caching: A cache added to the server will be used to store data that is requested for frequently such as group list and student list. Caches helps to serve the data faster. The caches are queried first before the databases are queried.
- Proxies: They collapse similar requests into a single request to query the database. They are useful given that the users of the group project team manager application will be making similar requests.
- Load Balancing: They are placed between the client and the server to balance the requests from the client between the nodes of the server. There are different methods for load balancing.
  - Random Node
  - Round Robin
  - Criteria Based
  - Location Based
- Queues: Each request is added to a queue. It is not very good for reading data from the database. The group project team manager application will have different users using it at the same time creating groups, adding new students, submitting requests and so on. Queues would slow the process which does not meet the performance requirement.

5.4 IMPLEMENTATION

The recommendation for the implementation of group project manager will make use of the following technologies and scaling methods:

- Multiple server nodes
- Proxies and cache scaling methods
- A database backup storage
- MongoDB
- PHP Laravel Web Framework
- HTML, CSS, JavaScript, AJAX, AngularJS for the front-end

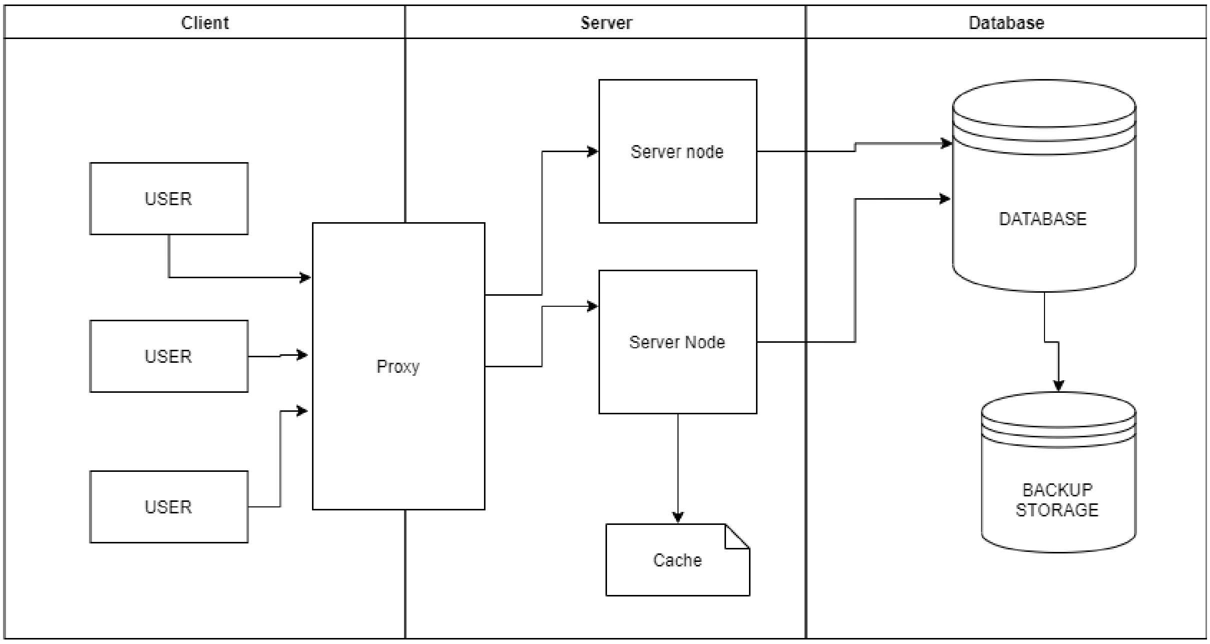

Figure 1: A simple diagram illustrating the Implementation

6 APPLICATION SECURITY

6.1 REGISTRATION SECURITY

- The input component of the form will have special characters provided as input removed to prevent SQL Injection attacks and an error generated.
- Adding a security question component to the form listing questions and asking the user to provide answers to them. This would help when trying to verify the user if any issue arises.
- The password field should have the characters not viewable.
- The input component for email should be formatted for email addresses only.
- Providing a phone number component in the form for users to help users if any issue arises.

6.2 LOGIN SECURITY

- The input component of the form will have special characters provided as input removed to prevent SQL injection attacks and an error generated.
- The password field should have the characters not viewable.

6.3 SESSION SECURITY

- The user’s session will be destroyed as soon as the browser is closed to prevent attacks such as session hijacking.
- The user’s session will be destroyed as soon as the user logs out.

6.4 STUDENT AND ADMIN PROFILE

- Input fields in any form accessible to the student or admin will have special characters removed to prevent attacks such as SQL injection attacks.
- Access given to the admin and student will be properly configured.
- Admin will be given access to add students, view students, create groups, view groups, change profile picture for admin, view requests but not submit requests.
- Student will be given access to view students, view groups, change profile picture for student, submit requests but not view requests by students.

6.5 CODE SECURITY

- File containing connection to the database will be placed outside root to avoid exposed access credentials.
- Passwords of users will be encrypted using SHA256 which is more secure than md5 before they are sent to the database.
- GET requests will be disabled for all forms. POST will be the standard for form submissions.
- CSRF will be added to all requests made to the server.
- SQL queries will be defined in advance to prevent SQL Injection attacks.

7 CONCLUSIONS

In this report, analysis of a prototype of an application for group project team management was carried out, recommendations outlined for scaling up the application from prototype to production level. Analysis of the various technologies was carried out. This report does not outline all the steps in verbose detail but outlines requirements and acts as a recommendation. If the recommendation is accepted, a more detailed documentation will be developed with a team to analyse all technical requirements that are necessary to get the application to production level. In summary, this report is a recommendation as evidenced by the lack of technologies such as Amazon web services. The principles behind the technology is explained under system architectures, the same principles apply. If the recommendation is accepted, a team of developers would have to consider the pros and cons of the different technologies that implement the architecture.

8 REFERENCES

1. https://blog.udacity.com/2014/12/front-end-vs-back-end-vs-full-stack-web-developers.html
2. https://brandongaille.com/19-html5-advantages-and-disadvantages/
3. https://developer.mozilla.org/en-US/docs/Web/CSS
4. https://developer.mozilla.org/en-US/docs/Web/javascript
5. https://www.freecodecamp.org/news/the-advantages-and-disadvantages-of-javascript/
6. https://www.uplers.com/blog/what-are-the-pros-cons-of-foundation-and-bootstrap/
7. https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_web_server
8. https://www.guru99.com/what-is-php-first-php-program.html
9. https://www.wordfence.com/learn/understanding-php-vulnerabilities/
10. https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX
11. https://blog.panoply.io/mongodb-and-mysql