Asking Net: Question and Answer Platform

Fernando P.R.L.

Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka.

randil.fernando.14@cse.mrt.ac.lk

*Abstract*— Knowledge is a human requirement which fulfills their aims and give them a successful future. Most of the knowledge humans gain at the stage of them as students. Students use internet the most to find solutions for their problems and to publish their knowledge. Although there are many question and answer platform to help students publish questions and to find answers they are riddled with problems of finding answers for their academic related problems at one place. Academic related questions occurred to students can categorize according to modules. If there is a question source which is organized according to modules then it will help students to find answers and to exchange their knowledge with fellow students. Asking question and answer platform provide undergraduate students and lecturers a web platform to submit, search, rate questions and answers. Additionally, lecturers can use asking platform to view overall report on questions submitted.

Keywords— Web platform; Service Oriented Architecture; JSON; REST; Model View View Model; Model View Controller;

# Introduction (*Heading 1*)

Internet held a major role of finding solutions and exchanging knowledge among humans. As a result of the popularity of the internet most people use it as their main information source and to solve their problems. Although there are various type of question and answer platforms available. But there is no proper one which is customized to the usage of undergraduates. Questions occurred to undergraduate students mainly on the modules that they study at university premises. Finding answers for academic related activities using existing question and answer platform would not provide sufficient information for that and the answers may not according to the marking strategy at the university. Hence ‘Asking net’ web application provide very accurate and easy solution to this by providing a question and answer platform which is specialized for undergraduate students.

Recently, much of the growing interest has been pursued in the context of using mobile phones to surf the internet. Hence a web site which only designed for desktop computers is not sufficient. With the introduction of client side rendering classical web sites are converted into web applications. One of the key drivers behind a web applications success is the types of devices it supports and user interface and user experience. Although client side rendering offer user a better user experience initial loading time of the application is the problem. Hence the application need to optimize in order to keep a balance.

Asking platform was built to give undergraduate students a good user experience buy providing them a question and answer platform which was customized according to their needs and to provide lecturers a source which can be used to interact with students. Asking question and answer platform implemented by categorizing questions according to their relevant module. It gives a better user experience to undergraduates. Also, lecturers can create their accounts through the system administrator and it gives them ability to view and analyze questions submitted and to get an overall understanding of students.

This paper describes how Asking question and answer platform was designed and developed. Section II surveys related work in building application. Section III provides an overview of the system. System models, design, implementation and testing of the application are described in latter sections respectively. Final section of the paper describes conclusion and future work.

# Literature Review

This section provides an overview of some web sites that are available as question and answer platforms to find answers and to exchange knowledge.

The most rated question and answer platform is Stackexchange network and its most famous web site Stackoverflow [14]. Stackoverflow [14] offers users to submit questions and answers. The web site is designed for software and programming related questions. The rating system and raking system of website offers users more functionalities. The structure and the functions of the stack overflow is remarkable.

Quora [15] is another famous question and answer platform which is for any type of questions. Quora enables users to submit questions using anonymous.

Moodle [16] which is using inside many university premises to automate and organize the learning process offers forums for exchange knowledge and to ask questions. Forums are categorized according to modules and only the students who are studying them can submit and view forums.

Stackoverflow [14] and Quora [15] offers better user interfaces and functionalities. But undergraduate students need a question and answer platform which is designed according to their modules and the questions and answers need to format according to their learnings. Moodle [16] offers a forum which is designed and categorized according to modules but the functionalities are limited.

Asking question and answer platform is a solution to overcome this by combining functionalities of remarkable websites like Stackoverflow [14] and Quora [15] and design and implement those to make it more suitable for undergraduate students. This web site is an enhancement over forum posts of Moodle [16] which offers a user-friendly interface and functionalities to use during teaching and learning. Asking also provide lecturers a way to get an overall understanding of the student performance.

# Overview Of The System And Process Followed

Rational Unified Process was used as the process model, which was based on four main phases; inception (where most of the planning work was done), elaboration (where most of the design work was done), construction (where the real implementation was taken place) and transition (where deployment and maintenance work were carried out). The construction phase focused on three main iterations, which included implementing student use cases, implementing lecturer use cases and implementing admin use cases and security. During the inception phase, the business model and the requirements of the project were specified. System architecture was designed in the elaboration phase. Construction phase was used for implementation of the features in accordance to the system architecture with proper testing. After each iteration implemented functions tested against unit test suit. During the transition phase, refactoring of the code, system testing, overall bug fixing and deployment were mainly carried out.

# System Models

## System Requirement Specification

This section describes the system requirement specification, which was launched at the inception phase of the project

Functional requirement of asking question and answer platform can described under data to be entered into the system, operations performed by each screen, work-flows performed by the system, system reports or other outputs, system meets applicable regulatory requirements. Asking questions and answer platform has three user types.

Students can search for questions and answers to find answers for their problems. Also, is there is no questions related student can submit new question providing title, module, topic, description and question tags. Then the users who subscribed for the module will get notification. Students can submit new answers for questions submitted by fellow students. Also, rating system improve the search result. Student can rate question and answer to increase or decrease its value. Highest value question or result will be at the top of the search result. Lecturers can submit new questions, rate question and answers, update questions and answers, remove questions and answers and view report. Students and lecturers can subscribe for modules to get notifications. Reporting section held a major role in lecturer accounts. Administrators capable of adding lecturer accounts, manage user accounts and manage questions.

Main non-functional requirements of the system are the speed. Search result should generate within 5-seconds time period. Most of the users interact with the system by searching questions and answers. Therefore, speed of the searching query held a major role in user experience. System should be up for 95% of the day and 5% of the day can be use by system administrators to manage and review system functionalities.

## Use Case Diagram

Mainly there are three types of users; student, lecturer and administrator. In Fig. 1 show that all the use cases related with platform. System is divided into four sub systems; Module management, Report and notification, question and answer management, user management. Fig. 1 will display the user types who can interact with each subsystem.

Fig. 1. Use case diagram

# System Design

## Architecture of the System

The system follows three-tier architecture. Fig. 2 shows the architecture of the system since asking consist of two parts: one web portal and mobile application. The mobile application is for students and it gives functionalities to search questions and to add new questions and to add answers.

Web application is for all the user types who can interact with the system where it gives search questions, add new question, add new answer, add new module, update module, update question, update answer, remove question and answer, blacklist account. There are three type of users involving with the system: student, lecturer and administrator. Asking question and answer platform provide users with all the functionalities related to a question and answer platform and the advantage of using asking question and answer platform is that it is customized according to undergraduates and questions available inside the platform is specialized to the undergraduate students and categorized according to modules and topics which is easier to browse and search.

Server side of the application use RESTful [13] architecture as the main architecture of the system. Client side interact with web server through JSON [4] objects. All the outputs of the server are JSON object type. Since the server is implemented according to the RESTful [13] architecture token based authorization is used and token is saved on the client side. Client interact with web server with http requests and then that request will invoke a method inside relevant controller finally resulting response with JSON [4] type.

Token based authorization generate the token as the result od the login request and then send back to the client side. Client then save it and send it with every other request it sends. Then the server can authorize request by decrypting token and viewing its content.



Fig. 2. Layered architecture of the system

## Process View

Asking question and answer platform use client side rendering and used angular.js [1] as the front-end framework. Therefore, server is accessed only if there is a data retrieval is needed. Fig. 3 shows the add user functionality sequence diagram.

When user requested add new question page client side will generate requested page using reusable components. Since the first step of the above use case does not include data access method no request sent to the server. When there is a data retrieve method then a function inside the angular service is executed and a http request send to the server. Then the URL matcher of the server side will match that request with the relevant route and then the relevant method inside controller will be called.

Controller method will invoke a query on model class which is a persistent class and the object relational mapper will query the database and retrieve the result. Then that result will send back to the server and view will update according to the retrieved data.

Since angular.js provide mapping mechanism with html document object model it is not need to directly manipulate html document.



Fig. 3. Add new question sequence diagram

## Database Design

Mongodb [12] used as the persistent database. Model classes that was implemented on the server were mongodb persistent classes. And those classes were implemented using the library which consist of an object mapper and a database driver. Database schema and the server side classes are identical to each other. Since mongodb [12] is an object based database join operations between mongodb [12] collections are minimized. Most of the collections are isolated with other. Modules, questions, Users are isolated collections while question has answer as its embedded document.

# System Implementation

## Implementation Procedure

Asking question and answer platform is a web portal. Back-end of the application is implemented using node.js [6] and express.js [3] framework. As the persistent layer mongodb [12] was used. Angular.js [1] used to front-end development.

Software engineering process followed Rational Unified Process methodology. Construction phase consist of four iterations and at the end of each iteration testing and bug fixing implemented. Implementation of functional requirements divided within second and third iteration of the construction phase while first iteration consists of architecture implementation and configuration and fourth iteration consist of security implementation and overall bug fixing.

Back-end of the system used node.js to develop system. JavaScript used as the programming language. Express.js [3] which is a minimal framework used to handle routes, middleware. Express.js is a minimal framework and all the route points need to implement manually hence it is more flexible than other frameworks. Asynchronous behavior of JavaScript increased the software performance and architectures relevant to that behavior was used. RESTful [13] architecture used to develop the server hence all the server requests and responses passed through JSON [4] type. Authentication process was implemented using a token base authentication. Token used to authenticate was JSON [4] web token which is called JWT [6]. Token used single time interaction with database and the token is then saved in the client side. When there is a request the token must send with it. No session was maintained on the server. User email confirmation used SMTP protocol to send emails. Nodemailer library used to send emails.

Front-end development used angular.js [1]. Model-view-viewmodel architecture used at the front-end. Asking platform used client side rendering and used reusable components. There are two routers in the application; view router and REST [13] API router. View router is capable if switching views with the user interactions. The application was a single page web application which consist only one page and generating different views by changing components.

Testing process used jasmine framework to implement test cases. Karma [10] test runner used to automate the unit testing process and protractor used automate end to end testing process which involved testing REST [13] API. Selenium IDE [11] used to automate the user interface testing.

## Main Interfaces

As shown in Fig. 4 home page contain sidebar which will give the functionalities available to the logged-on user. Featured modules list contains list of modules which has highest number of questions submitted which is organized according to the descending order. Functionalities available to users will change with the user type hence options available in the sidebar will slightly change with the user. Featured module section is common to all user accounts.

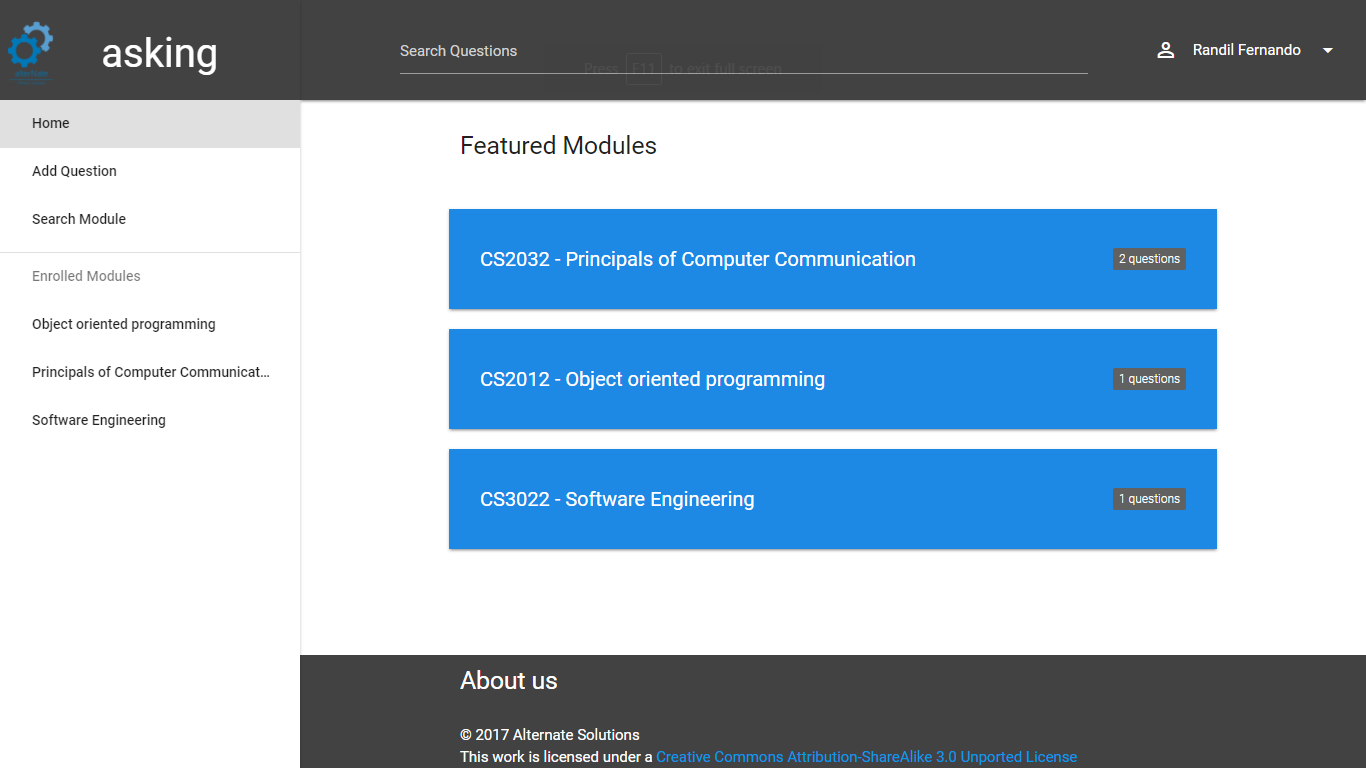


Fig. 4. Home user interface

As show on Fig. 5 students can add new questions to the system. Module can be searched through a search bar and then number of matched module names will be displayed to the user to choose from. After the selection topics will be updated according to the selected module and then the user can select a topic which correctly identify the question. For further identification tags can be added. And then the user can submit new question to the system

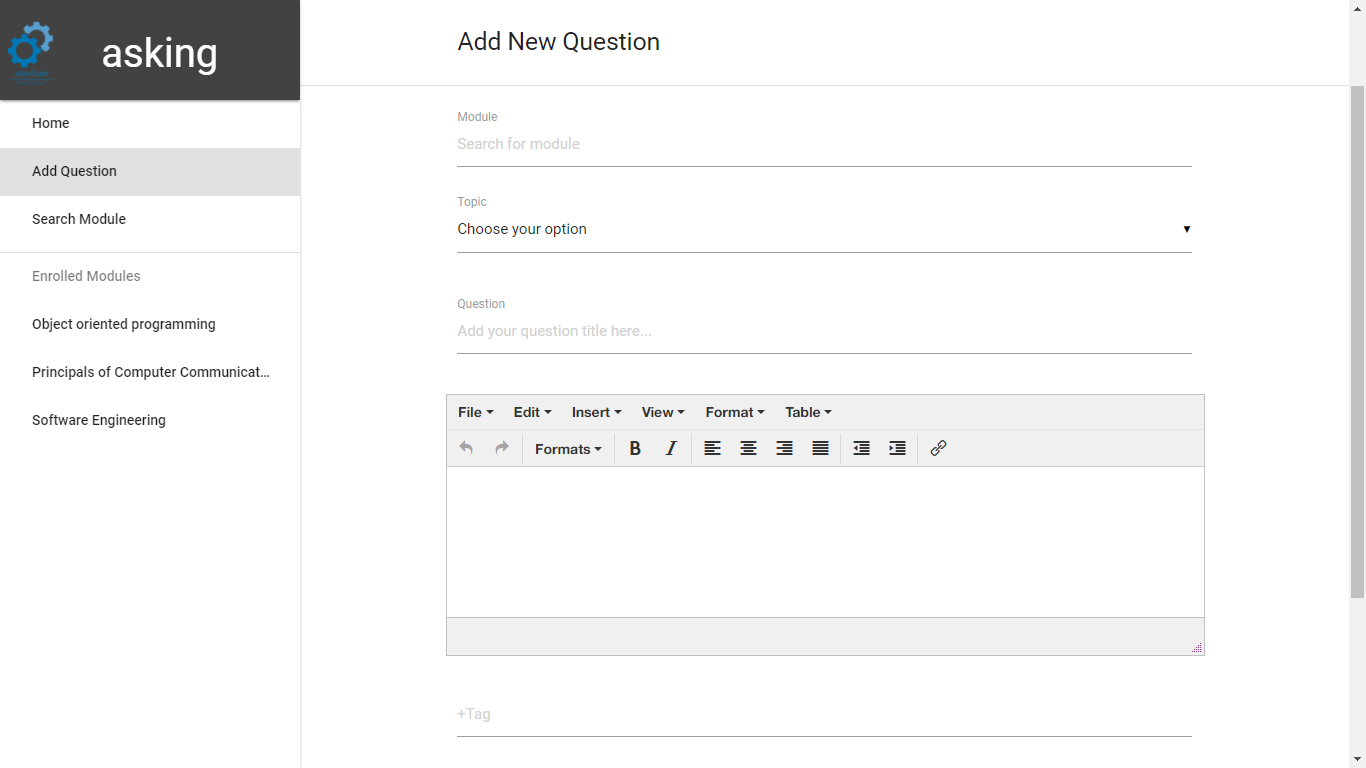


Fig. 5. Add new question user interface

As shown in the Fig. 6 student, lecturer can search for questions buy providing keywords. Search result will be organized according to the number of words matched with the questions. When user enter a search, prepositions are removed from it and then search process will further. Also, user can browse questions through modules.

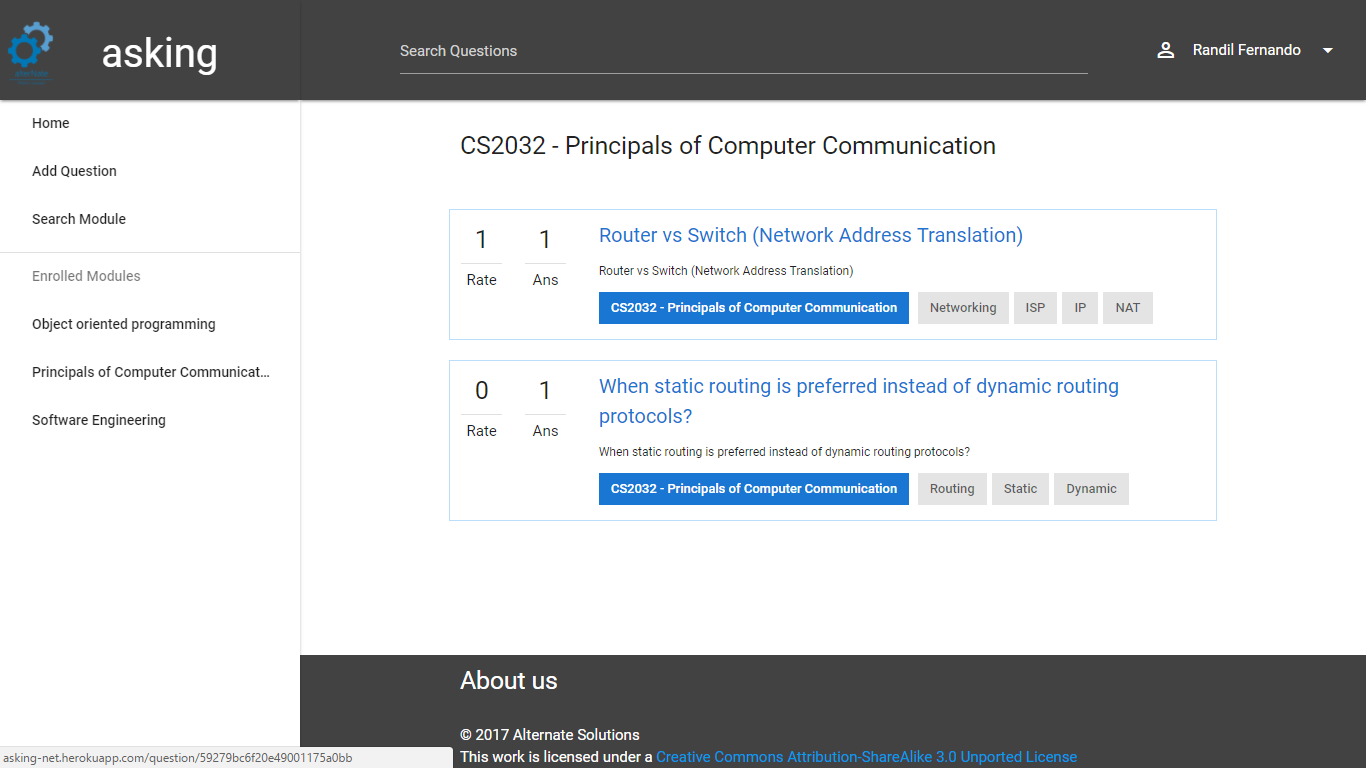


Fig. 6. View search result user interface

As shown in the Fig. 7 lecturers can generate an overall report. Report displays the number of questions submitted to the system with the number of answered questions to get an overall understanding. The bar chart shown above will display a comparison between answered and unanswered questions. Also, the report will display a detail of questions submitted to the system categorized by modules.

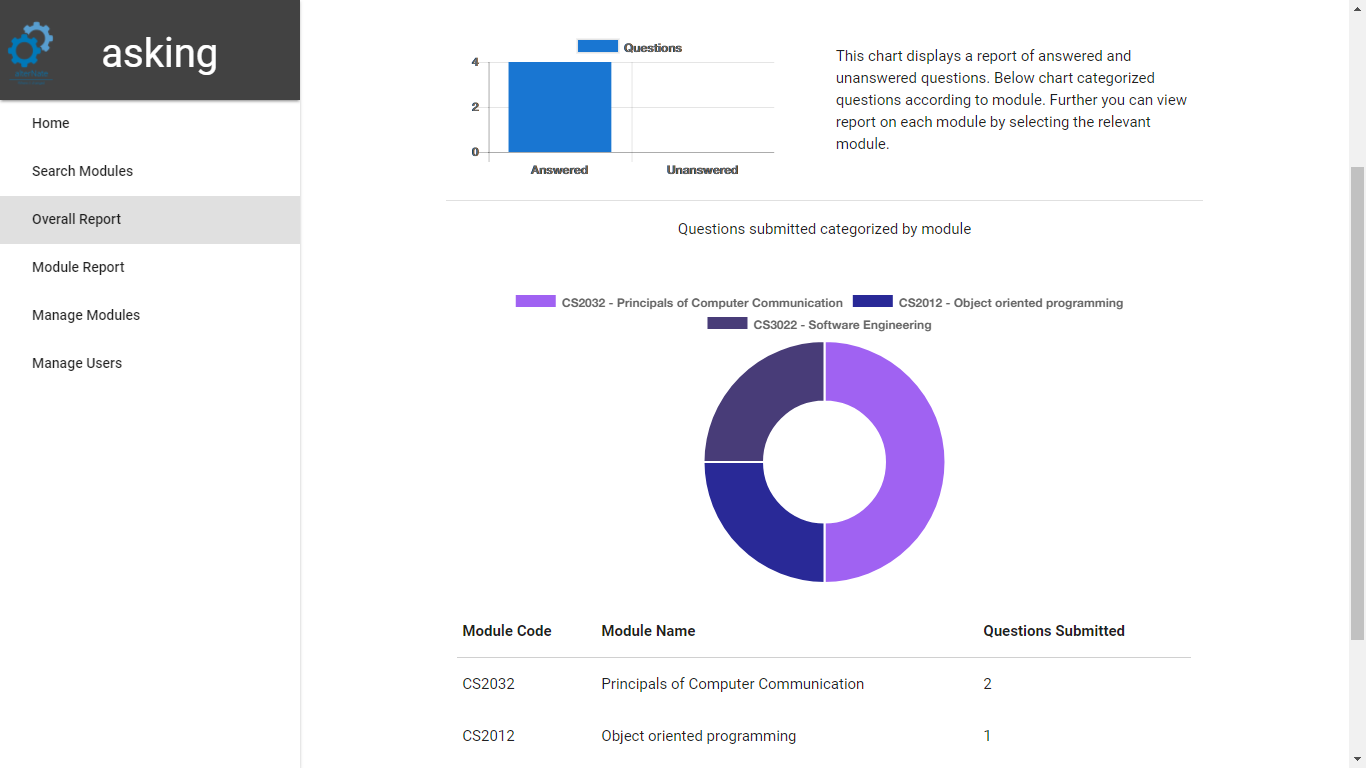


Fig. 7. View overall report user interface

# System Testing And Analysis

The web application was tested using google chrome web browser, Firefox web browser and Microsoft edge web browser with ADSL internet connection with 7.2mbps bandwidth. Web application was tested with both large screen size and small screen size (mobile devices).

A select number of methods will be tested in a couple of classes with black and white box testing to ensure that they function correctly. Asking question and answer platform based on front end angular.js [1] and back end node.js [9] while JavaScript on both front-end and back-end. There are lots of frameworks that provide unit testing functionality to node.js [9] and angular.js [1]. Since both the front-end and back-end use JavaScript same framework can be used to implement test cases. Jasmine [5] framework used as the unit testing framework it is a behavior driven development style which is focusses of the language used. Karma [7] which is an automated test runner used to automate unit testing of the front-end. Also, protractor [10] used to automate the end to end testing process. On the server side, mocha [8] with chai [2] assertion library used to automate the testing process.

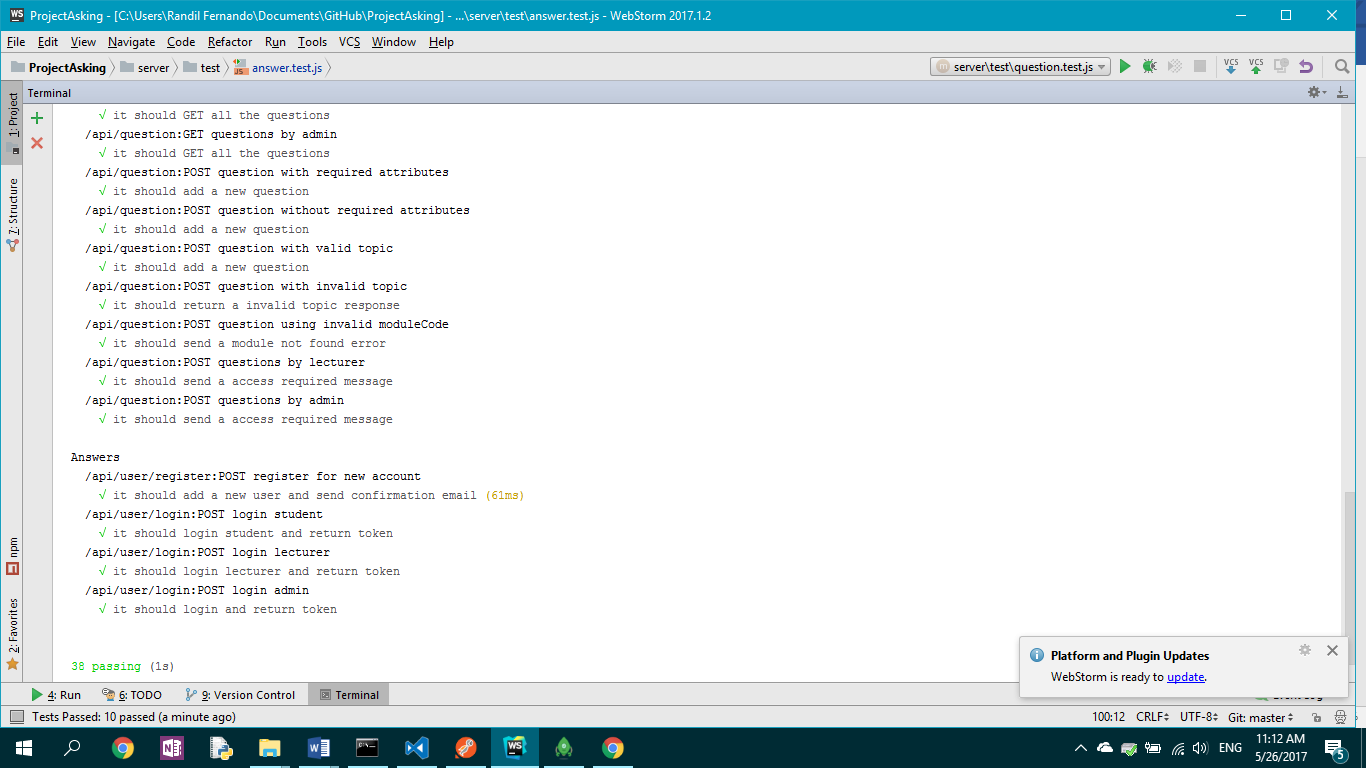


Fig. 8. Results of unit tests

Automated UI testing used for testing the functionalities of user interface controls. Different graphical user interface controls were tested using selenium IDE [11] and selenium web driver, added to a user interface control map and a series of actions were recorded and tested. Testing using Code Analysis Window was useful to manage and resolve code analysis warnings. Static code analysis was used to analyze source code prior to the execution in order to detect issues on codebase, identify potential design, interoperability, performance and security. This enhanced the quality of software by examining common defects and violations. Testing using Code Clone Analyzer was useful to search duplicate code throughout the solution. This analyzer detects direct copies and fragments, which differ in names of variables and parameters and in which some statements have been rearranged

Web site tested with multiple device types (desktop computer, mobile phone) to deliver an optimal user experience by making a responsive user interface as seen on Fig. 9.

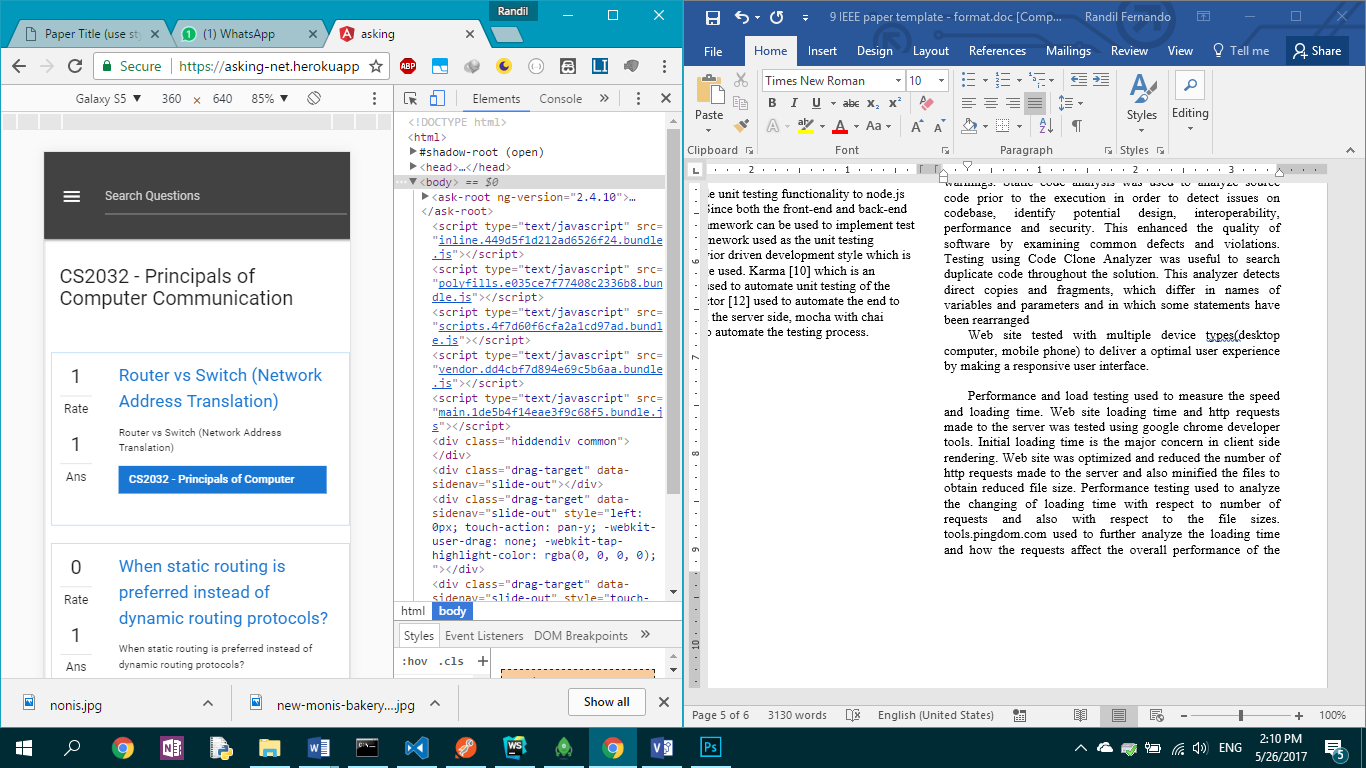
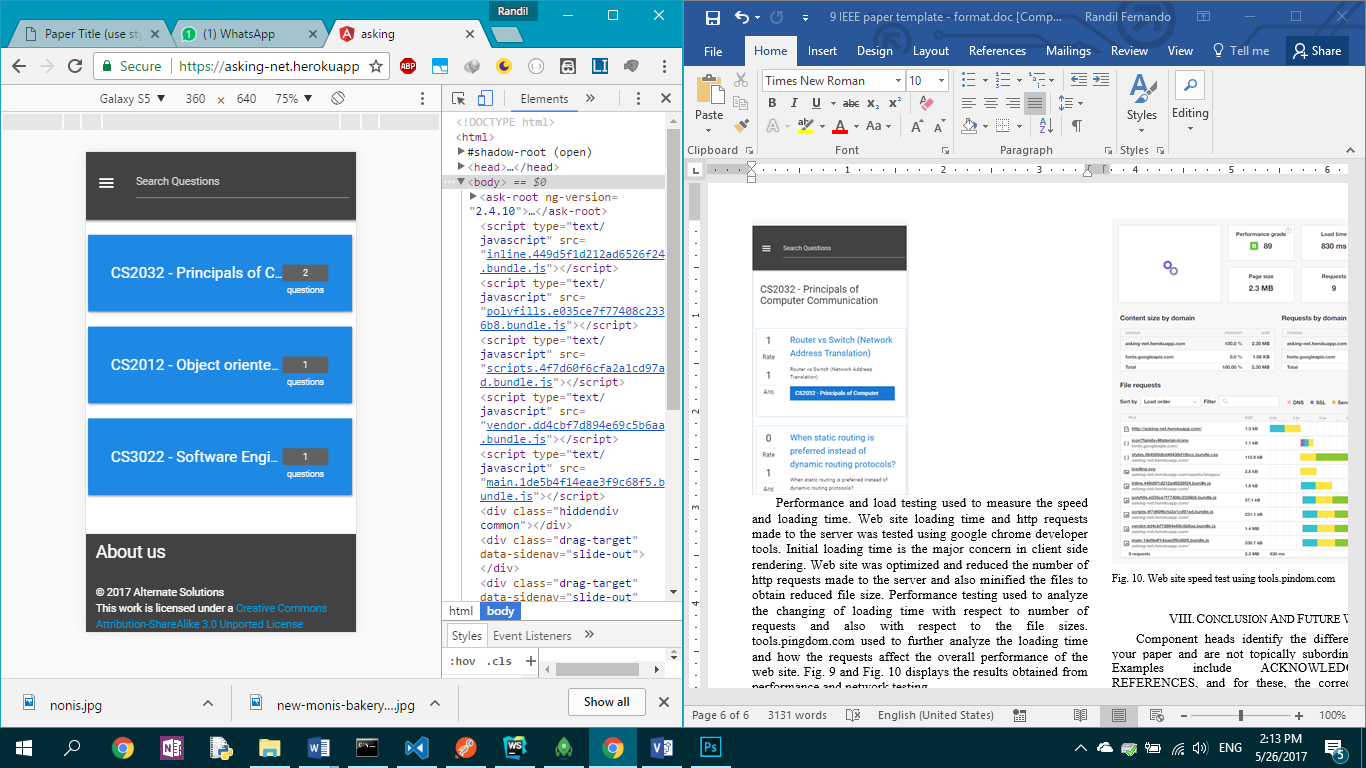
 

Fig. 9. Mobile view

Performance and load testing used to measure the speed and loading time. Web site loading time and http requests made to the server was tested using google chrome developer tools. Initial loading time is the major concern in client side rendering. Web site was optimized and reduced the number of http requests made to the server and also minified the files to obtain reduced file size. Performance testing used to analyze the changing of loading time with respect to number of requests and also with respect to the file sizes. tools.pingdom.com used to further analyze the loading time and how the requests affect the overall performance of the web site. Fig. 10, Fig. 11 and Fig. 12 displays the results obtained from performance and network testing.

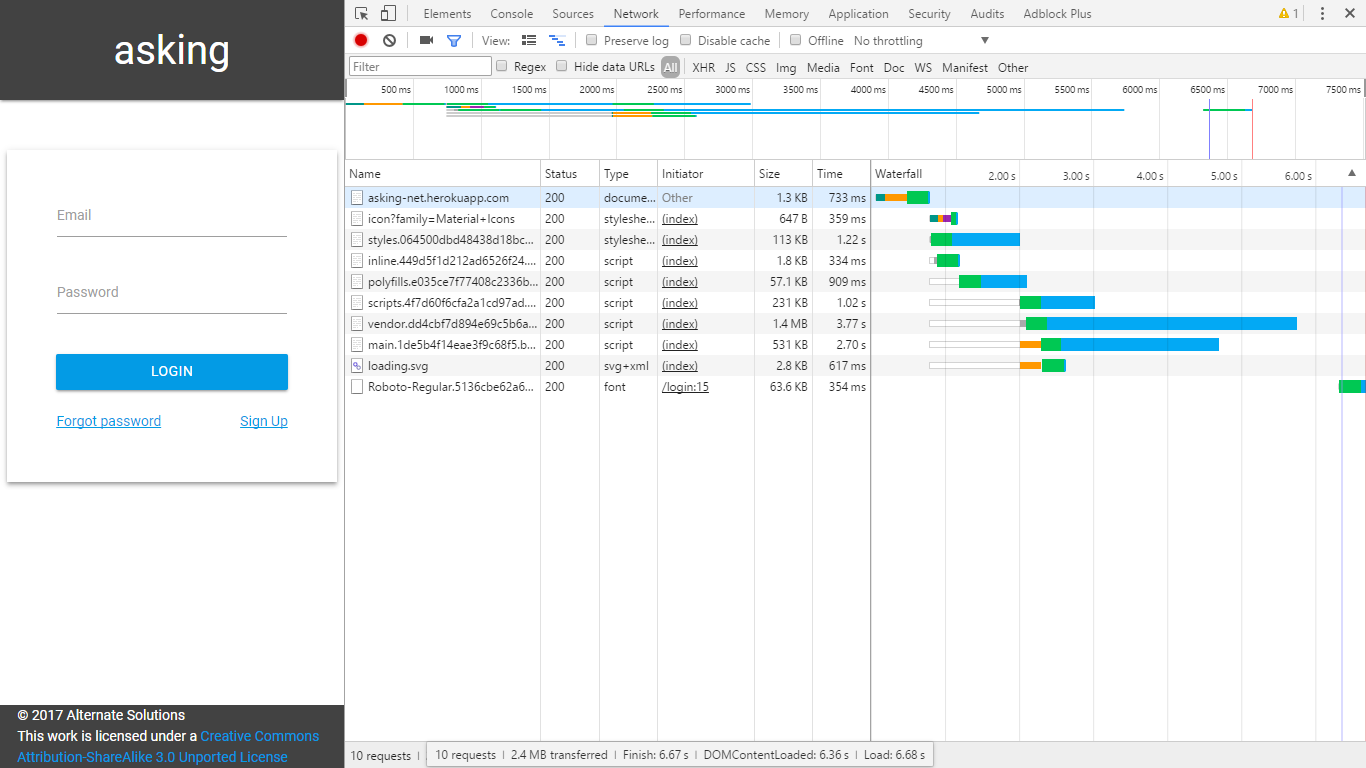


Fig. 10. Network performance analysis using google chrome developer tools

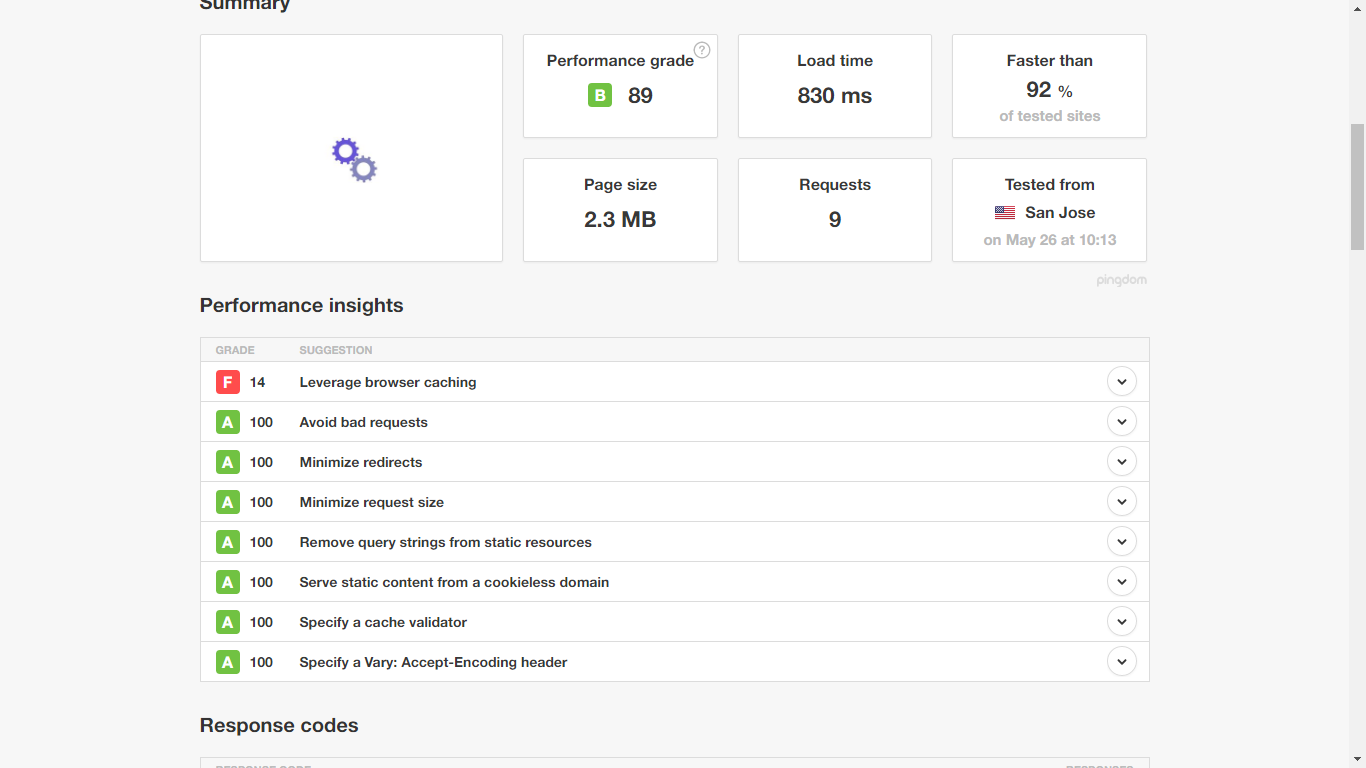


Fig. 11. Web site speed test using tools.pingdom.com

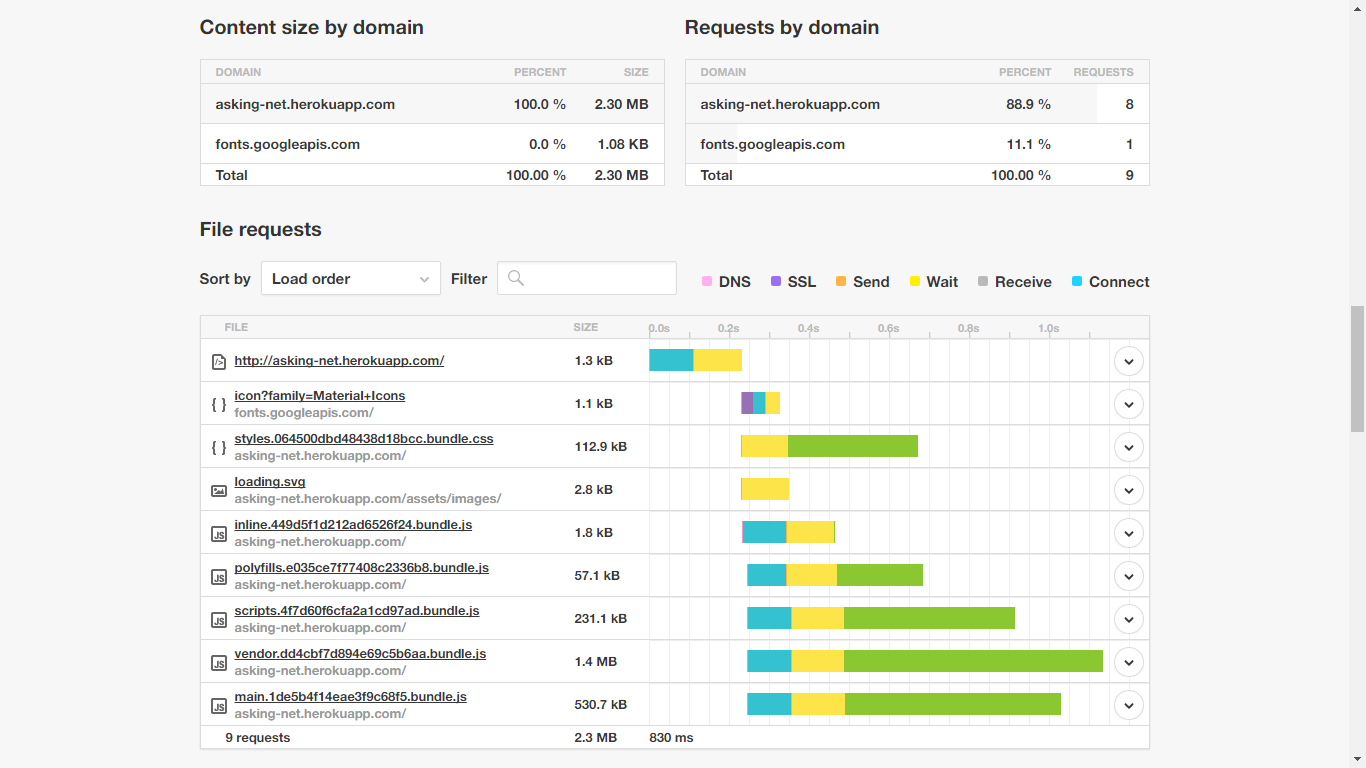


Fig. 12. Web site speed test using tools.pindom.com

# Conclusion And Future Work

Asking platform is a web application which provide question and answer platform for undergraduate students and lecturers which is designed and customized to suite university academic related questions. Main purpose of the asking platform is to provide a customized platform which categorized questions according to modules and to provide lecturers a source to view report on questions occurred within students and to get an overall understanding. Asking platform provide students a media to find answers to their academic related problems and to exchange their knowledge with fellow students. Students can search questions, browse questions which is organized according to modules. Lecturers have higher privilege than students hence more functionalities. Lecturer can modify question and answers, view report of question submitted. Students and lecturers can rate questions and answers. Rating will increase the quality of search results. System administrator is capable of remove questions and answers, create lecturer accounts, manage user accounts. System administrator provided with these functionalities to improve and to maintain the quality of the system. Also, the lecturer is provided with updating question and answers to maintain the quality of answers.

When there is huge number of users accessing the system sometimes there can be situations where more than one user viewing and modifying the same question or adding answer to the same question. Since the system is not syncing sometimes there can conflicts between users. Therefore, real time implementation of the system is needed and socket.io can be used to introduce real time feature to the system.

Some times there may be situations where users can connect as a group and solve a question. Messaging system with group support can increase the usability of the system. Groups can be created automatically using subscription data. And then if there are signed in users inside that group then a messaging session can be started. Since the group made with users who has a good knowledge on that field it will produce good solutions.

##### References

|  |  |
| --- | --- |
| [1] | Google Inc., "Angular," Google Inc., [Online]. Available: https://angular.io/. [Accessed 5 May 2017]. |
| [2] | "Chai.js," [Online]. Available: http://chaijs.com/. [Accessed 5 May 2017]. |
| [3] | Node.js Foundation, "Express.js," Node.js Foundation, [Online]. Available: https://expressjs.com/. [Accessed 5 May 2017]. |
| [4] | "Introducing JSON," [Online]. Available: http://www.json.org/. [Accessed 12 May 2017]. |
| [5] | PivotalLabs, "Jasmine Documentation," PivotalLabs, [Online]. Available: https://jasmine.github.io/. [Accessed 12 May 2017]. |
| [6] | Auth0, "JSON web tokens," Auth0, [Online]. Available: https://jwt.io/. [Accessed 12 May 2017]. |
| [7] | Google Inc., "Karma - Spectacular test runner for javascript," Google Inc., [Online]. Available: https://karma-runner.github.io/1.0/index.html. [Accessed 5 May 2017]. |
| [8] | Mocha.js Org, "Mocha," Mocha.js Org, [Online]. Available: https://mochajs.org/. [Accessed 5 May 2017]. |
| [9] | Node Foundation, "Node.js," Node.js Foundation, [Online]. Available: https://nodejs.org/en/about/. [Accessed 5 May 2017]. |
| [10] | "Protractor, End to end testing for angular.js," [Online]. Available: http://www.protractortest.org/#/. [Accessed 5 May 2017]. |
| [11] | Selenium HQ, "Selenium IDE - Plugins," Selenium HQ, [Online]. Available: http://www.seleniumhq.org/projects/ide/. [Accessed 5 May 2017]. |
| [12] | MongoDB, Inc, "What is mongoDB," MongoDB, Inc, [Online]. Available: https://www.mongodb.com/what-is-mongodb. [Accessed 5 May 2017]. |
| [13] | D. K. Barry, "Service Architecture," [Online]. Available: http://www.service-architecture.com/articles/web-services/representational\_state\_transfer\_rest.html. [Accessed 12 May 2017]. |
| [14] | StackExchange Inc., "StackOverflow," StackExchange Inc., [Online]. Available: https://stackoverflow.com/. [Accessed 25 May 2017]. |
| [15] | Quora, "Quora," Quora, [Online]. Available: https://www.quora.com/. [Accessed 25 May 2017]. |
| [16] | "Moodle Open source learning platform," 21 April 2017. [Online]. Available: https://docs.moodle.org/33/en/Main\_page. [Accessed 25 May 2017]. |