Project Design Document

For

Spartan United

Smart and Connected Educational Community

SAN JOSÉ STATE UNIVERSITY

CMPE 281: Cloud Technologies

SPRING 2017

Submitted to:

Dr. Jerry Gao

Submitted by:

Group 19

Team member	Student ID
Aayushi Mittal	010953215
Aditya Pradip Parashar	011440195
Rucha Marathe	011418511
Sohrab Ali	011423958

ABSTRACT

SpartanUnited – A Educational Social network and its various Design Workflow Defined

By

Aayushi Mittal, Aditya Pradip Parashar, Rucha Marathe and Sohrab Ali

Social media is a biggest successful buzzword used in the recent time. Its success opened various opportunities for the developers. With the rapid growth of social platforms such as Facebook, bebo, and LinkedIn, social networks have become an important part of our daily life. Social networks have established the communication structures and social behaviors based on various factors, such as genders, ages, friendship, common interests, and career backgrounds.

Educational system such as schools and universities have also discovered the potential to improve the public and private communication, information sharing, and outreach. Today, the existing Universities and schools does not have their own social network community and faces many challenges, such as untenable connectivity and permissions, manual and error-prone defined workflows. Moreover, the software systems are inflexible to adapt to change, and insecure in the way information is shared, which usually become the potential target for hackers. In this project, we propose the Design document for educational system, which can create and maintaining multi-layered social network topology.

The approach enables a feasible hierarchy of connectivity, permissions, and services; enhances security and flexibility in managing workflows among educational system social groups and departments. The web app presents fully integrated workflow dashboard, which provides a friendly end-to-end user experience while ensuring security, regulating information sharing, and enhancing the overall system monitoring

Developing any application requires storage of large data into databases. Many databases are available for the developers, Choosing the right one make development easier. MongoDB is a cross platform document oriented, schema-less database eschewed the traditional table based relational database structure in favor of JSON like documents.

***https://arxiv.org/ftp/arxiv/papers/1503/1503.06548.pdf

Table of Contents

List of Figures

Chapter 1. Project Overview

1.1 Introduction – Project Goals and objectives

With the growth of social media platform and online community, social network has become new tools for public sector to communicate publicly and internally, to improve the transparency, participation, and collaboration. Education system has promoted the new provision of learning, where workflow and process could be integrated and executed securely and seamlessly. There is also increasing needs to secure the student data and its personal information.

Users building explicit networks that represent their social relationships and often share a wealth of personal information to their own benefit. The potential privacy risks of such behavior are often underestimated or ignored. The problem is exacerbated by lacking experience and awareness in users, as well as poorly designed tools for privacy-management on the part of the system.

Spartan United aims to create a flexible social platform for engaging and managing the department and students of schools and university. The platform offers web solution to configure and manage educational system for communication, authorize explicit and implicit permission on significant process, and notify on changes between process in real-time.

1.2 Problems and Motivations

Currently, there aren't many existing Educational connected community platforms that could support relationship and workflow between various departments and its students. Support for scalability and security in current Educational system is another prone area.

The inspiration of this venture is to give an easy way to use interface and adaptable arrangement and framework to deal communication system for the students with secure quality. Verifiable information ought to be put away with important measurements for future review and research SpartanUnited with various nodes.

Chapter 2: Project Architecture

2.1 Social Network Model

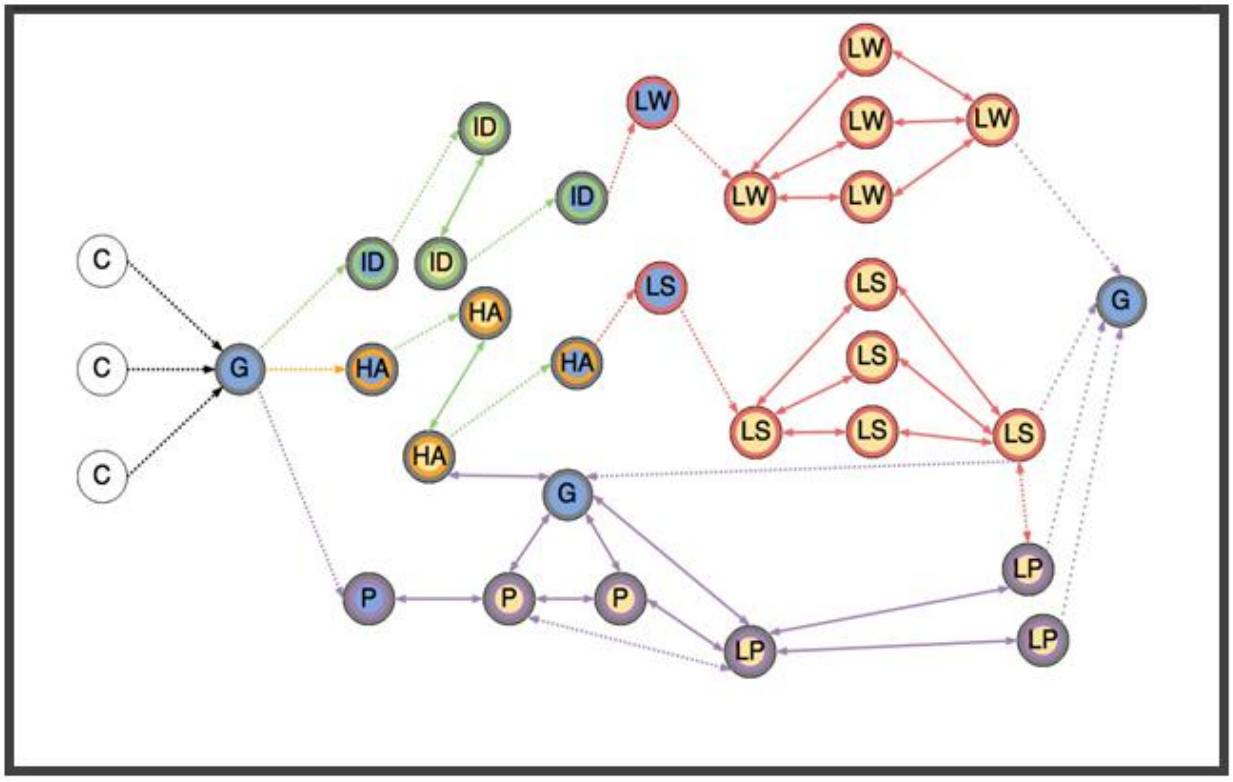

Figure 1: Graph model design concept

- *Nodes*: We are representing three different kinds of nodes in our project namely
  1. Smart nodes
  2. Client nodes
  3. Service nodes which represent members and department in educational social network.
- *Links*: links describe relationship between 3 different nodes, based on the relationship type, services, or functionalities.
- *Clusters*: Clusters defined as the relationship group that every node could be able to communicate to each other internally.
- *Graphs*: Visual display of group of nodes and their relationships in the cluster.

2.2 System Architecture Design

Client will send the http request using front end, build using angular js and html5. The request will be forwarded to the express web app server. The server will forward the request to the service node using messaging queue to service node, which in turn will run the CRUD operations to fulfill the request. Redis is used to cache the most used data and facilitate extremely fast retrieval of the information and passport js will be used to define local and third party authentication strategies.

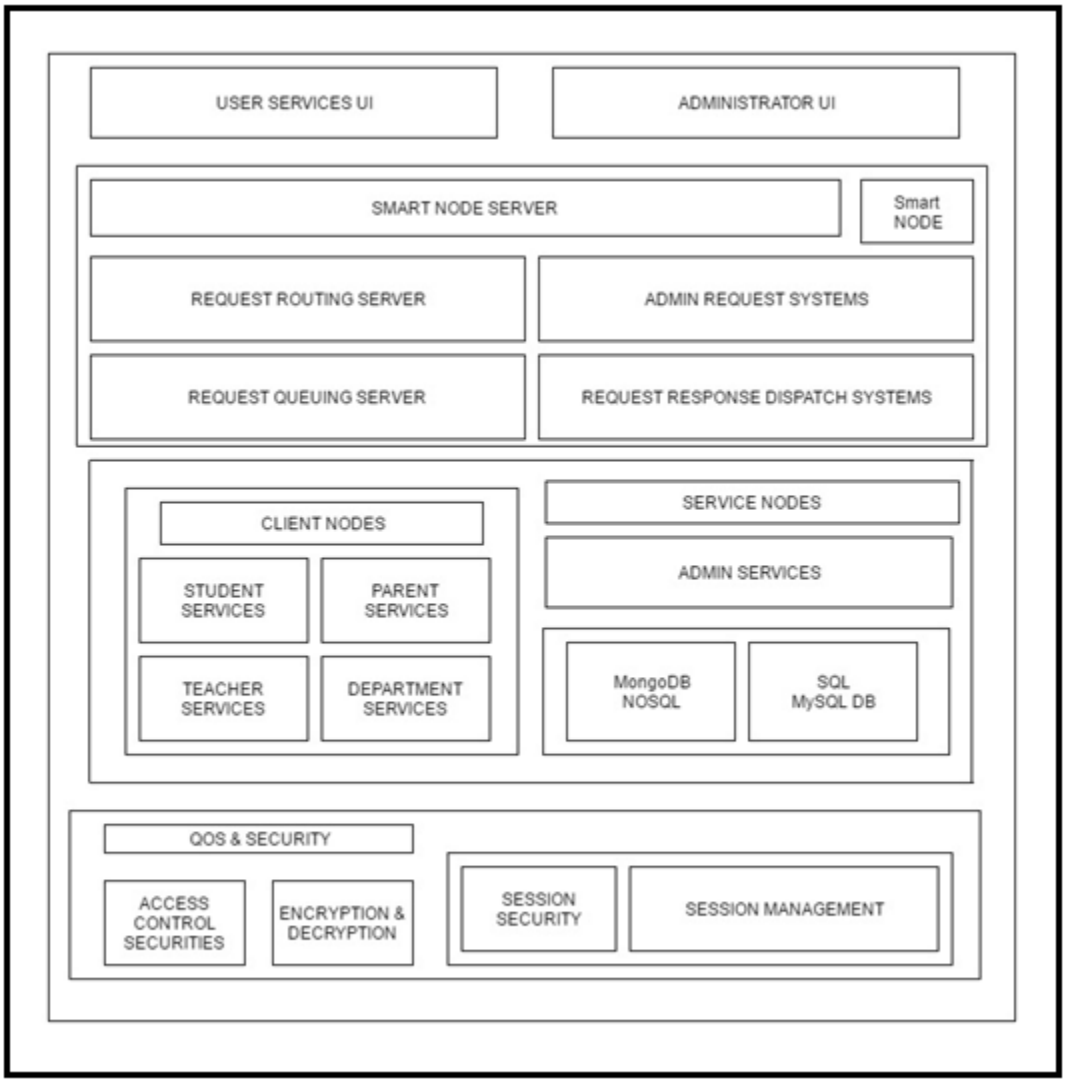

Figure 2: System Architecture

2.3 System Functional architecture

The architecture diagram is shown below. The design consists of two AWS EC2 instances for two Application servers which are deployed on the cloud. The API's are designed using node Js. When data is transferred over the internet, based on the load on servers, AWS provides the service which helps in load balancing. There are two instances, one for the user and the other for admin. There is successful message and data transfer between the two clients.

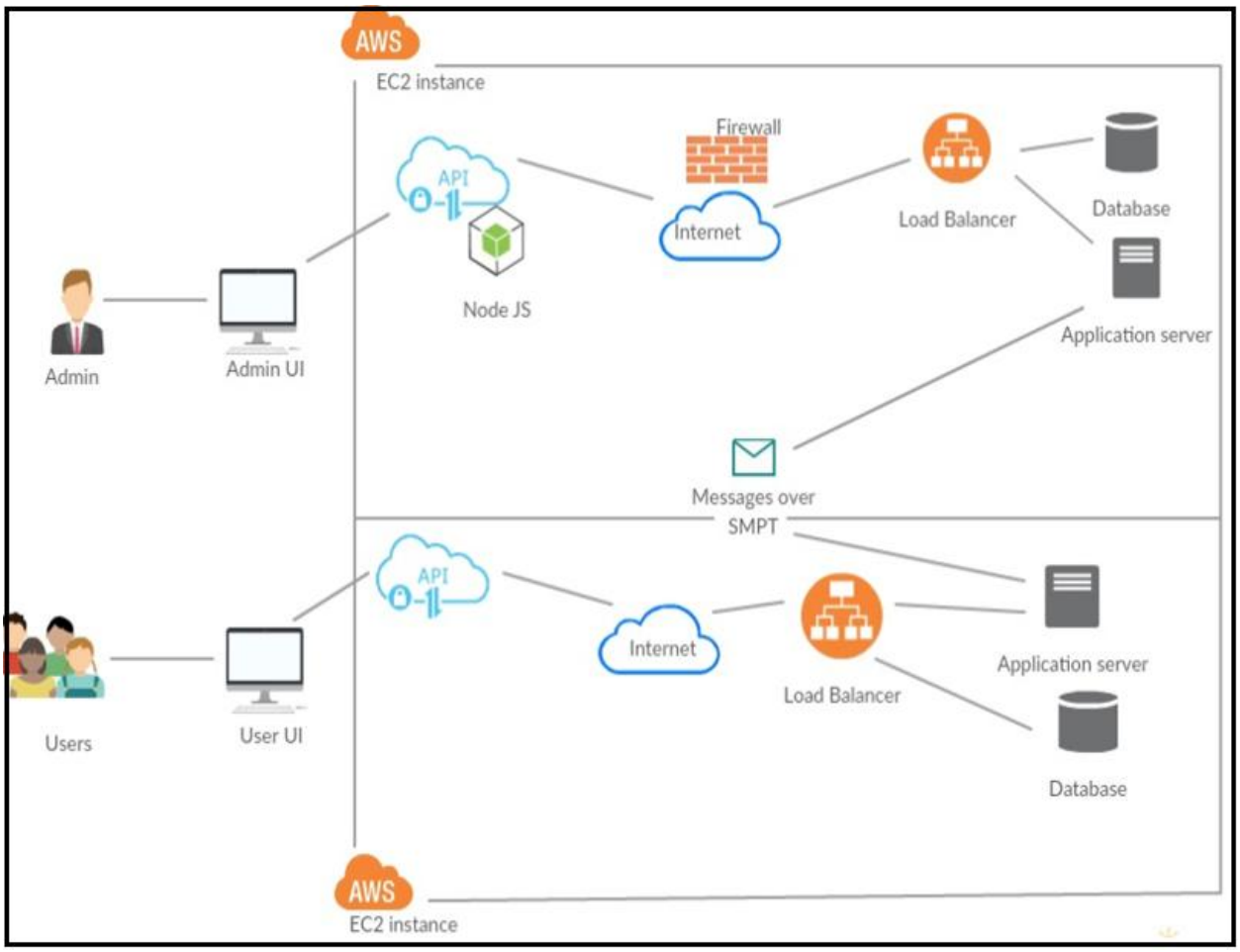

Figure 3: System Functional Architecture

2.4 Components Architecture

The nodes included in the project along with their functionality are described below.

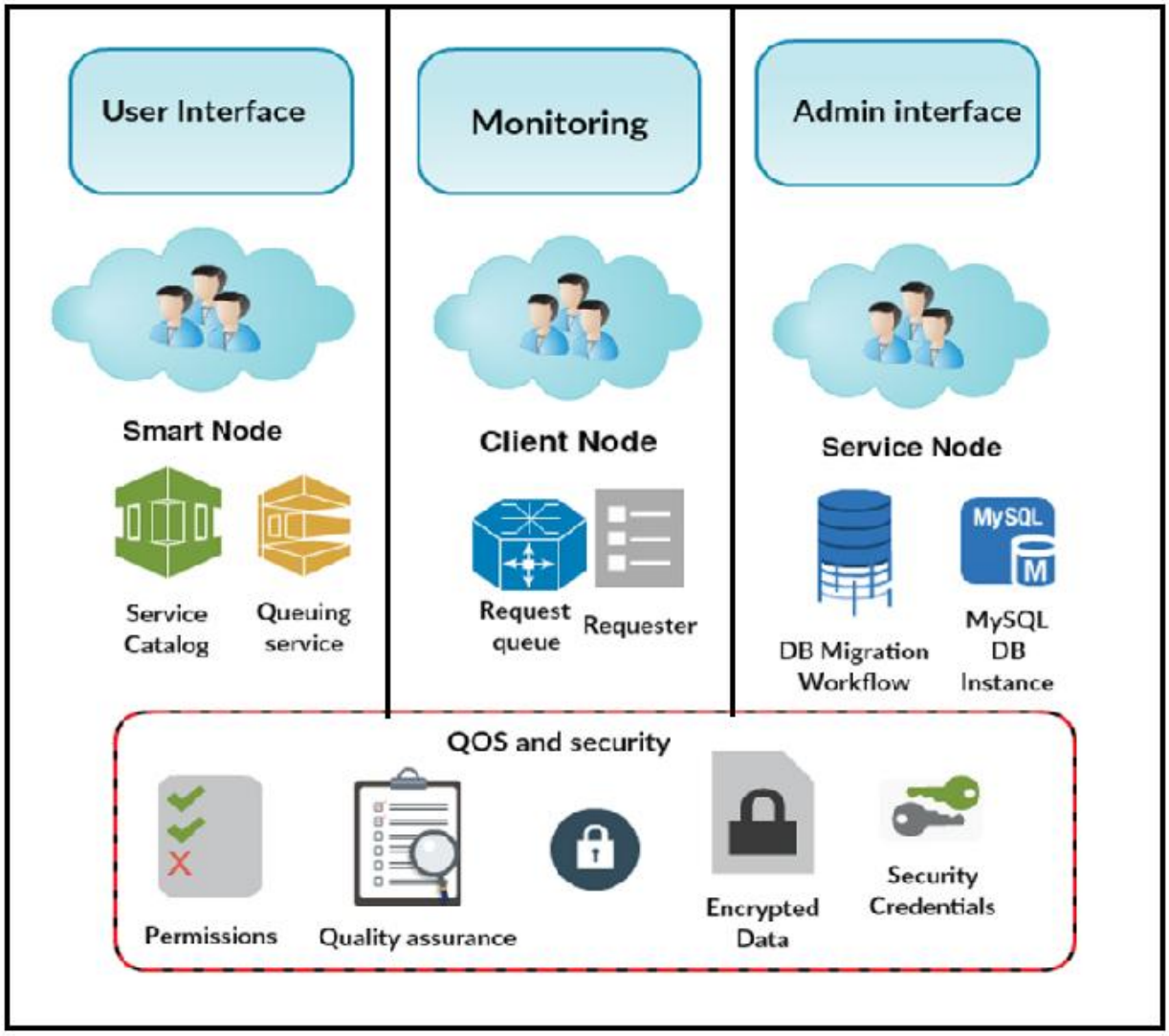

Figure 4: System component architecture

Client Node:

1. Jquery: We will use Jquery for dorm manipulation
2. CSS: We will use CSS to make our frontend responsive, user friendly and to make it compatible with displays with different resolution for all user display services.
3. HTML: We will use HTML to make our frontend.
4. AngularJs: We will use this technology to make our frontend dynamic, extraordinary expressive and quick to develop.

Smart Node:

1. ExpressJs: We will use this technology as it is a light weight web application development framework which is used to fit your application into MV* Pattern.

2. Redis : It is in memory no sql key-value data store which will facilitate extreme fast retrieval of frequently accessed data.
3. RabbitMQ : RabbitMQ is a open source message broker software which we are using to introduce fault tolerance in our system.
4. NodeJs: We are using NodeJS to create restfull services in our project.
5. PassportJs: We are using PassportJs for authentication engine.

Service Node:

1. NodeJs Server: It will listen to the queue, process the request and post response on a reply queue.
2. MongoDB : To store the application data.

Chapter 3. Technologies Descriptions

Technology Used

Figure 5: Technologies used

3.1 Client technologies

Components	Usage	Technology/Tools
Web Server	Application Server	Node.js
	Front End	HTML5, CSS3, Angular.js, Bootstrap
Database	Persistence	MongoDB, SQL
	Caching	Redis
Message Passing	Message Queuing, Fault Tolerance, Communication	RabbitMQ
Deployment	Security, Scalability	Amazon EC2
Middleware	Web Application framework	Express.js

Table 1: Client technologies

3.2 CI/CD technologies

Category	Name	Description
Cloud	AWS • ECS • RDBS • EC2 • ElastiCache	Amazon Web Services for building cloud infrastructure (deploy application, host DB and API services)
Repository	Git	Open source sub-version repository.

Table 2: CI/CD technologies

Chapter 4. Project Design

The Educational SpartanUnited community system supports the following roles:

a) ***Admin***: System admin has full access to the system. Admin has the right to configure or edit workflow, task, and other users' access.

b) ***Department***: This role gives user full access to control department, such as adding/editing users in department, configure department information, monitor overall tasks flow in department.

c) ***Students/ teachers/parents***: Users who submit/view information in the SpartanUnited network.

The functional use case diagram is as shown below. The various functionalities as listed above are made available to the users.

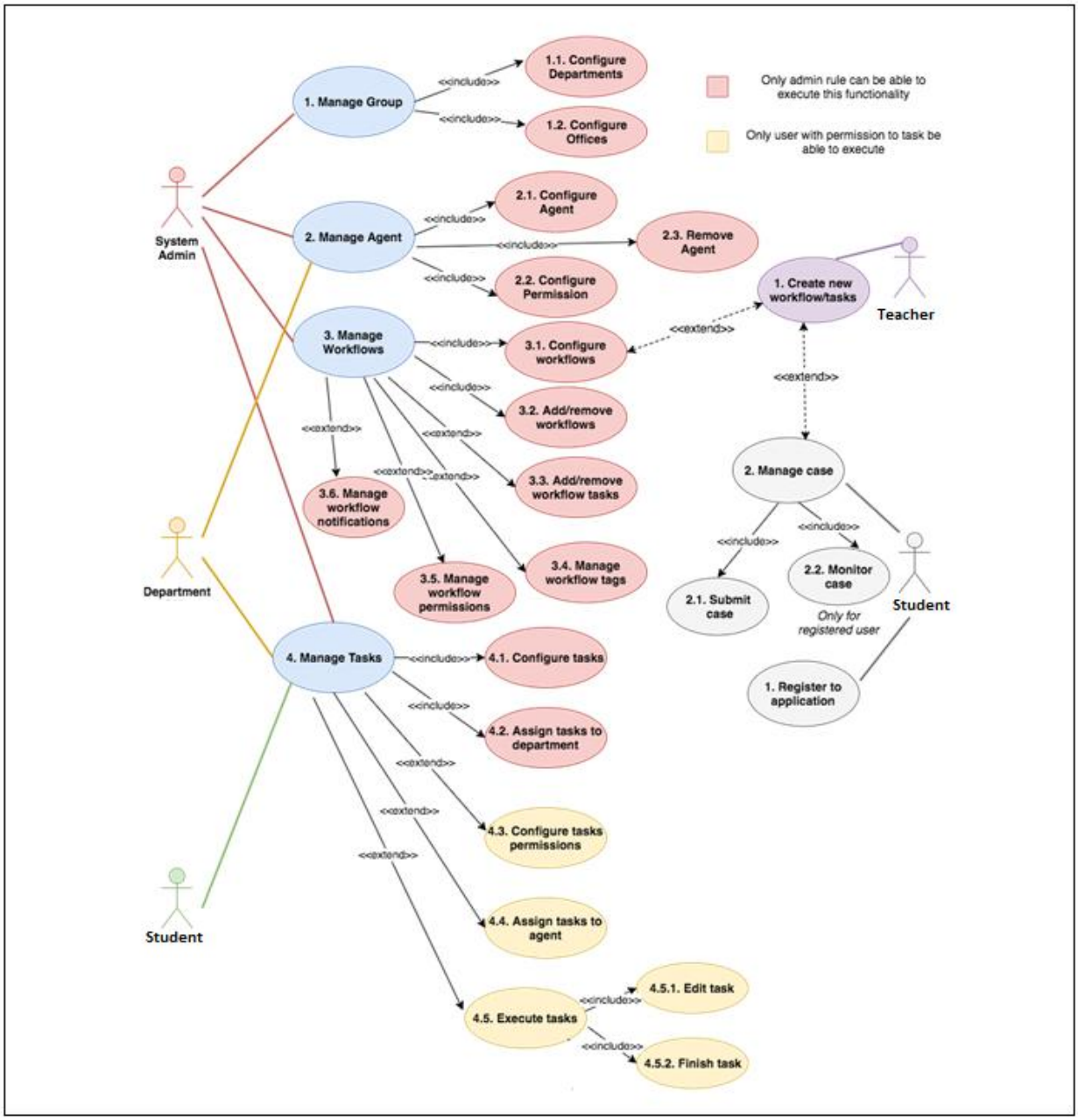

Figure 6: System Functional Use Cases

4.1 Client-Tier design

4.1.1 Web UI Design

This is the front page of the web application

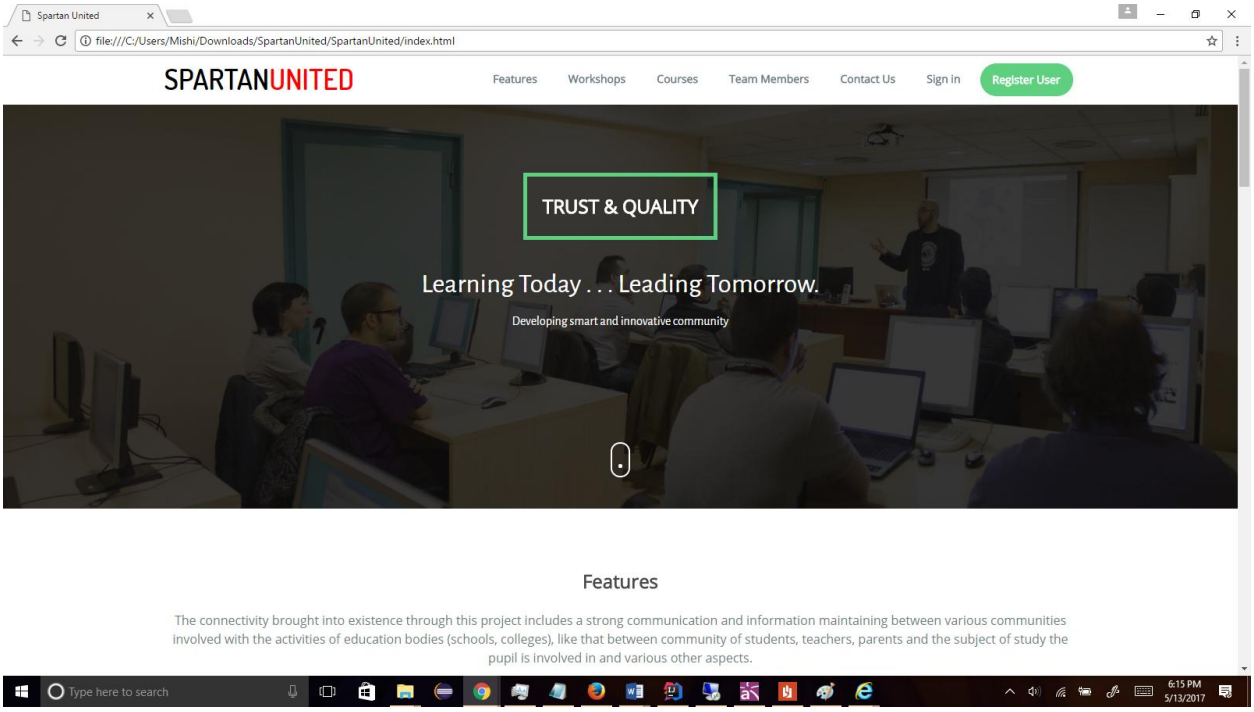

Figure 7: Web UI First page

4.1.2 Sign UP page

The Sign Up page appears as follows:

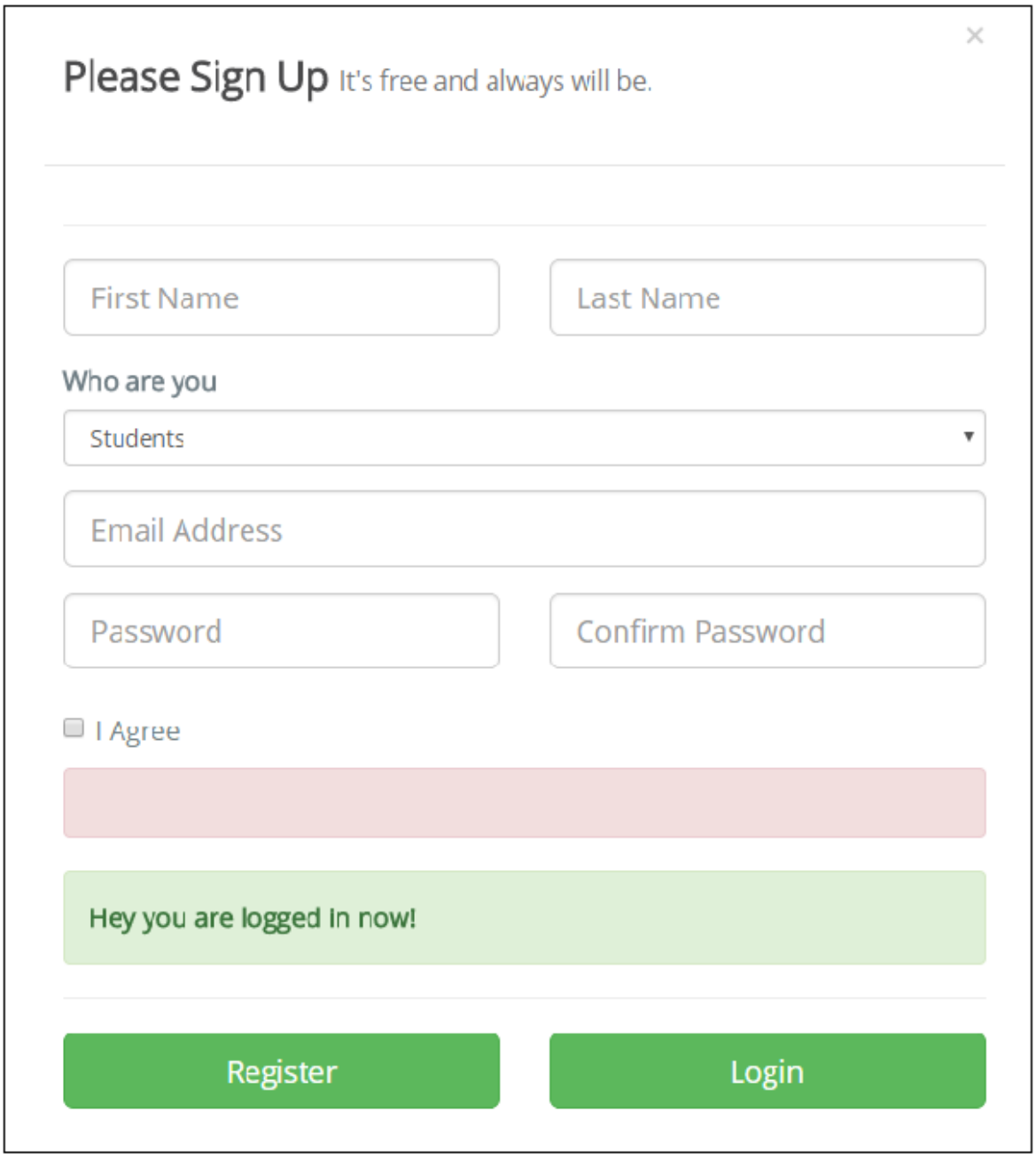

Figure 8: Web Registration – Sign Up module

4.1.3 Login page

The login page appears as shown below:

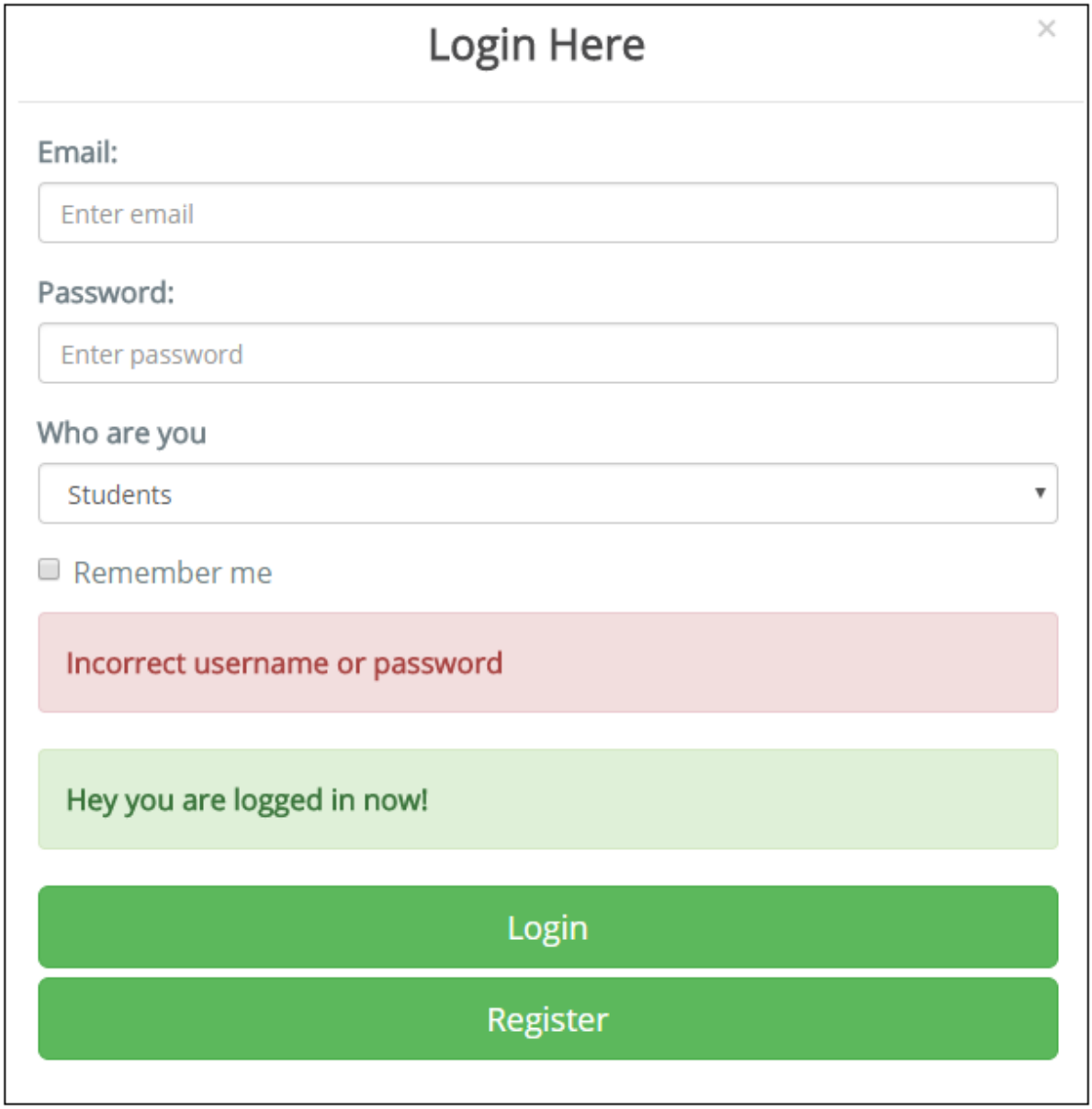

Figure 9: Web Registration –Login module

4.1.4 Client Select page

One can access the required client page through here directly.

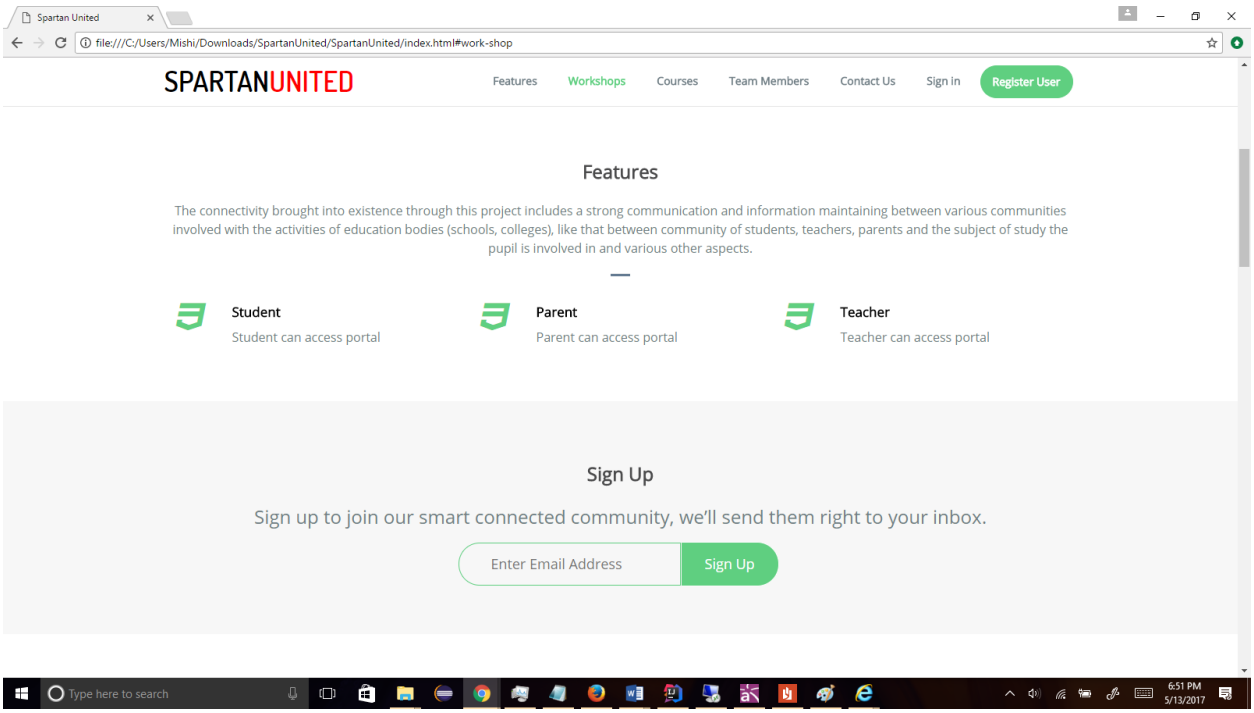

Figure 10: Client select page

4.1.5 Announcement Page

This page displays the upcoming announcement page.

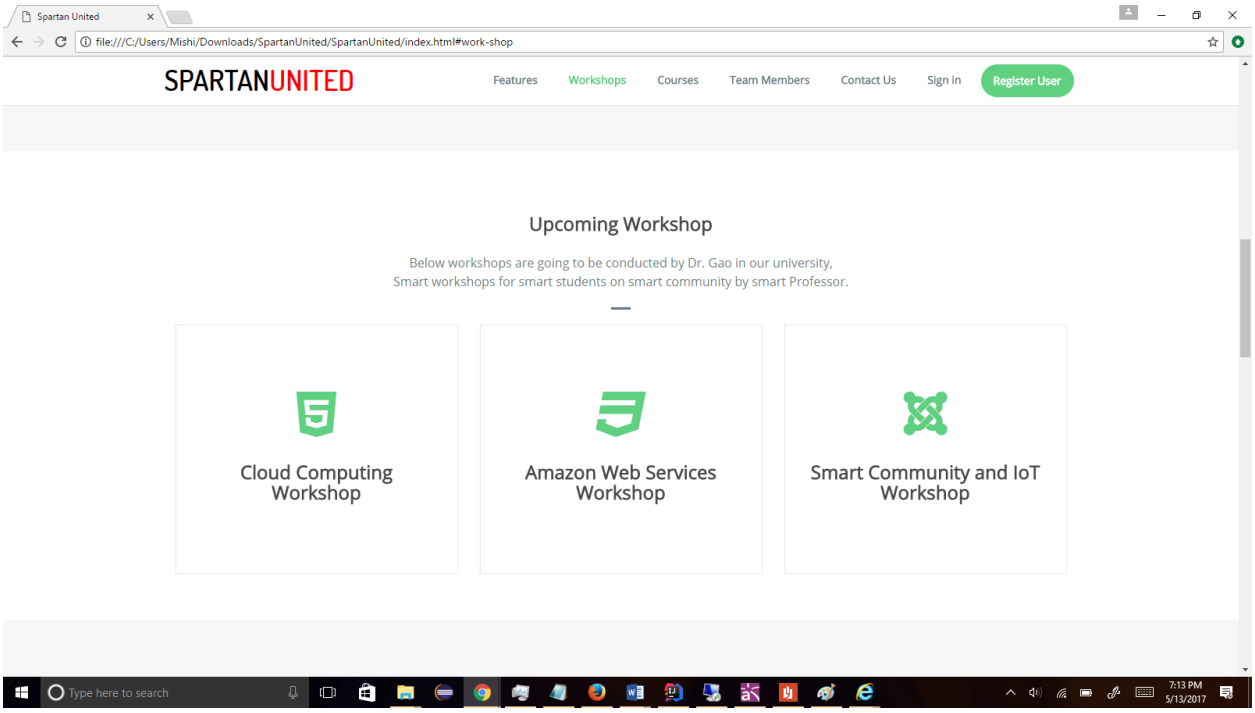

Figure 11: Upcoming Announcement

4.1.6 Courses select page

This page appears as follows:

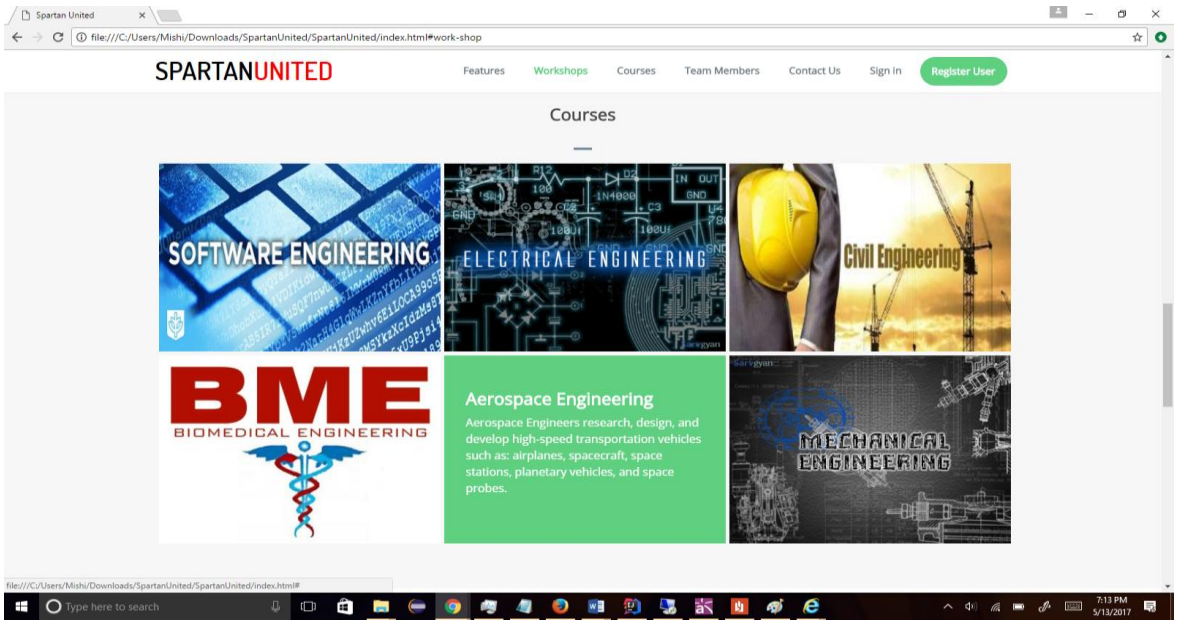

Figure 12: Courses displayed Page

4.1.6 Contact page

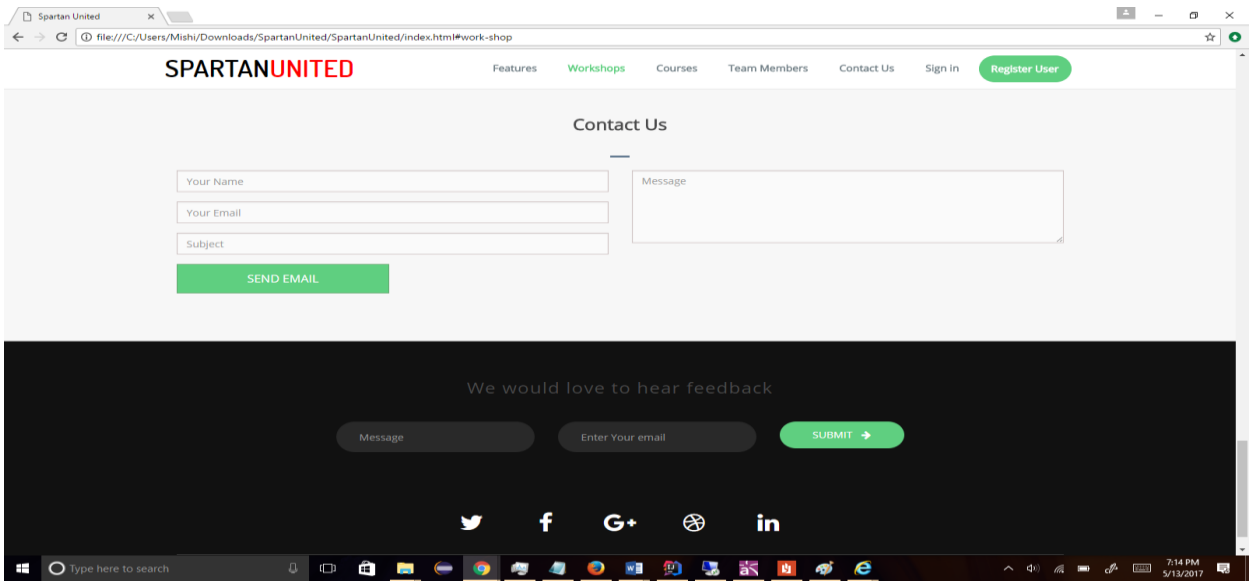

Figure 13: Contact Page

Chapter 5. Application Scalability

In this project, we use AWS ECS and set thresholds for each of these parameters depending upon usage patterns. If the thresholds are surpassed, the system will trigger the dynamic creation or deletion of instances to efficiently scale out and scale in respectively.

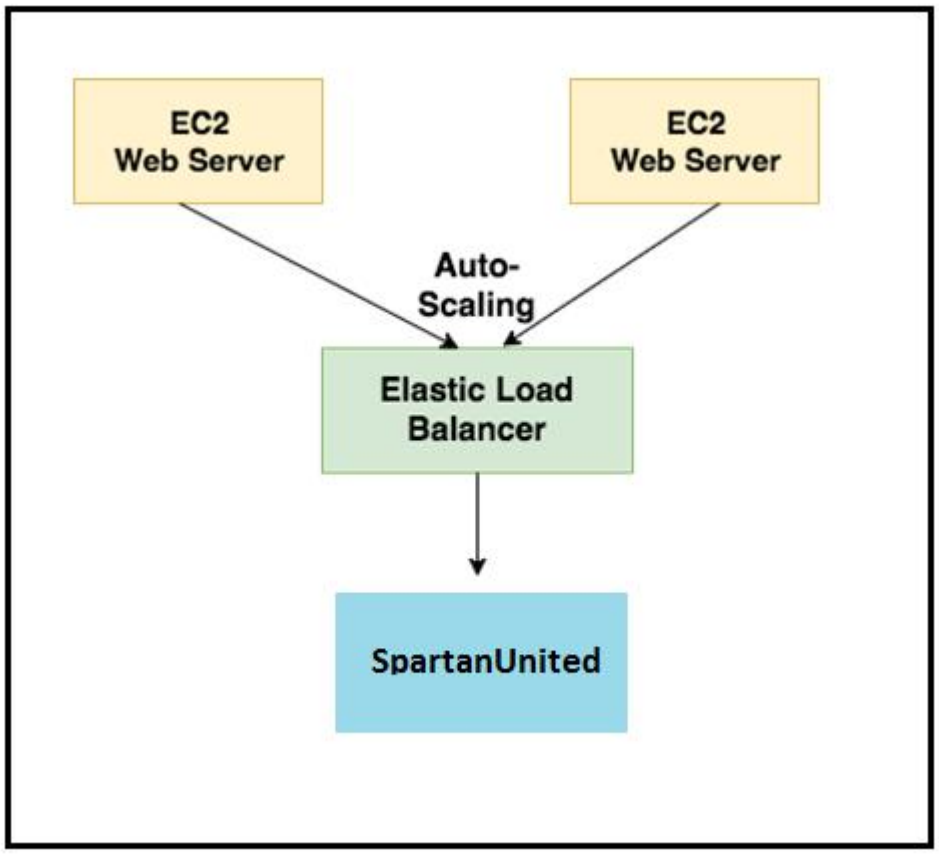

Figure 14: Scalability Algorithm

Below is the diagram which demonstrates the AWS Auto Scaling cool-down architecture:

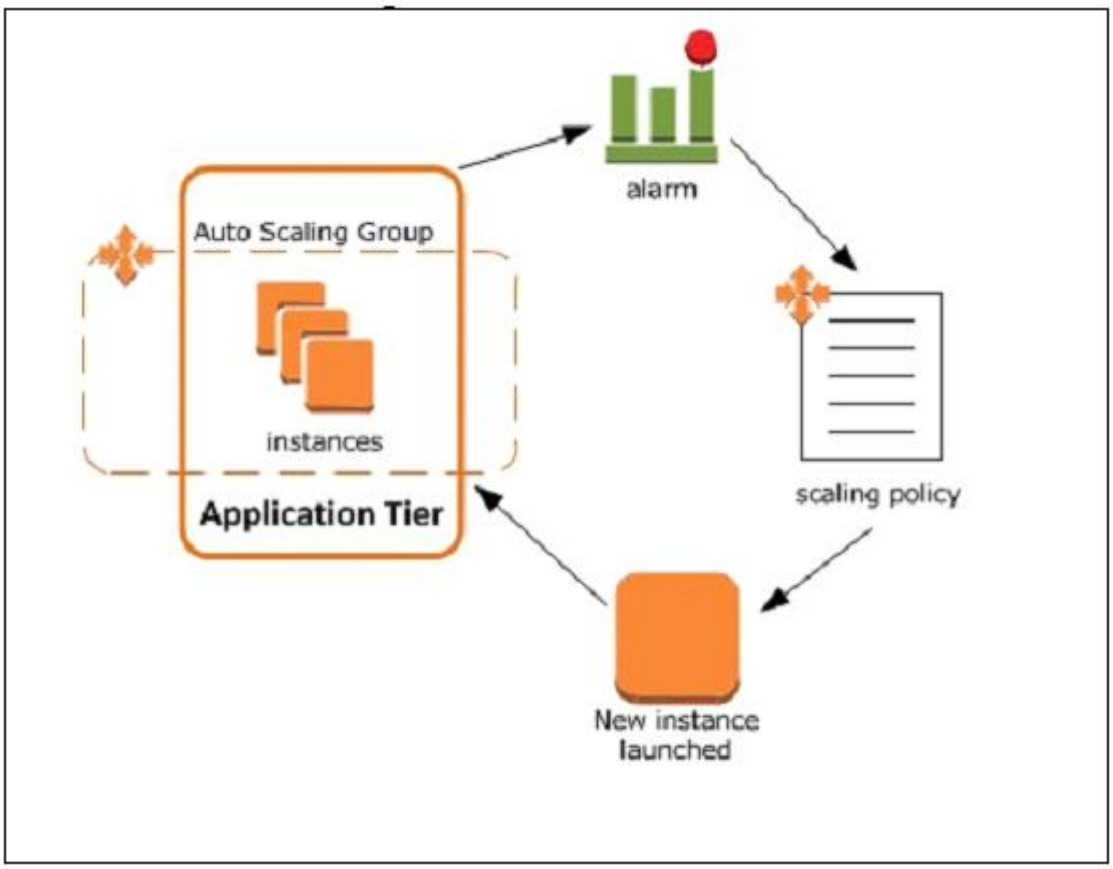

Figure 15: AWS Scaling

(*Referenced from*: http://docs.aws.amazon.com/autoscaling/latest/userguide/Cooldown.html)

Chapter 6. Summary, Conclusions, Recommendations

6.1 Summary

The venture is planned to build up an educational organization stage where students can utilize it to lead and process chats, issues, and request. The consent and security level ought to be characterized certainly and expressly. The work process to deal with a case ought to be obviously characterized, self-driving and traceable.

6.2 Conclusions

Keeping in mind the end goal to give an easy to use interface and adaptable arrangement for SpartanUnited administration and bolster dynamical work process order, the accompanying things are conveyed with the fulfillment of the venture:

● Flexible DB mapping and RESTful API to deal with interpersonal organization progressive system setup.

● Web UI to show client connections among various department.

● Desktop and versatile inviting administrator dashboard web applications to deal with association, bunch administrations and work process setups.

Each colleague has picked up a ton of encounters amid this project. In the first place we led a writing research about informal organization and work process administrations, which help us to construct the essential comprehension about the unique circumstance and reason for this project. At that point we concentrated a few structure and select from them what we think could be utilized to develop our framework, and we took in a considerable measure in this procedure, for example, Neo4j and angularjs.

6.3 System design properties

From the framework configuration perspective, the framework ought to be improved to have the accompanying properties:

●Scalability: The framework ought to have the capacity to ingest new equipment assets rapidly proportional on a level plane when important. In the meantime from the project perspective, it ought to be simple and fast for the framework to incorporate or associate with new associations.

●Availability: Smart load adjusting calculation ought to be connected to guarantee that the reaction time of the framework is as little as could be expected under the circumstances and the server ought to stay online every minute of every day.

●Redundancy: The delicate information ought to be duplicated and kept up in various machines to evade single purpose of disappointment.

Glossary

1. *Load Balancer*: Improves the distribution of workloads across multiple computing resources, such as computers, a computer cluster, network links, central processing units, or disk drives. Load balancing aims to optimize resource use, maximize throughput, minimize response time, and avoid overload of any single resource. Using multiple components with load balancing instead of a single component may increase reliability and availability through redundancy. Load balancing usually involves dedicated software or hardware, such as a multilayer switch or a Domain Name System server process. *[Wikipedia]*
2. *Unit testing*: The software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation.
3. *End-to-end testing*: The technique used to test whether the flow of an application right from start to finish is behaving as expected. The purpose of performing end-to-end testing is to identify system dependencies and to ensure that the data integrity is maintained between various system components and systems.
4. *Benchmark testing*: The process of load testing a component or an entire end to end IT system to determine the performance characteristics of the application. The benchmark test is repeatable in that the performance measurements captured will vary only a few percent each time the test is run. This enables single changes to be made to the application or infrastructure in an attempt to determine if there is a performance improvement or degradation.

**http://doc.utwente.nl/81270/1/Beye12privacy.pdf

References

1. Danah M. Boyd, Nicole B. Ellison (2007 Oct). Social Network Sites: Definition, History and Scholarship [Online]. Retrieved from: http://onlinelibrary.wiley.com/doi/10.1111/j.1083-6101.2007.00393.x/full

2. Pratt CA, Arteaga S, Loria C (2014). Forging a future of better cardiovascular health: addressing childhood obesity.

3. Albert-László Barabási, Zoltán Dezso˝, Erzsébet Ravasz, Soon-Hyung Yook and Zoltán Oltvai (2002, Nov 25). Scale-free and hierarchical structures in complex networks [Online].
Retrieved from: https://www3.nd.edu/~networks/Publication%20Categories/03%20Journal%20Articles/Physics/ScalefreeHierarchical_Sitges%20Proceedings-Complex%20Networks,%20.pdf

4. F. Toriumi, I. Okada , H. Yamamoto , H. Suwa , K. Izumi and Y. Hashimoto (2011). Classification of Social Network Sites based on Network Indexes and Communication Patterns. Retrieved from: http://users.cecs.anu.edu.au/~sguo/swm2011_submission_10.pdf

5. David F. Nettleton (2015 Oct). A personalized privacy tool for online social network users + Generator for synthetic online social network graph data. Retrieved from: https://www.researchgate.net/publication/284900614_A_personalized_privacy_tool_for_online_social_net work_users_Generator_for_synthetic_online_social_network_graph_data

6. Riitta Toivonen, Jukka-Pekka Onnela, Jari Sarama¨ki, Jo¨rkki Hyvo¨nen, Kimmo Kaski (2006, Feb 21). "A Model for Social Networks - Harvard University." [Online].
Retrieved from: https://www.hks.harvard.edu/netgov/files/fellows/onnela_j/papers/A7.pdf

7. Lobna Sameer Hassan, Hany Abdelghaffar (2015, Jun). The use of Social Network in Enhancing eRulemaking.
Retrieved from: https://www.researchgate.net/publication/280554765_The_use_of_Social_Network_in_Enhancing_eRulemaking

8. A B M Moniruzzaman, and Syed Akhter Hossain (2013). NoSQL Database: New Era of Databases for Big data Analytics - Classification, Characteristics and Comparison [Online].
Retrieved from: http://arxiv.org/pdf/1307.0191v1.pdf

9. Aravind Mohan, Mahdi Ebrahimi, Shiyong Lu, and Alexander Kotov. A NoSQL Data Model For Scalable Big Data Workflow Execution [Online].
Retrieved from: http://arxiv.org/pdf/1307.0191v1.pdf

10. Konglong Tang, Yong Wang, Hao Li, Yanxiu Sheng, Xi Wang, and Zhiqiang Wei (2013). Design and Implementation of Push Notification System Based on the MQTT Protocol [Online].
Retrieved from: http://www.atlantis-press.com/php/download_paper.php?id=9566

11. Wei Tan, M. Brian Blake, Iman Saleh, and Schahram Dustdar (2013, Oct). Social-Network-Sourced Big Data Analytics [Online].

Retrieved from: http://s3.amazonaws.com/academia.edu.documents/32663735/Social-NetworkSourced_Big_Data_Analytics.pdf?AWSAccessKeyId=AKIAJ56TQJRTWSMTNPEA&Expires=147521 40 70&Signature=59%2Bugk8n0FJjRqrrGUKgFgPOpbI%3D&response-contentdisposition=inline%3B%20filename%3DSocial-Network-Sourced_Big_Data_Analytic.pdf

12. Michael Beye, Arjan Jeckmans , Zekeriya Erkin, Pieter Hartel, Reginald Lagendijk, and Qiang Tang (2012). Privacy in Online Social Networks [Online].
Retrieved from: http://doc.utwente.nl/81270/1/Beye12privacy.pdf

13. Mason A. Carpenter, Mingxiang Li, and Han Jiang (2012). Social Network Research in Organizational Contexts: A Systematic Review of Methodological Issues and Choices [Online].
Retrieved from: http://jom.sagepub.com/content/38/4/1328.full.pdf+html

14.Katherine Faust. Comparing Social Networks: Size, Density, and Local Structure (2006).
Retrieved from: http://www.stat-d.si/mz/mz3.1/faust.pdf

15. Enterprise Social Network (ESN) Vendor Comparison (2013).
Retrieved from: http://www.dialogueconsulting.com.au/documents/2013/05/report-enterprise-social-networking-platformvendor-comparison.pdf

16. Andry Alamsyah. Big Data Analytics: A Social Network approach (2015).
Retrieved from: https://www.slideshare.net/andrybrewok/big-data-analytics-a-social-network-approach