



CMPE 281: CLOUD TECHNOLOGIES
Fall 2017

Team 11

Team Members:

Ishwarya Varadarajan(SJSU ID: 011549473)

Sowmya Viswanathan(SJSU ID: 011432668)

Table of Contents

1. Introduction	4
1.1. Characteristics of a Smart City.....	4
1.2. Need for Smart City Connected Services	5
1.2.1. Smart city creates efficiency.....	5
1.2.2. Monitor environmental effects.	5
1.2.3. Help/benefit people or community	5
1.2.4. Sustainability	5
1.2.5. Growth in city's business.....	5
1.3. Components of a smart city community service	5
1.4. Goal of the Project	6
1.5. Expected Results	6
2. System Infrastructure Design	7
2.1. System Infrastructure.....	7
2.2. System Architecture	7
2.3. Architectural Components:	8
2.3.1. Amazon EC2 Instances	8
2.3.2. AWS Elastic Load Balancer	8
2.3.3. XAMPP.....	8
2.3.4. MySQL DB.....	8
2.3.5. Notification Service	9
2.3.6. EventUp	9
2.4. Technologies	9
2.5. Application Users and their functions.....	9
2.5.1. Administrators	9
2.5.2. Event Moderators/Managers	9
2.5.3. Students	9
2.6. Functional Components	10
2.6.1. Workflow component	10
2.6.2. Messaging component	12
2.6.3. Notification service component	12

2.6.4.	Dashboard Component	13
2.6.5.	DB Component.....	14
2.7.	Use Case Analysis.....	15
3.	Billing Module/Dashboard	17
4.	Smart Node Module	18
5.	Scalability	19
6.	Multi-Tenancy	20
7.	Benefits.....	20
8.	References:	20

1. Introduction

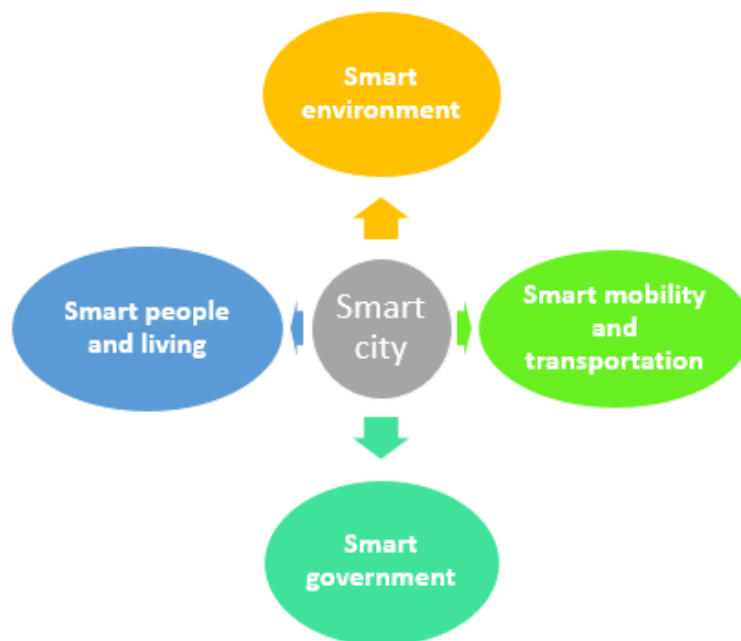
The Internet of Things(IoT) is sweeping across the globe and is engulfing our lives with technology. IoT helps in setting up a Smart City. A smart city is a concept which integrates the different domains in a city like transport, health care, education, communication, agriculture, energy, etc. with information and communication technology to connect people with each other, to utilize the resources efficiently for smoother operations and providing better services to different communities in the city. It gives a clear picture of how a city is evolving.

Information and communication technology (ICT) is used to enhance quality, performance and interactivity of urban services, to reduce costs and resource consumption and to increase contact between citizens and government. [1] Smart City technology is mainly used for real time purposes including notifying the medical care of any accidents that occur immediately, smart management of city's infrastructure, smart connect between people ensuring safety, smart shopping, etc.

This brings us to our application. We have developed a smart city connect service which aims at bringing students from various disciplines together on a community platform by helping them discover academic and cultural events matching their passion by registering to different events. This application will allow event organizers host events, students to register to these events and allow management of the entire platform by an Administrator.

1.1. Characteristics of a Smart City

A smart city brings together technology and community and proves to be of a great value for the people of the community. The communities benefit a lot from the services and operations of a smart city. It provides a sustaining environment to the community people. Some of the characteristics of a smart city is given below:



1.2. Need for Smart City Connected Services

A smart city provides many benefits to a community if constructed properly. Some of the benefits that are derived from a smart city are:

1.2.1. Smart city creates efficiency.

Well-designed technology can provide many services and operations to the community. It can help in reducing redundancies and can streamline responsibilities and tasks. For example, the 'clipper cards' that people have in San Jose for travelling in VTA. It provides easy payment of money for utilization of city's transport. With a swipe of card on the card reader, a wireless transaction deducts the correct amount from the card. The only responsibility of the person carrying the card is to top it up whenever necessary. Transit users in this case, benefit from the quick and convenient payment method.

1.2.2. Monitor environmental effects.

Energy use can be monitored, and real-time updates can prove to be beneficial to the environment and the communities. Sensors and cameras can be set up around any city to capture the air quality or the water quality or monitor traffic, thereby which providing sufficient analysis and data which can help save environmental degradation and help improving quality of life.

1.2.3. Help/benefit people or community

Largest potential benefit from smart cities will be enhancing people's quality of lives. The opportunities cover a broad range of issues, including housing and transportation, happiness and optimism, educational services, environmental conditions and community relationships. [2] Communities will tend to be more strongly connected. Resulting in a safe neighborhood for its people. The people will be well informed because of smart communication resulting in increase in safety.

1.2.4. Sustainability

Smart cities can optimize resource usage like water, fuel, energy and even waste. They help sustain the environment and in saving huge chunks of money. For instance, Los Angeles' LED bulbs save the city \$8 million per year.[3]

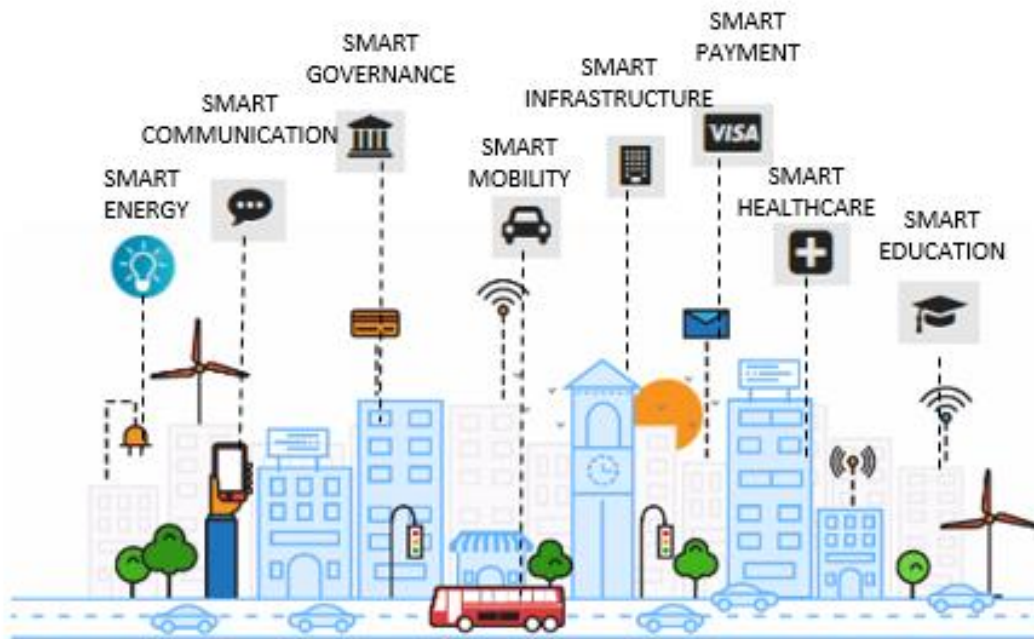
1.2.5. Growth in city's business

Smart cities, not only help in growth and improve the lives of the current residents, but also are a source of attraction for new residents. Everyone will prefer and opt for a cleaner environment which will have many amenities like efficient transportation, efficient communication services, better retail and shopping services. Businesses in such cities will bloom and one of the main reasons for it will be reduced operating costs.

1.3. Components of a smart city community service

With tremendous growth in economy, other challenges pop up alongside like population explosion, environmental degradation, increase in crime rate, increase in levels of pollution, high

demand in infrastructure, etc. Solutions to these challenges can be obtained from Internet of Things which leads to making a city or a community smarter. It helps in connecting different communities. Smart cities include the below mentioned components:



1.4. Goal of the Project

Our project deals with building a service for a smart connected city using an open social network platform. Our project aims in using cloud-based system infrastructure and an open source social community platform to build an application for smart student community services. Our application targets the Education domain of a community and helps in setting up a Smart Education Service. We have developed an application called EventUp, built on an open social network platform – XOOPS, which brings students from various disciplines together on a community platform by helping them discover academic and cultural events matching their passion. This platform will enable creation of student community groups by users/students, creation and management of event pages by event organizers and management of the entire platform by an Administrator.

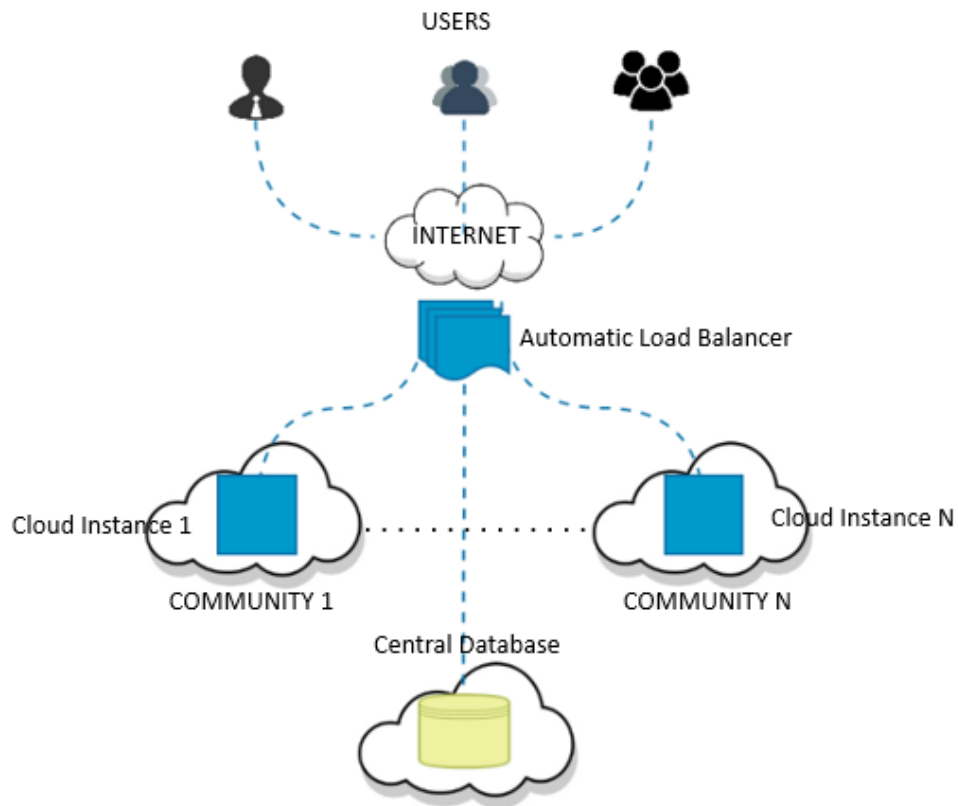
1.5. Expected Results

We hope to develop and deploy a fully functional application built on an open source social network platform for smart community services. The application will be deployed and managed in a cloud environment.

2. System Infrastructure Design

2.1. System Infrastructure

Our application is a web based application built on an open social network platform. It will be hosted on cloud. The users will be able to access it via Internet. Application's basic infrastructure design is shown below:



2.2. System Architecture

The system architecture diagram is shown below. The different components are explained in further section.

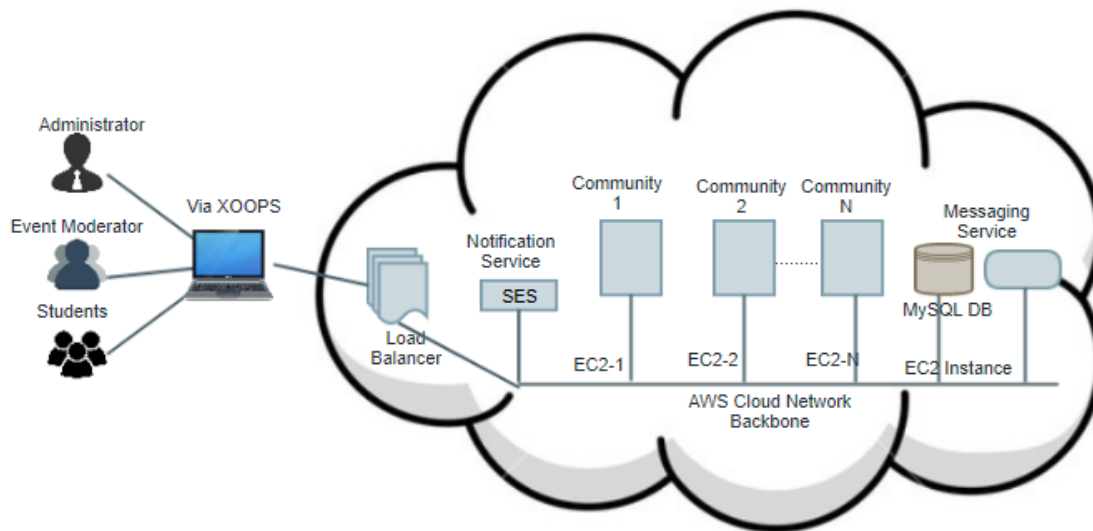
The application EventUp makes use of the cloud infrastructure of AWS for hosting.

Below are the various services of AWS being used to successfully implement this smart community event management application

Each AWS EC2 instance hosts a community connected with a centralized MySQL Database to provide high availability and scalability as the number of users increase.

- The application is supported by a load balancer which diverts traffic to different EC2 instances based on the load

- MySQL Database is hosted on a separate EC2 instance to be central to the entire application and to achieve high data integrity
- AWS's Simple Email Services (SES) is used to provide email notification services to the users registered to this application



2.3. Architectural Components:

2.3.1. Amazon EC2 Instances

Our application will be deployed on Amazon EC2 instances with which it can be accessed remotely. To support scalability, each community will be hosted on one instance.

2.3.2. AWS Elastic Load Balancer

It is used to distribute incoming traffic to the web application to available multiple EC2 instances. This leads to the application being available always preventing, any downtime.

2.3.3. XAMPP

It is an open source full stack platform containing Apache HTTP server for hosting the web application, MySQL server and relational database for user and event data management and PhpMyAdmin, GUI for database management operations. Our application will be deployed on XAMPP which will in turn be deployed on the Amazon EC2 instances.

2.3.4. MySQL DB

Our application has a central DB, which is connected to all the Amazon EC2 instances.

2.3.5. Notification Service

Our application uses the Amazon Simple Email Service. It sends email to different community users of any updates in events.

2.3.6. EventUp

It is a social networking application developed using the open source social community platform XOOPS

1. This application is for student communities to interact and collaborate with each other through academic and cultural events
2. The application facilitates creation and management of user profile and user clusters, creation and management of events, registration of events, collaboration of user clusters through an events' forum
3. It also provides an advanced access control layer to grant specific access controls to different types of users based on their roles such as admin, students & event organizers

2.4. Technologies



2.5. Application Users and their functions

2.5.1. Administrators

Administrators of the application have the following functionalities

- Approve Event Updates/requests
- Create user profiles – Event moderators and students

2.5.2. Event Moderators/Managers

Event Moderators using the application have the following functionalities

- Create events
- Delete events
- Update events

2.5.3. Students

Students using the application have the following functionalities

- View Events
- Register in events
- Chat within the community
- Chat between communities

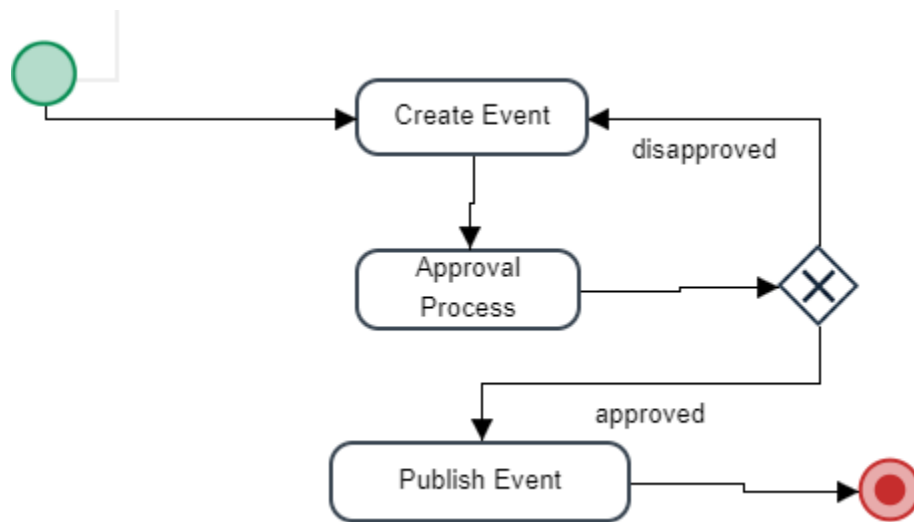
2.6. Functional Components

2.6.1. Workflow component

The workflow for our project has been designed using process maker. The workflow of three main functionalities of our application is given below.

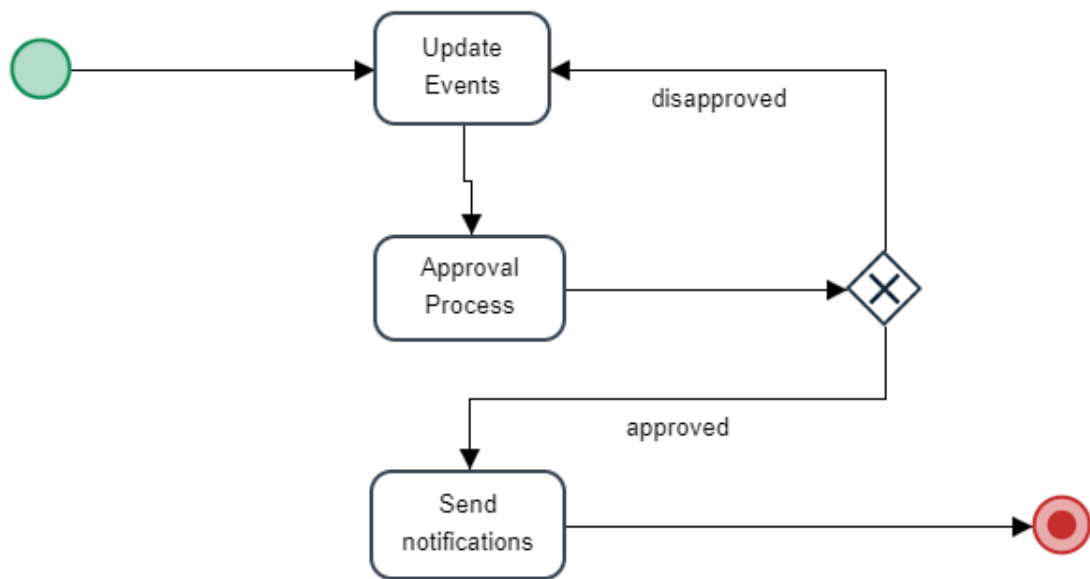
2.6.1.1. Create Events

An event moderator can perform this action. An event is a 'community' in our project. As soon as an event is created, it goes for approval to the Administrator after which a notification is sent to the students through Amazon Simple Email Service. The workflow diagram depicting the flow is shown below.



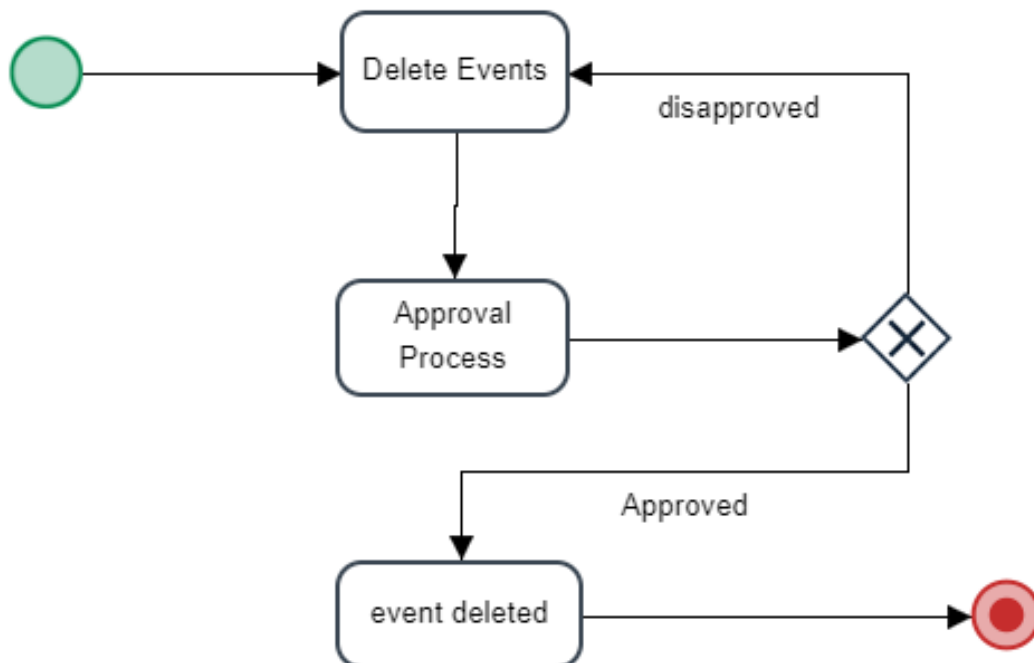
2.6.1.2. Update Events

An event moderator can perform this action. As soon as an event is updated, it goes to the Administrator after which a notification is sent to the students through Amazon Simple Email Service. The workflow diagram depicting the flow is shown below.



2.6.1.3. Delete Events

An event moderator can perform this action. As soon as an event is deleted, it goes to the Administrator after which a notification is sent to the students through Amazon Simple Email Service. The workflow diagram depicting the flow is shown below.



2.6.2. Messaging component

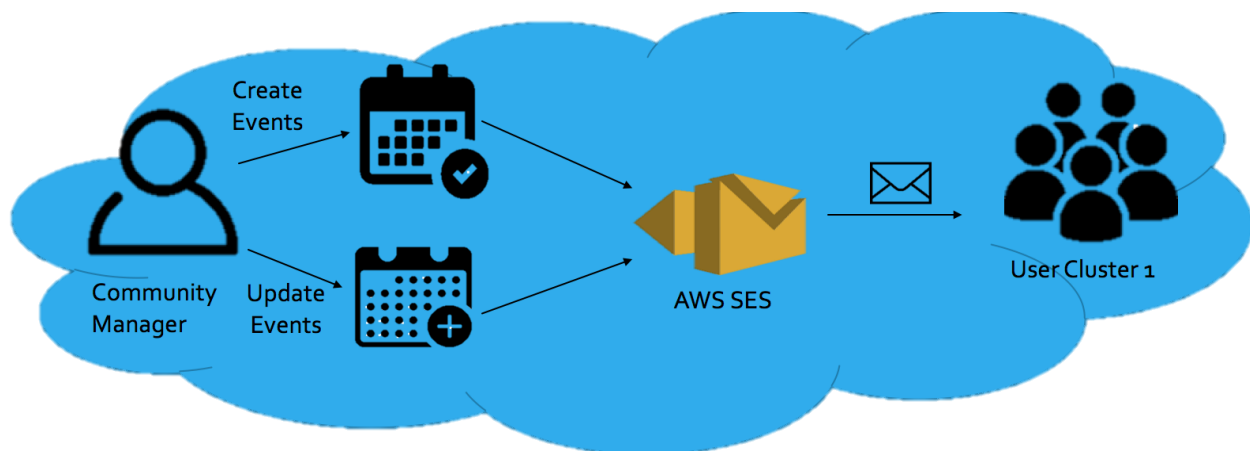
Messaging or communication within a community is important. Our application supports communication within the community and also between two communities through a component called WebChat which uses the socket.io API. The API consists of client and server components. Client module has been written in JavaScript. The server consists of nodejs server module. The server will be hosted on an amazon EC2 instance. Students from communities hosted on other EC2 instances will be able to access and communicate via this nodejs server. This component has several functionalities which are listed below.

- The users in a community, when they log into the chatroom site, will have the ability to communicate with other users of the same community/ or different communities in the same application instance.
- When a user joins the chatroom, a message will be broadcasted to all the users on the chat window.
- As a user, if you post a message, everyone in the chatroom will get message
- If a different user posts a message, it will appear on your screen.
- A client joining the chatroom will be able to see the earlier chat messages. And who all are available in the chatroom.

2.6.3. Notification service component

The project implements the Simple Email Services of AWS to send mail notifications to students who have registered to the application.

- Email notifications are sent to users/students, when an event is created by the Event Organizer and approved by the Administrator.
- Email notifications are sent to users/students, when an already existing event is updated by the Event Organizer and approved by the Administrator.



2.6.4. Dashboard Component

Dashboards are provided for the Admin and the Event Organizers to have a graphical representation of the data as below

Admin's Dashboard:

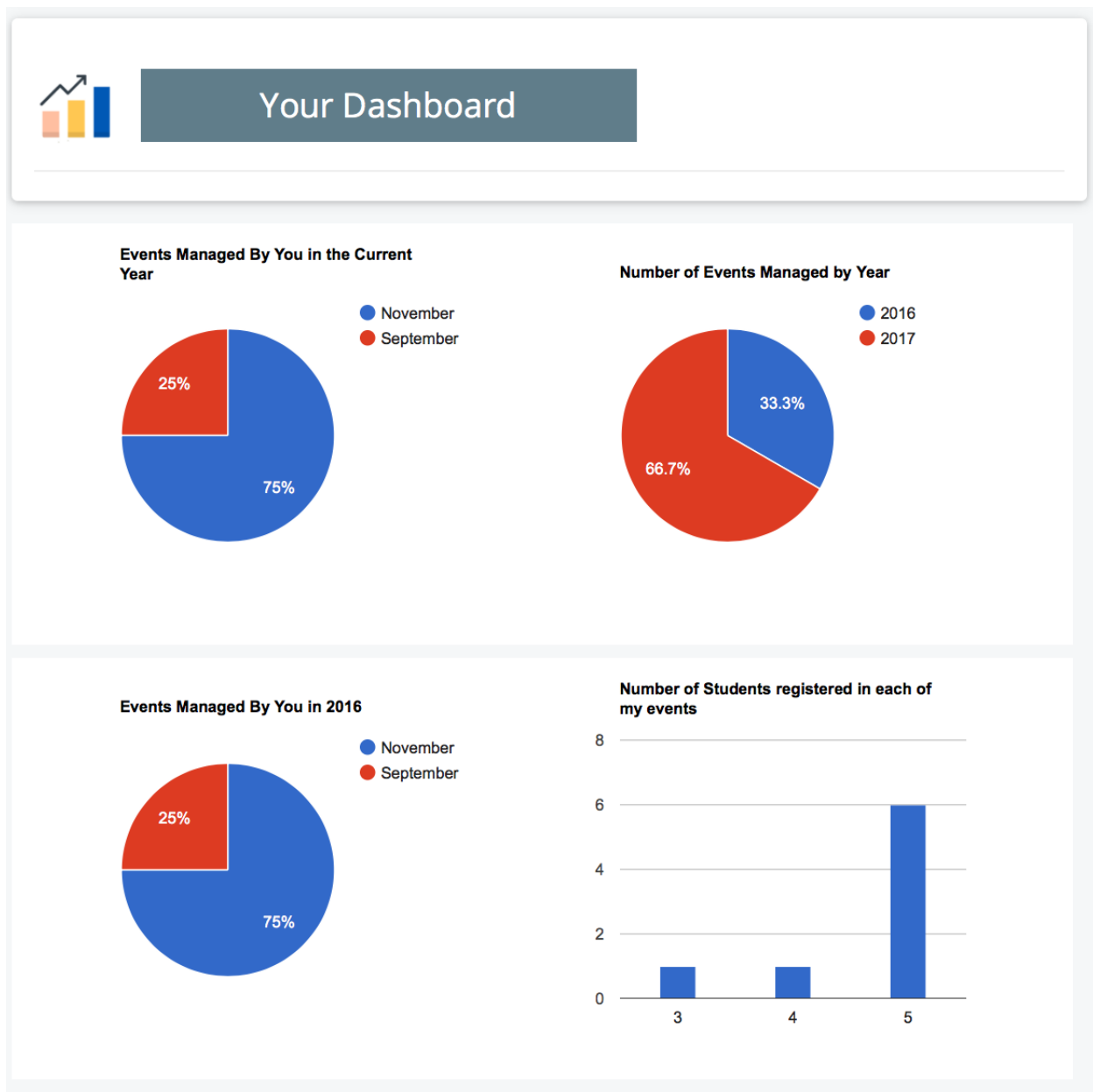
- Number of events created in the current year
- Number of students registered to each event
- Number of users (Students and Event Organizers) registered to the application
- Number of events requiring approval



Event Organizer's Dashboard:

- Number of events organized by Month in the current year
- Number of events organized by the Event Organizer in the current year
- Number of events organized by the Event Organizer in the previous year

- Number of students registered to the each of the events organized by the concerned Event Organizer



2.6.5. DB Component

The project implements a MySQL database centralized on an EC2 instance of AWS.

It contains the following tables

1. Events – Contains the details of the events created by various Event Organizer
2. register_event – Contains all the details of the users/students who have registered to the events organized

3. login – Contains the user name and the respective password for each of the users registered to the application
4. Billing – Contains the charges cost for the Event Organizer for the events registered through this application

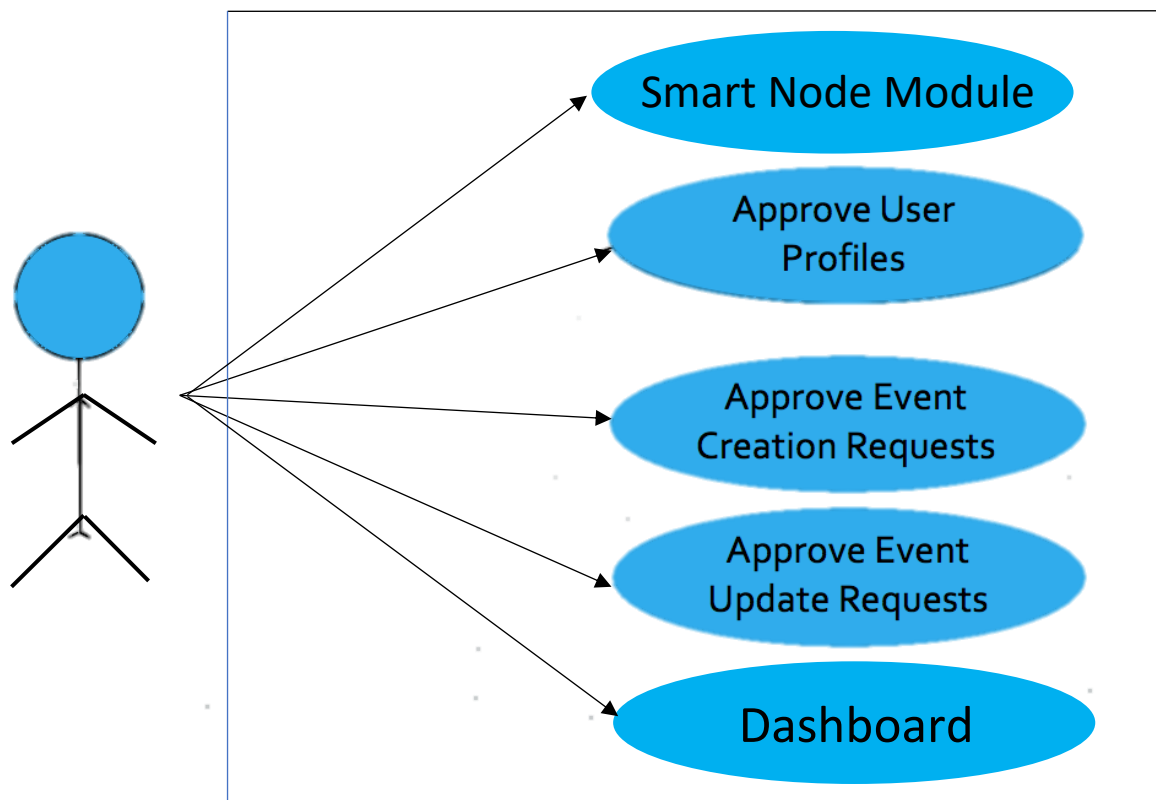
2.7. Use Case Analysis

This application serves three types of users namely,

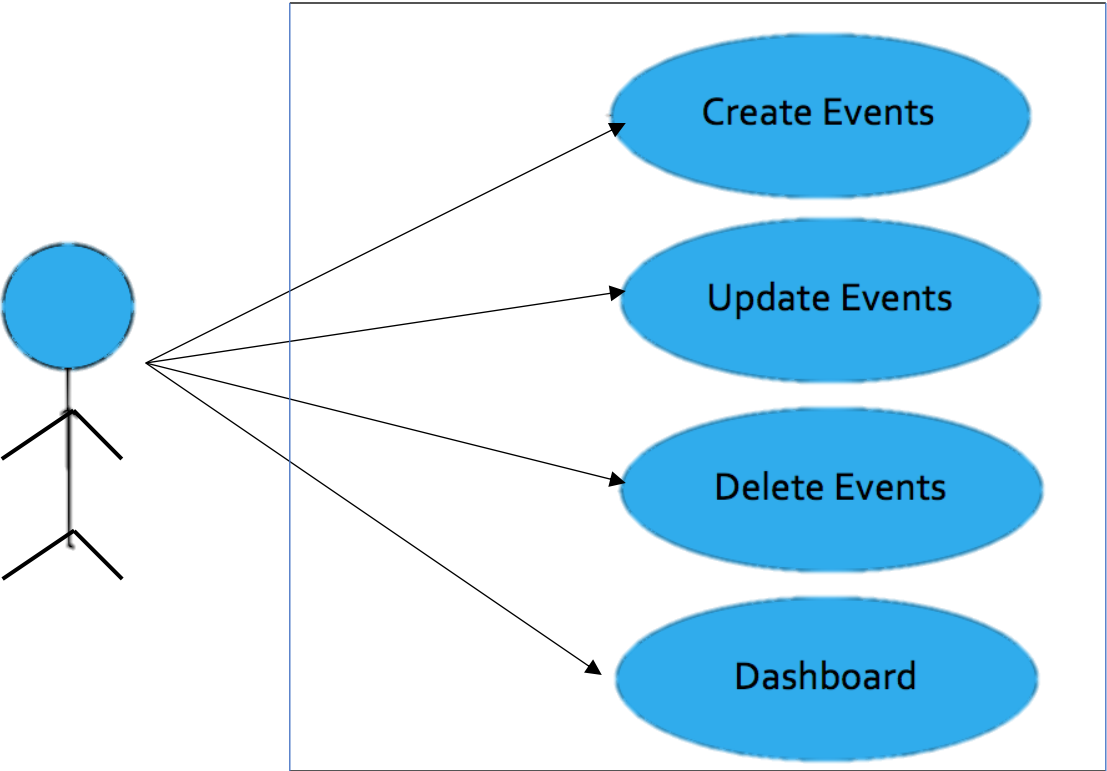
1. An administrator
2. Event Organizers
3. Students

Each of the users can perform various activities and make use of the services provided by this application based on their roles. Below diagrams describe the various use cases of each of the users.

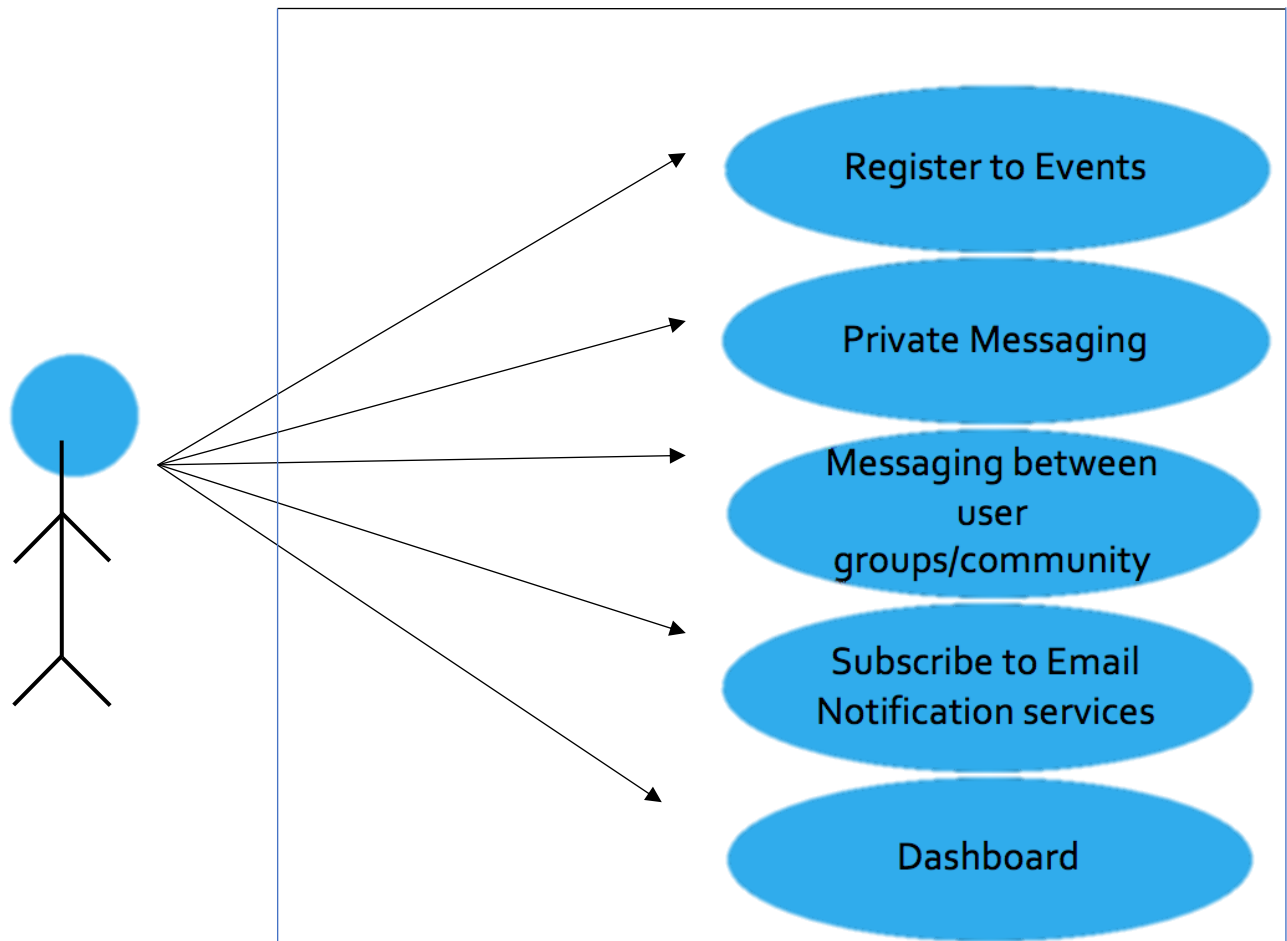
Use Case of an Administrator



Use Case of an Event Organizer



Use Case of a Student



3. Billing Module/Dashboard

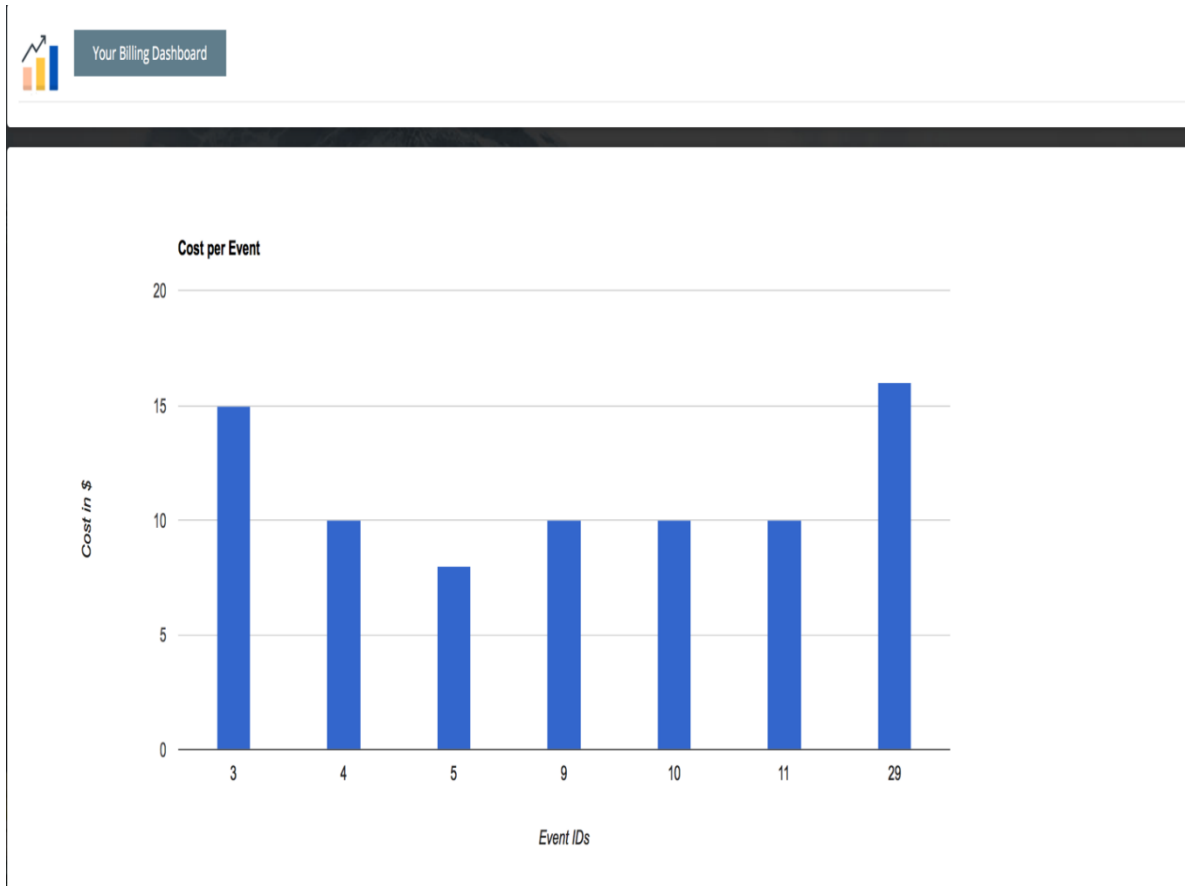
The project implements a 'pay as you go' utility service model. This module is available for the Event Organizers. They are charged based for the events hosted and the number of students that register to events through our application 'EventUp'.

Billing – Utility Model is described as below:

Event Registrations incurs a basic fee of \$10 for each event registered in the application by the Event Organizer

- For 'Basic' users (Event Organizers having users Less than or Equal to 25), the usage cost is \$0
- For 'Premium' users (Event Organizers having 26 to 100 users), the usage cost is \$2/day
- For 'Premium' users (Event Organizers having 101 to 500 users), the usage cost is \$5/day

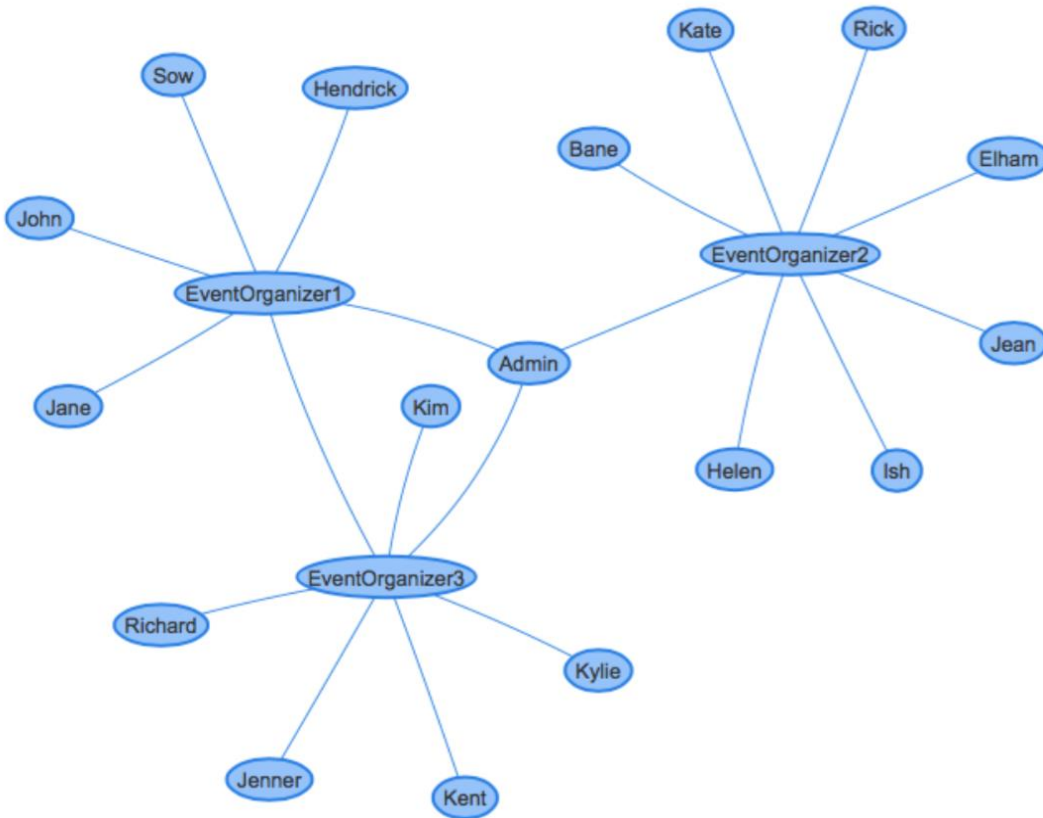
- For 'Premium' users (Event Organizers having users greater than 500), the usage cost is \$10/day



4. Smart Node Module

Smart Node module has been create using a visual library call VisJs which is a dynamic visualization tool. It helps to visualize the networking or connectivity in a community.

The nodes in this smart node module are that of the various users in the application and edges represent the connectivity between them. Each Event organizer represents one community, and each community has a set of users. There is an admin to manage the various events and their users.

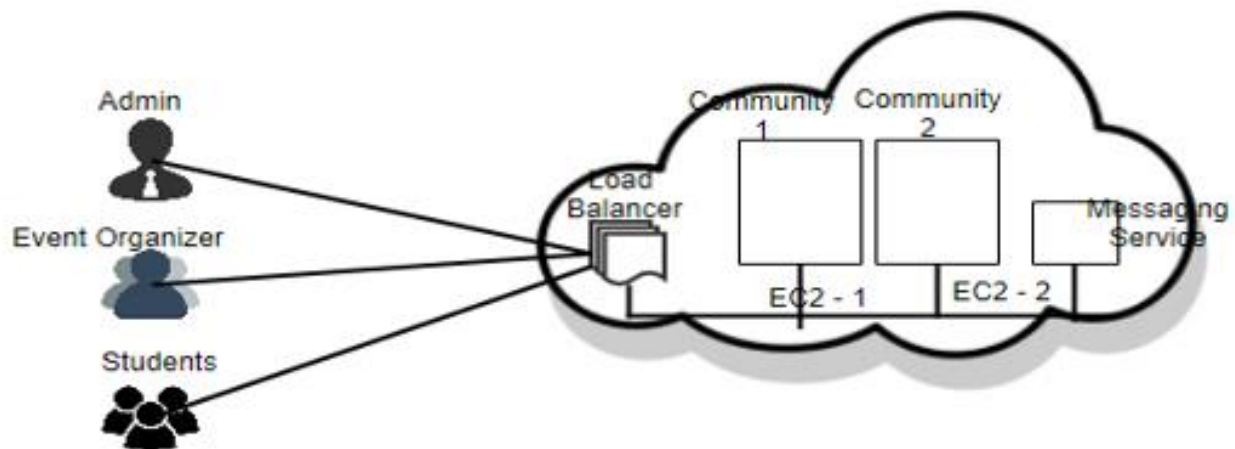


5. Scalability

The application is hosted on a cloud infrastructure and hence it is highly scalable.

The components required to achieve scalability are

- Automatic load balancing using an AWS Load balancer
 - The load balancer provides automatic balancing of network traffic between instances
 - Hence prevents overloading of instances, thus achieving high availability
- Community instances using AWS EC2
 - AWS EC2 instances are used to host the application and the instances can be increased based on the number of users
 - Hence provides high scalability



6. Multi-Tenancy

Our application has been hosted on multiple EC2 instances and each instance serves one community. This provides scalability. Each community can have multiple users/students who can register into one event. Thus, our application satisfies the multi-tenancy criteria also.

7. Benefits

1. The application is hosted on cloud infrastructure and hence it is reliable and secure
2. No upfront investment cost required
3. Maintenance cost are greatly reduced, since security patches, upgrades to the infrastructure will be taken care by the cloud infrastructure service provider
4. Application can be accessed and is independent of location and highly available
5. Highly scalable
6. Uses pay as you go utility service model and hence users only pay according to their usage

8. References:

1. https://en.wikipedia.org/wiki/Smart_city
2. <http://theconversation.com/how-to-ensure-smart-cities-benefit-everyone-65447>
3. <https://readwrite.com/2016/12/14/3-benefits-cities-can-gain-smart-infrastructures-dl4/>