CHRISTIAN SERVICE UNIVERSITY COLLEGE

DEPARTMENT OF COMPUTER SCIENCE

TOPIC: STUDENT CIRCLE

THIS PROJECT IS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF A DEGREE OF BACHELOR OF SCIENCE IN COMPUTER SCIENCE

BY:

KOBINA HOYTE-WILLIAMS 10148597

SUPERVISOR:

MRS. LINDA AMOAKO-BANNING

JUNE, 2015

DECLARATION

"I declare that I have wholly undertaken the study reported herein submitted"

_____ _____

Kobina Hoyte-Williams Date

SUPERVISOR

"I declare that I have supervised the students in undertaking the study report herein and I confirm that the students have the permission to submit it for assessment."

_____ _____

Mrs. Linda Amoako-Banning Date

ABSTRACT

Student Circle is a web based chatting application built to enhance effective teaching and learning especially in the tertiary education settings. Student Circle provides users cost effective means of communicating through private as well as group chatting. Also video calling can be done. A discussion forum is also present for students to share their ideas on the various subject matters and a voice recognition feature that enables audio feeds be converted to text. In order to ensure proper planning for students, the feature for the creation of a timetable is present.

This system provides a cost effective means of communication amongst the student body. All that the user requires is a browser that supports JavaScript such as Mozilla Firefox, Google Chrome, Safari and internet access.

DEDICATION

This project is dedicated to the Almighty God for His protection, love and care throughout the time spent on campus. I also dedicate the project to my supervisor for the encouragement and all the support she gave me during the course of the project. The last dedication goes to my parents, family and all loved ones who supported me in diverse ways.

ACKNOWLEDGEMENT

Sincerest gratitude appreciation goes to the Almighty God, who has made this project very successful. To my family and colleagues, thank you for your financial and moral support. Special thanks also go to Mrs. Linda Amoako-Banning, who supervised this project. May God bless and reward you for your guidance and support.

Table of Contents

CHAPTER ONE

INTRODUCTION

Information sharing is an integral aspect of student education and a major significance of communication .Student Circle is an application that is to foster this goal, through an easy and cost effective means of communication among students and also between lecturers and students. It is a web application which facilitates communication between users on the network as it is less expensive communicating via the internet than through normal phone calls. It also provides students with the certain services that aim at easing course work.

Many other existing multimedia communication software were studied before undertaking this project. I chose to work on this project not to condemn the existing works but to have solved some of their various short comings. Thus the previous works are greatly acknowledged as this system is built on the technological platform they laid.

This System, Student Circle is developed using Node.js, which is an open source, platform runtime for server-side and networking applications in real time. In the past when users wanted content, she would load up her browser, point it at a URL, and get the content. If she wanted to write a blog post. She would load up her browser, fill out a form and press submit. But now the website must contact the user wherever the user may be [3]. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, Linux, FreeBSD, and IBM. Node.js provides an event-driven architecture and a non-blocking I/O API that optimizes an application's throughput and scalability. These technologies are commonly used for real-time web applications. Node.js uses the Google V8 JavaScript engine to execute code, and a large percentage of the basic modules are written in JavaScript. Node.js contains a built-in library to allow applications to act as a Web server without software such as Apache HTTP Server or IIS. This project

has been developed to achieve text chatting, video calling, voice recognition among many others. Node.js provides a favorable framework for development of java scripts for the creation of the graphical user interface.

1.1 MOTIVATION FOR PROJECT

The main motivation for this project was to bridge the communication gap between lecturers and students especially those on the distance learning programme. The fact that communication is not so easy when the parties are in different geographic locations; it's either an issue of the cost or the network problems. The challenge faced by students in meeting for group discussions, the problem of finding conducive timing for all as well as the issue of venue. Most importantly is also the issue of security.

1.2 GOALS AND OBJECTIVES

1.2.1 GOALS OF THE PROJECT

The goals of the project include:

- To bridge the communication gap between students and lecturers as well as other computer users.
- To provide a cost effective means of communication among the parties involved
- To increase productivity in terms of school work making it more effective and efficient.

1.2.2 OBJECTIVES OF THE PROJECT

The project objectives include:

- To enable communication among users through text, audio, video.
- To create a forum that enables users to share ideas on various topics.
- To enable users to obtain a transcribed version of audio feeds provided.
- To enable users create a personal timetable.

1.3 MAJOR CHALLENGES

The major challenges faced in the development of the project include;

- The Application Programming Interface (API) chosen was new to me, hence much time was spent researching and studying it .As well as how to implement the various features using the language.

The timing Constraint:

- The project is to be developed and produced in 4 months, which isn't sufficient to produce an excellent work.
- Also the period is too short to obtain relevant feedback for the necessary improvements.

1.3.1 METHODS OF INVESTIGATION

It was basically through observation and through personal usage of the system and the complaints of other users. Thereby making it easier to identify the shortcomings major of which is a not so user friendly GUI.

1.4 APPROACHES TO THE PROJECT

During the development of the software Student Circle, extensive research was done in the following areas:

- Already existing systems that performs similar functions in a way such as Skype.
- Node.js, socket IO.

1.4.1 PROJECT EVALUATION

At the end of the project, it would be evaluated against two major criteria namely:

ECONOMIC FEASIBILITY

For a project to be economically feasible, it means it is worth more than its cost of production.

Through the response of the users it would be determined if the project

- Solved problems better than existing systems.
- Achieve its proposed features as stated in the proposal.

  How the users feel about the project would also be known. Whether the system is

- Error free
- User friendly

TECHNICAL FEASIBILITY

This determines whether existing technological frameworks would be able to support the features introduced in the project

1.5 STRUCTURE OF THE REPORT

CHAPTER 1:

Introduction to the project

This chapter is an opening to the project and gives the major reasons for the coming up with this project.

Chapter 2: Review of Literature and Existing Systems.

This chapter gives an in depth look at existing systems, problem identification. The methodology, the proposed systems, its advantages and the scope of the system is discussed here.

Chapter 3: Requirement specification

This chapter looks at the approaches used to derive the functional and non-functional requirements of the system.

Chapter 4: Design specification

This chapter discusses the various design techniques employed in developing the system. The various UML diagrams and the relationship between the various components.

Chapter 5: Implementation of Solution

This chapter discusses how implementation and testing is performed.

Chapter 6: Project Evaluation

This chapter evaluates the project, the solution and the methodology.

Chapter 7: Conclusion

This chapter identifies the achievements, challenges, limitations of the project, and then considerable recommendations are given for future work on the project.

CHAPTER TWO

REVIEW OF LITERATURE

2.1 BACKGROUND REVIEW

The relevance of Communication cannot be underestimated in our everyday lives. It is the channel through which we share ideas, suggestions made, problems and their corresponding solutions are obtained.

Thus the software Student Circle has come to foster that by placing emphasis on lecturer student communication and communication amongst the student body. The setting in question is the CSUC campus, which comprises over 5,000 students. A number of which are part-time and weekend learning students. Thus they would have to travel from where ever they are to convene on campus and have class periodically. But these are not regular students who probably reside on campus and may also be working. Thus this constant travelling might not actually favor them. In this wise Student Circle affords both lectures and students the comfort of having class online without necessarily converging at a common venue. All that is left would be to schedule time for the class. Thereby saving cost and time of commuting.

Also as mentioned earlier are group studies which play a vital role in learning on campus. Here, students decide to meet at a location at a particular time to discuss certain topics taught in class. The challenge in this direction is finding a safe and conducive atmosphere for everyone while not strongly inconveniencing anyone. Student Circle has come to help find solutions to some of the mishaps associated with the group studies. The focus is to provide a serene, conducive and most especially a secure environment for learning where students can actually have this group studies in their homes.

The other features are there to enhance communication, enable a more productive and focus driven studying that is with the discussion area, the voice recognition and timetable

creator. This would give students direction as to how to plan their time judiciously. As the saying goes: “If we fail to plan we plan to fail”.

2.2 REVIEW OF EXISTING SYSTEM

On campus communication is conducted through phone calls and the various other media available like whatsapp and viber. These come with the shortcomings of the internet cost, as well as the complexity of downloading, installing and configuration especially for IT challenged students.

Many systems exist that enable user’s chat by audio, by text and even by video, such as Skype, Whatsapp and Facebook. Despite the existence of these systems and many more, there is no existing application that presents all the features of text chat, video calling, group chatting, discussion forum, voice recognition and personal timetable creation in one simple application.

Thus to have access to all these features, users would have to install and configure multiple applications before being able to implement these modules in multimedia conferencing. This would cost them money for the data as well as storage space on the devices on which they are installed.

A comprehensive study of the existing system, showed the lack of an intuitive graphical user interface whose importance cannot be undermined. Just as appearance speaks volumes about a person so does the user interface say for software. It gives the user insight as to what the system is about and thus it should be easy to use. With a few button clicks users should be able to navigate through the system with ease and without much trouble.

With previous systems students would have to meet physically before group studies or project meeting are effective.

Another shortcoming of the existing system is cost of making phone calls .Hence when students are short of money that means of communication is curtailed.

2.2.1 EXISTING SYSTEM PROCEDURE

Currently the trend in CSUC is, have credit, make a call, otherwise have data and use some of the other chatting applications to communicate. If none of the above conditions are available then there will be no communication.

Students who wish to have group studies to further enlighten themselves on the topics being lectured in class, after having agreed on the course to discuss would have to work obtaining a safe and conducive venue and pick a time that is convenient for everyone.

Project students must periodically meet to work on their project together.

Missing out on lecture notes due to inaudibility of the dictation.

The misuse of time due to the lack of proper planning by the students.

2.3 PROBLEM IDENTIFICATION

These are some of the existing problems that the software seeks to solve.

- The issue of obtaining a venue and its security, which is key in organizing group studies. Many at times we hear of occurrence of theft, robbery or even rape of students returning from campus due to long group study hours and the like. This is something that is discouraging the trend of group studies despite its importance. Security can never be compromised.
- Also project partners having to meet in person before they can work on their project. Considering the time and money involved and how inconvenient it is to travel to meetings despite your schedule.

- The money, time and stress it cost distance learning students to make it for class periodically. The inconvenience there as well.
- The cost of making calls using phone credits. The problem of a dead battery or insufficient credit is one that hampers communication in this direction.
- The lack of prioritization or proper time management due to the lack of planning on the part of students especially first years.
- Missing out on lecture notes due to lecturer's inaudibility.

2.4 TYPES OF SYSTEM USERS

The type of users of the system would include;

SYSTEM ADMINISTRATORS

These are people who are responsible for the upkeep and configuration of an organizations server or systems. They are trained to install, maintain and support an organizations information technology system and planning for and responding to service outages and other problems. They also monitor system performance and network communication and ensure system security.[7]

AVERAGE COMPUTER USER

This includes the people who use the computer for basic functions such as typing documents, watching videos and playing games. This range may include the students.

2.5 REVIEW OF RELATED METHODOLOGY

A number of software engineering methodologies exist such as the water fall model and the incremental method. After a careful consideration of what the system was about, the methodology chosen was the incremental approach based on its simplicity and flexibility.

This approach interleaves the activities of specification, development, and validation. The system is developed as a series of versions (increments), with each version adding functionality to the previous version. This method is also ideal for developing web based application like Student Circle.

Incremental development is based on the idea of developing an initial implementation, exposing this to user comments and evolving it through several versions until an adequate system has been developed. Specification, development, and validation activities are interleaved rather than separate, with rapid feedback across activities.

Incremental software development, which is a fundamental part of agile approaches, is better than a waterfall approach for most business, e-commerce and personal systems. Incremental development reflects the way we solve problems. We rarely work out a complete problem solution in advance but move toward a solution in a series of steps, backtracking when we realize that we have made a mistake.

By developing the software incrementally, it is cheaper and easier to make changes in the software as it is being developed as it's a web application. Each increment or version of the system incorporates some of the functionality that is needed. Generally, the early increments of the system include the most important or most urgently required functionality. This means that the system can be evaluated at a relatively early stage in the

development to see if it delivers what is required. If not, then only the current increment has to be changed and, possibly, new functionality defined for later increments.

ADVANTAGES OF THE INCREMENTAL MODEL

Incremental development has three important benefits, compared to the waterfall model:

1. The cost of accommodating changing customer requirements is reduced. The amount of analysis and documentation that has to be redone is much less than is required with the waterfall model.

2. It is easier to get customer feedback on the development work that has been done. Customers can comment on demonstrations of the software and see how much has been implemented. Customers find it difficult to judge progress from software design documents.

3. More rapid delivery and deployment of useful software to the customer is possible, even if all of the functionality has not been included. Customers are able to use and gain value from the software earlier than it is possible with a waterfall model.

DISADVANTAGES OF THE INCREMENTAL MODEL

From a management perspective, the incremental approach has two problems:

1. The process is not visible. Managers need regular deliverables to measure progress. If systems are developed quickly, it is not cost-effective to produce documents that reflect every version of the system.
2. System structure tends to degrade as new increments are added. Unless time and money is spent on refactoring to improve the software, regular change tends to

corrupt its structure. Incorporating further software changes becomes increasingly difficult and costly.

2.6 THE PROPOSED SYSTEM

This web application encompasses quite a number of communicating tools that allows communication between lecturers and students, tools to assist students and lecturers in their course work.

2.6.1 OBJECTIVES OF THE PROPOSED SYSTEM

The objectives for Student Circle include:

- To enable communication among users through text or video calling.
- To enable users create personal timetables.
- To allow as many users chat through group chat.
- To enable users share varied ideas on various topics through the discussion forum.
- To enable users transcribe audio messages into text.

2.6.2 FEATURES OF THE PROPOSED SYSTEM

The features of the system are grouped into three main features, namely:

- Chat application
- Workspace
- Student Assistance

Under the chat application are the following features:

TEXT CHATTING: This would enable connected users chat using text, by providing a text area that accepts the input of the user. All text can then be displayed as a conversation.

VIDEO CALLING: This captures video from a webcam and audio from the computer microphone and transmits it to the other connected user. This allows users receive audio and video streams from one another.

GROUP CHATTING: This enables groups to be created by the class representative, where members of the class contribute to various subject matters under the topic headers.

Under the workspace is

VOICE RECOGNITION: This enables users transcribe audio messages into text which can be saved on the computer or sent to their emails.

Under the student assistance are the following:

CLASS TIMETABLE: This enables timetable of a class to be uploaded and made available to all members of the class.

PERSONAL TIMETABLE: This creates a timetable for users after they have specified the time and the activity to be carried out.

DISCUSSION FORUM: This enables users post questions or problems in their course work and have members help find solutions to them.

2.6.3 SCOPE

The scope of the project is in all areas where computers are used, as communication occurs everywhere in the world, but the focus is mainly among the student body.

The target users are system administrators and average computer users.

2.6.4 CASE STUDY

A survey on campus, CSUC revealed the challenges students have to go through communicating through phone calls and other similar costly media (internet cost).

Also weekend learning programmes being run on campus, the challenge of students having to move from various destinations to converge on campus to be taught. Considering the fact that most of these students work as well, thus, it would be stressful after having to sacrifice weekends for lectures.

2.7 REVIEW OF PROJECT PLATFORM

2.7.1 DEVELOPMENT TOOLS AND ENVIRONMENT

- Node is command prompt
- JavaScript
- Php
- MySQL
- Socket.io
- Express Framework
- Internet for research

1. NODE.JS

It is an open source, cross-platform runtime environment for server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, Linux, FreeBSD, and IBM i.Node.js provides an event-driven architecture and a non-blocking I/O API that optimizes an

application's throughput and scalability[5]. These technologies are commonly used for real-time web applications. Node.js uses the Google V8 JavaScript engine to execute code, and a large percentage of the basic modules are written in JavaScript. Node.js contains a built-in library to allow applications to act as a Web server without software such as Apache HTTP Server or IIS.

Why node.js

Node.js is a new platform and is still evolving, but even in its infancy it is probably one of the most popular platforms on the web. It is already powering popular services. [4]

Interoperability or Cross Platform Compatibility

It provides interoperability across network of different operating systems. This project has been developed to achieve text, audio, video conferencing, multiuser workspace and multiuser presentation among many others. Node.js provides a favorable framework for development of java scripts for the creation of the graphical user interface.

Scalability

Node.js is scalable in the sense that it can have multiple jobs for node to process and it can handle it without much burden. I/O is safer and easier to do parallel as it tends to share no data between execution threads. Node.js lets you do it using event programming which is simple, elegant and easy to use. Is an old and proved programming paradigm, used for years by GUIs and other graphical intensive apps like games.

I.O Bound

I.O bound problem are alleviated by increased throughput in I/O such as disk memory, network bandwidth, and improved data caching. Many problems are I/O bound, and it's in this domain that Node.js truly shines.[2]

2. SOCKET IO

Socket IO is a java script library for real time web applications. It enables real time, bi-directional communication between web clients and server. It has two parts a client-side library that runs in the browser, and a server-side library for node.js [6]. Both components have a nearly identical API. Like node.js is event-driven.

The benefit of using socket IO is that it handles the connection transparently. It will automatically upgrade to Web Socket if possible .This requires the programmer to only have Socket.IO knowledge

CHAPTER THREE

REQUIREMENTS SPECIFICATION

3.1 REQUIREMENTS CAPTURE

The software requirements document (sometimes called the software requirements specification or SRS) is an official statement of what the system developers should implement. It should include both the user requirements for a system and a detailed specification of the system requirements. Sometimes, the user and system requirements are integrated into a single description. In other cases, the user requirements are defined in an introduction to the system requirements specification. If there are a large number of requirements, the detailed system requirements may be presented in a separate document.

3.2 COLLECTION OF REQUIREMENTS SPECIFICATION

This is the means through which the necessary information was gathered for this project work.

- THE INTERNET
- DIRECT OBSERVATION

3.3 USER REQUIREMENTS

Requirements specification is the process of writing down the user and system requirements in a requirements document. Ideally, the user and system requirements should be clear, unambiguous, easy to understand, complete, and consistent. In practice, this is difficult to achieve as stakeholders interpret the requirements in different ways and there are often inherent conflicts and inconsistencies in the requirements.

This description may use natural language, diagrams, or other notations that are understandable to customers. Product and process standards that must be followed should be specified.

Ideally, they should specify only the external behavior of the system. The requirements document should not include details of the system architecture or design.

The user requirements for a system should describe the functional and nonfunctional requirements so that they are understandable by system users who don't have detailed technical knowledge.

3.3.1 FUNCTIONAL REQUIREMENTS

The functional requirements for a system describe what the system should do. These requirements depend on the type of software being developed, the expected users of the software, and the general approach taken by the organization when writing requirements.

When expressed as user requirements, functional requirements are usually described in an abstract way that can be understood by system users. However, more specific functional system requirements describe the system functions, its inputs and outputs, exceptions, etc., in detail.

Functional system requirements vary from general requirements covering what the system should do to very specific requirements reflecting local ways of working or an organization's existing systems.

The functional requirements are what the system should do but should describe the services performed by the system into details.

For this project the functional requirements include;

The user will be able to

- Login into the system through an authentication process.
- Support multiple connections.
- Enable the user to be able to chat with their users through video calling
- Have a group text chat
- Create a personal timetable
- Transcribe audio messages into text.
- Have a private text chat with another user.
- Have contributed to discussions under the various headers.

Transmission Mode

The system supports the following transmission modes;

- **Broadcast**: user sends messages to all active clients.
- **Multicast:** user sends messages to more than one client on the system
- **Unicast:** user sends message to only one other user on the system.

3.3.2 NON FUNCTIONAL REQUIREMENTS

These are constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process, and constraints imposed by standards. Non-functional requirements often apply to the system as a whole, rather than individual system features or services.[1]

Non-functional requirements, as the name suggests, are requirements that are not directly concerned with the specific services delivered by the system to its users. They may relate to emergent system properties such as reliability, response time, and store occupancy.

Alternatively, they may define constraints on the system implementation such as the capabilities of I/O devices or the data representations used in interfaces with other systems.

Non-functional requirements, such as performance, security, or availability, usually specify or constrain characteristics of the system as a whole. Non-functional requirements are often more critical than individual functional requirements. System users can usually find ways to work around a system function that doesn't really meet their needs. However, failing to meet a non-functional requirement can mean that the whole system is unusable.

The non-functional requirements of the system will include:

Usability by target user community

This is the ease of use and learnability of the system by the target user. The existence of an intuitive Graphical user interface made this possible

Security

As a web application, the HTTPS protocol present ensures this. It provides authentication of the visited website and to protect the privacy and integrity of exchanged data.

Efficiency

Fast response time, quick data processing and lesser usage of system resources are the characteristics of this feature

Maintainability

This involves a system of continuous improvements, learning from the past in order to improve the ability to maintain systems, or improve reliability of systems based on maintenance experience.

Extensibility

This defines the ability to have new functionality extended without affecting the system's internal structure and dataflow.

Portability

The system is platform independent as it was developed to work with a number of browsers. Mozilla Firefox, Internet Explorer, Safari are among the list.

3.4 SYSTEM REQUIREMENTS

The system requirements are in two folds namely

- Software requirements
- Hardware requirements

3.4.1 SOFTWARE REQUIREMENTS

The software may run best on a system with the following capabilities

- Windows operating system
- Mac ,Unix and other similar operating system
- Mozilla Firefox, Internet Explorer, Google Chrome among others

3.4.2 HARDWARE REQUIREMENTS

The system requires a Pentium II device or more with specifications as

- Processor speed of about 500Mhz
- At least 128MB RAM
- A minimum hard disk space of about 20 GB
- A wireless local area network adapter or local area network port to connect RJ45 network cable or a network modem.

3.4.3 GRAPHICAL USER INTERFACE

- The graphical user interface was designed to give user an easy to use web application.
- Message boxes as well as menus were used to ensure the ease of use for users.
- Buttons and various emoticons were incorporated to guide users through the use of the system.
- Legends were also used in the chat bar, to indicate the status of various users on the system at a time.
- The use of the right font ,images and colours made the application appealing and attractive

3.5 REQUIREMENTS ANALYSIS

This involved the use of the Unified Modeling Language. This a general purpose modeling language in the field of software engineering, which is designed to provide a standard way to visualize the design of the system. The UML presents two different views of the system model. The structural and dynamic format.

- Structural

This emphasizes the static structure of the system using objects, attributes operations and relationships. This structure includes class diagrams and composite structure diagrams.

- Dynamic

This view emphasizes the dynamic behavior of the system by showing collaborations among objects and changes to the internal state of objects. This includes the state machine, sequence and activity diagrams.

3.6 USE CASE DIAGRAM

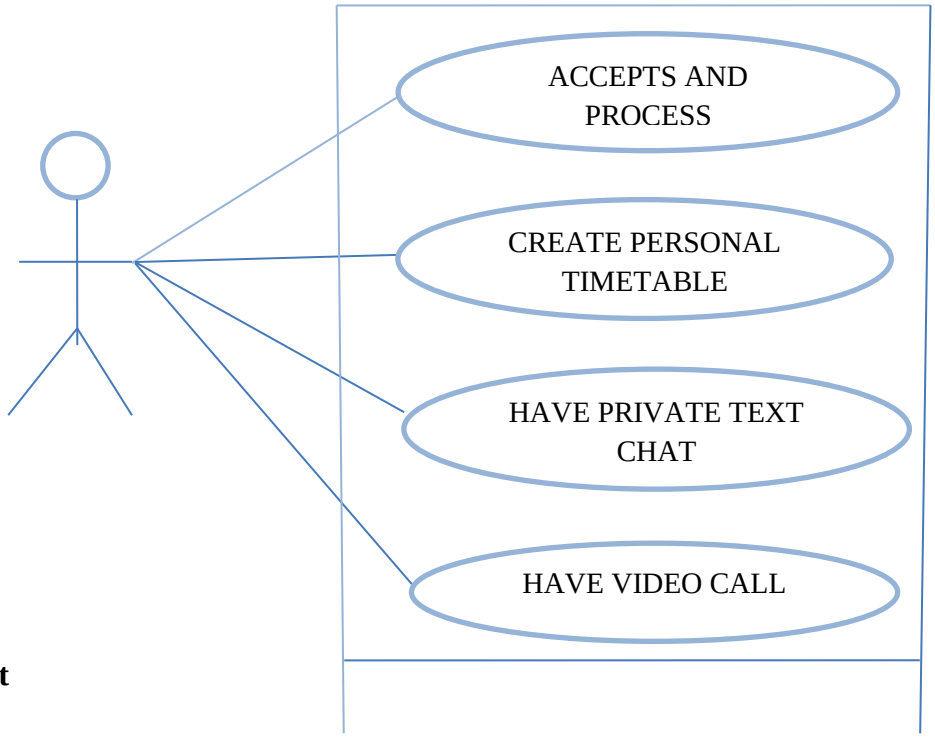
Use case is a requirement discovery technique that was first introduced in the object oriented method .They have now become a fundamental feature of the unified modeling language.

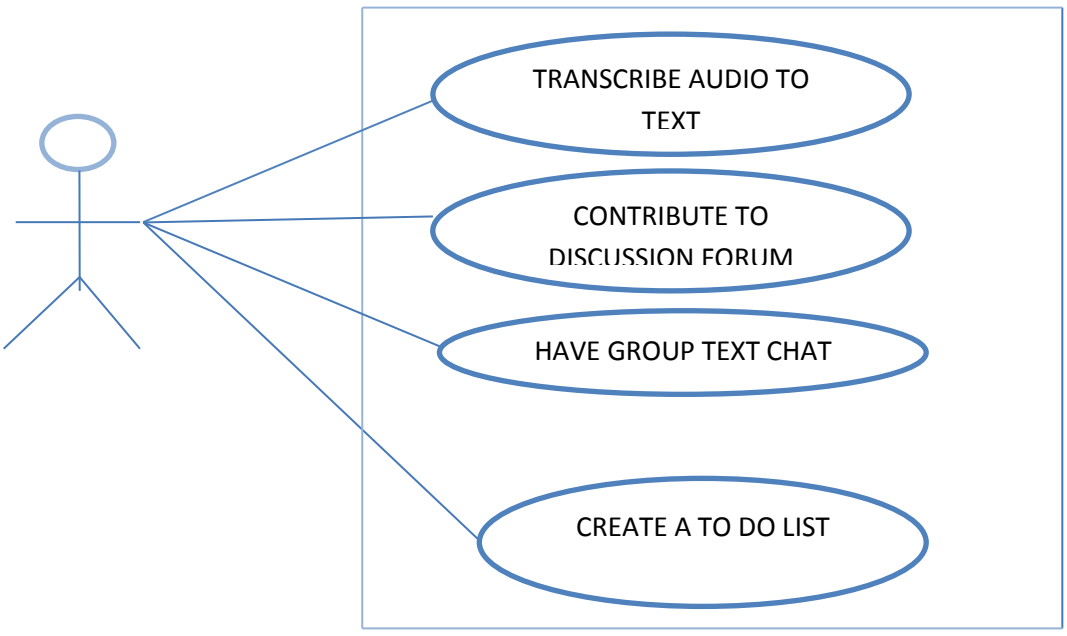
Use cases are documented using a high-level use case diagram. The set of use cases represents all of the possible interactions that will be described in the system requirements. Actors in the process, who may be human or other systems, are represented as stick figures. Each class of interaction is represented as a named ellipse.

Lines link the actors with the interaction. Optionally, arrowheads may be added to lines to show how the interaction is initiated.

Each scenario is a single thread through the use case. Therefore, there would be a scenario for the normal interaction plus scenarios for each possible exception Use cases identify the individual interactions between the system and its users or other systems. Each use case should be documented with a textual description. These can then be linked to other models in the UML that will develop the scenario in more details.

3.6.1 USE CASE DIAGRAM





3.6.2 DESCRIPTIONS

Use case description lists the components of the project step by step so the team can survey the territory ahead. A clear and concise use case description can act as a road map for the project. These descriptions can take either a conversational form (focusing on the interactions between the actors and the system) or a narrative form (telling the story of your project).

Actors refer to:

- Person, organization or external system that plays a role in one or more interactions with your system.
- Users or groups necessary for the system to perform its stated functions.
- Users or groups who use the system to perform tasks
- External systems that use the system to perform a task

The main actors in Student Circle are the students and lecturers. The system works on a peer to peer.

Each of these actors can:

- Have a text chat with other users
- Engage in video chats with other chat mates.
- Transcribe audio feed into text.
- Create a personal timetable.

System boundaries

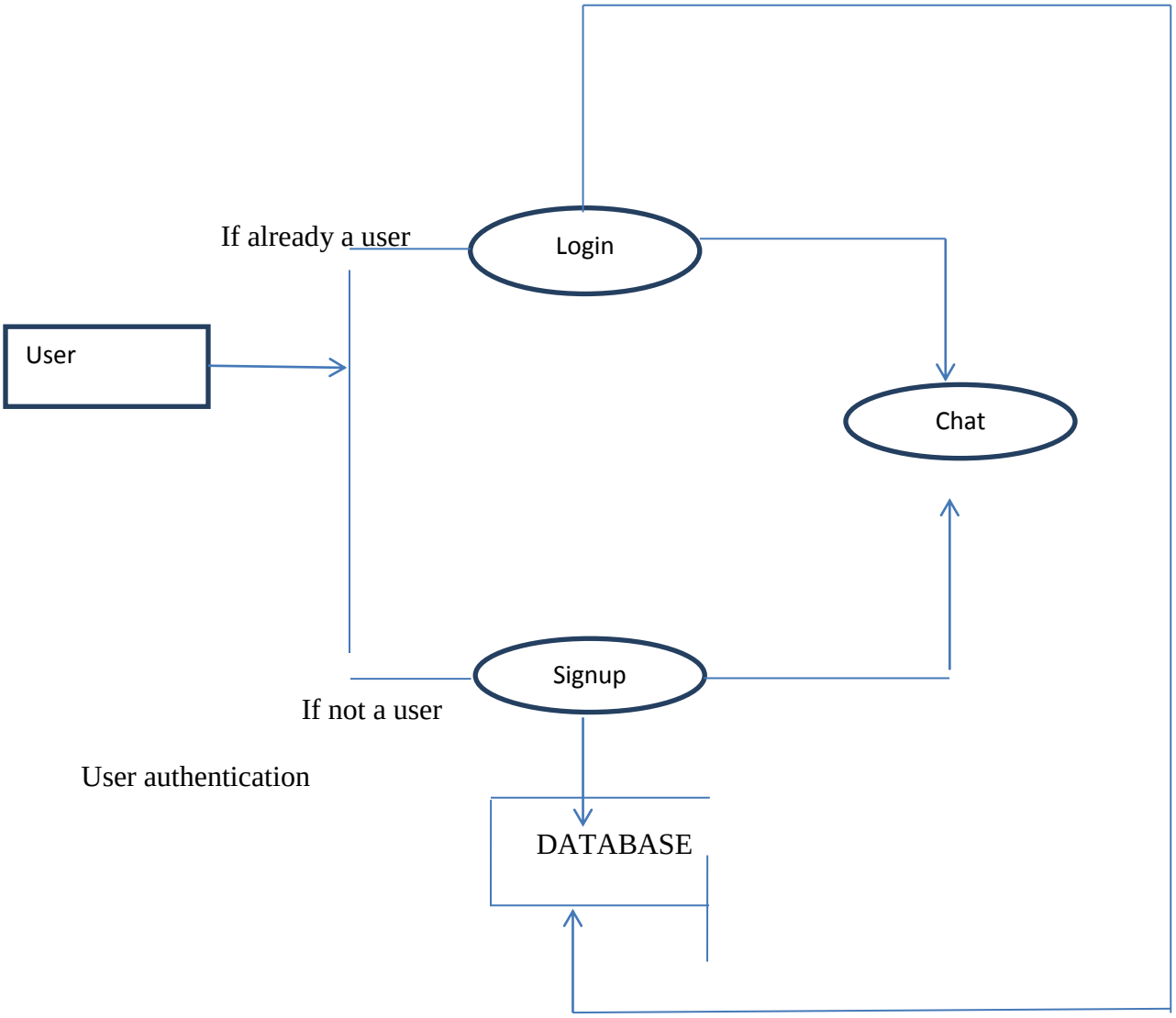
This is indicated by a rectangle around the use cases, signifying the system scope. Whatever is within the box falls within the scope and whatever is out is not.

Use case description

USE CASE NAME	**Use computer as client to execute text, video chat, audio transcriptions**
ACTOR	USER
DESCRIPTION	User logins onto the webpage
PRECONDITION	User should have already signed up to page's account
REQUIREMENTS	• The computers must meet the system requirements • A JavaScript enabled browser must be installed on user's computer • Established internet connection
MAIN FLOW OF EVENTS	ACTOR ACTION User is able to select a chat chosen module available on the Website by clicking the Icon indicating it SYSTEM RESPONSE The system opens the module by the user
POST CONDITION	The connected users online should be able to chat, create timetable or transcribe audio into text

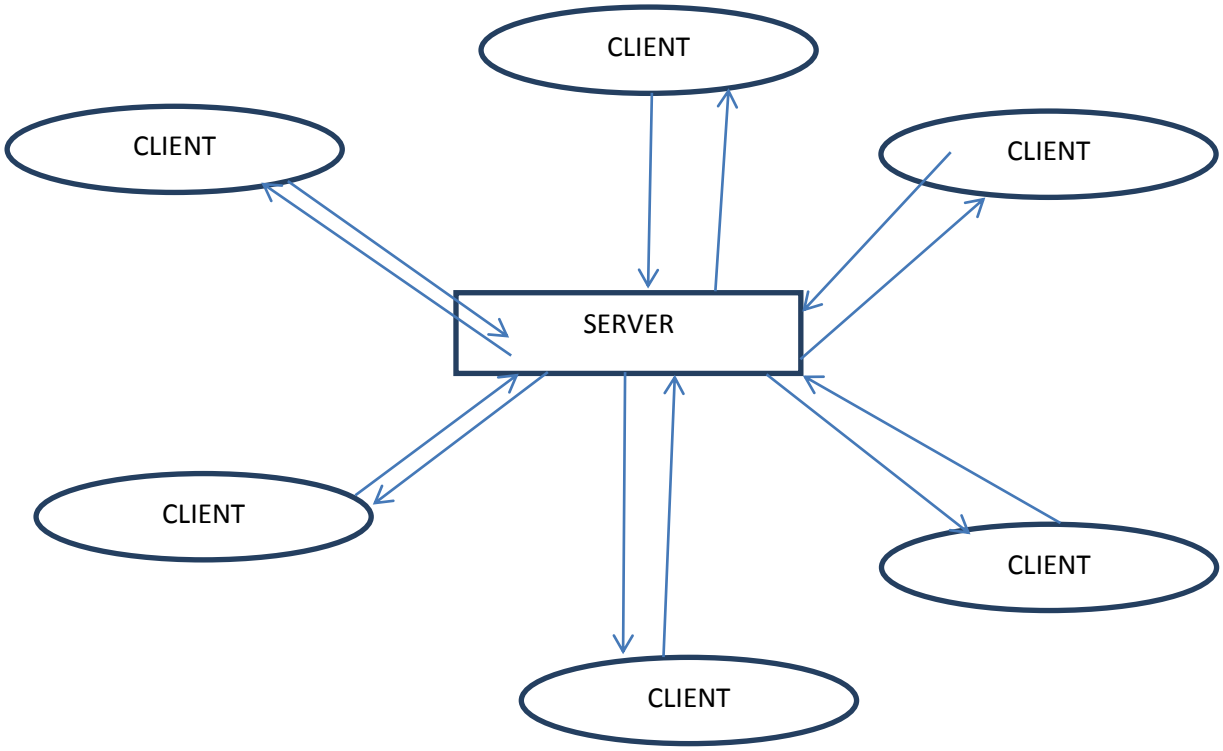
DATAFLOW DIAGRAM

User verification



User authentication

THE SYSTEM MODEL AND ARCHITECTURE



3.8 SECURITY CONCEPT

Basic security feature employed was the login form to ensure the proper authentication of users. This is to enable the right people access the right ends of the system.

Passwords were hashed to ensure they are secured and not available for the view of everyone around.

Socket IO affords a feature that enable connections be closed once the user closes the tab. Hence web sessions would be secured from hackers who wish to spoof.

Also when users logout, IP address of the site changes the next time you log in due to the web RTC technology incorporated. This secures the site from man –in- middle attacks as well as various forms of spoofing by hackers.

CHAPTER FOUR

DESIGN AND IMPLEMENTATION

The various design techniques used during the development of Students Circle is described as follows:

1. **Architectural design**

This is where you identify the overall structure of the system, the principal components (sometimes called sub-systems or modules), their relationships and how they are distributed.

2. Interface design

This is where you define the interfaces between system components. This interface specification must be unambiguous. With a precise interface, a component can be used without other components having to know how it is implemented. Once interface specifications are agreed, the components can be designed and developed concurrently.

3. Component design

This is where you take each system component and design how it will operate. This may be a simple statement of the expected functionality to be implemented, with the specific design left to the programmer. Alternatively, it may be a list of changes to be made to a reusable component or a detailed design model. The design model may be used to automatically generate an implementation.

4. Database design

This is where you design the system data structures and how these are to be represented in a database. Again, the work here depends on whether an existing database is to be reused or a new database is to be created.

Implementation involves translating a design into a program and removing errors where present. Testing is discovering faults in the program and fixes them through the debugging process.

4.1 USER INTERFACES

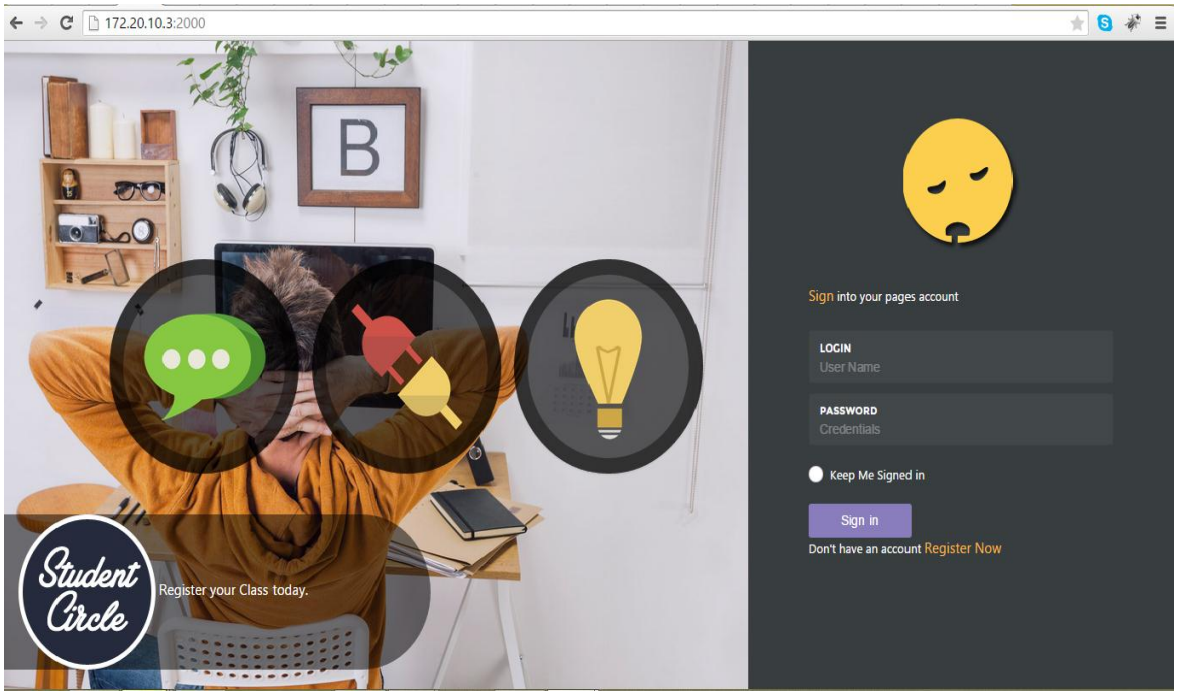


Figure 1: Login Page

This is the first page the user encounters after enters the website address. After the details provided are authenticated they are allowed access to the system.

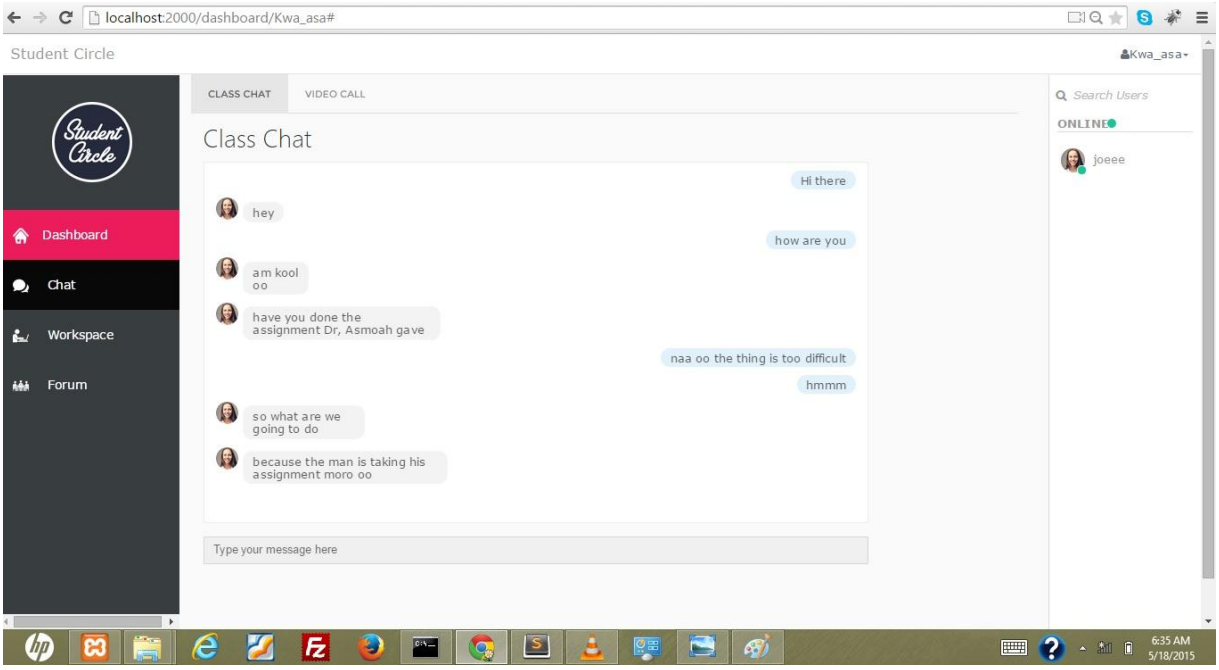


Figure 2: Private Chat

This page shows two users who have logged onto the system chatting via text.

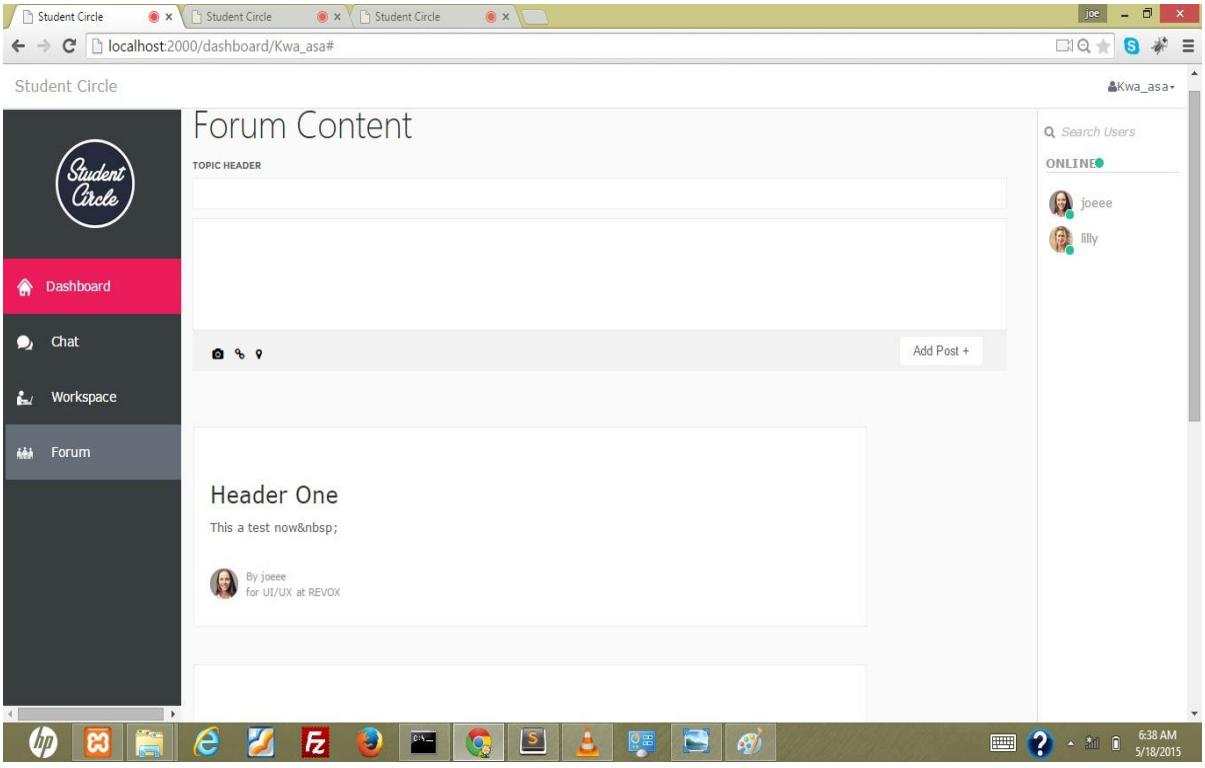


Figure 3: Discussion Forum

This is the where users registered under one class are having a discussion under a particular header.

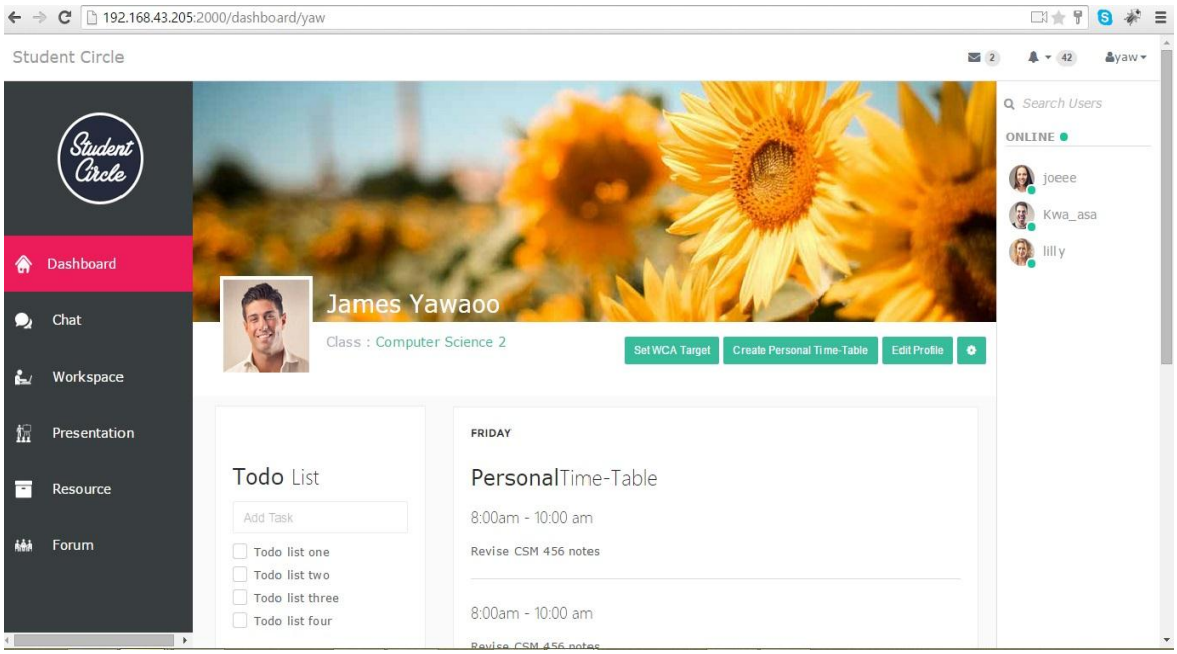


Figure 4: Dashboard

This is the page that a user is redirected to after they login in.

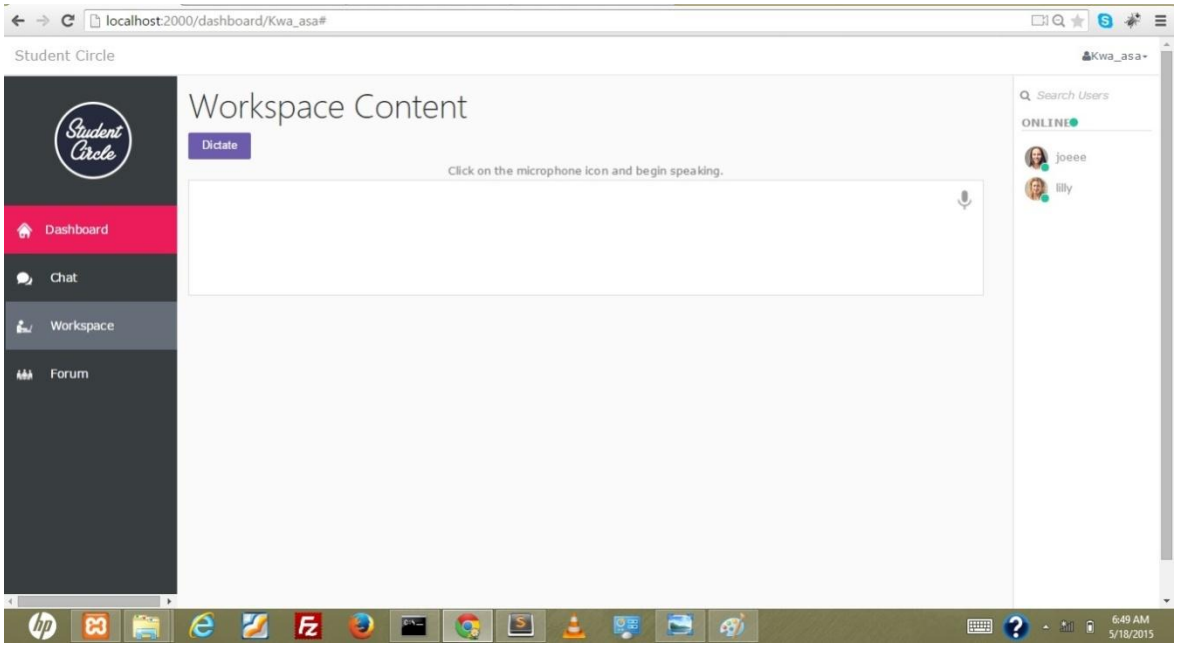


Figure 5: Voice Recognition

This is the page that users submit audio feed to transcribe into words.

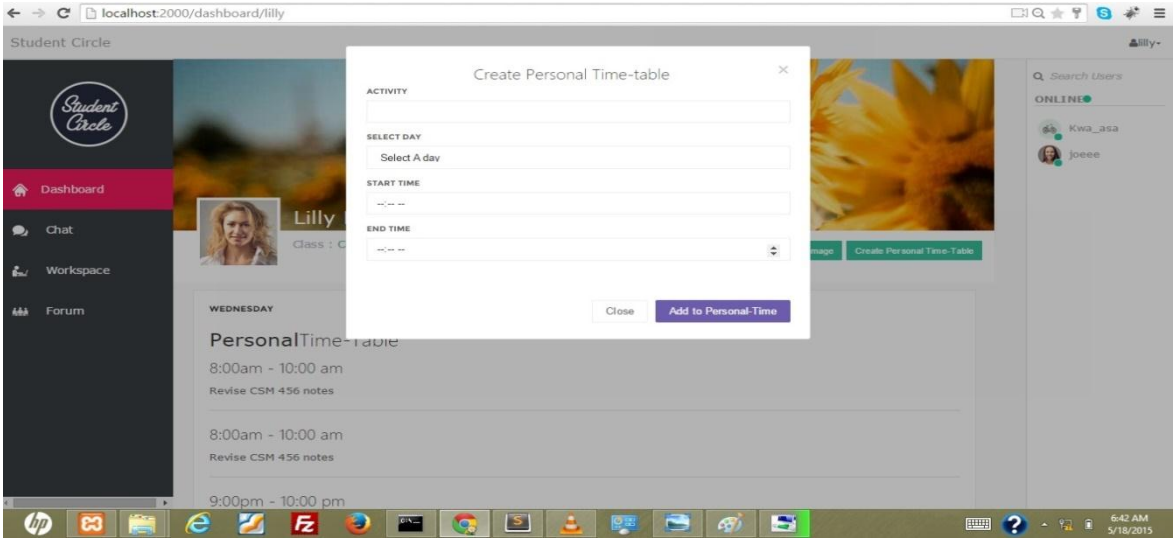


Figure 6: Personal Timetable

This page shows a user creating a personal timetable.

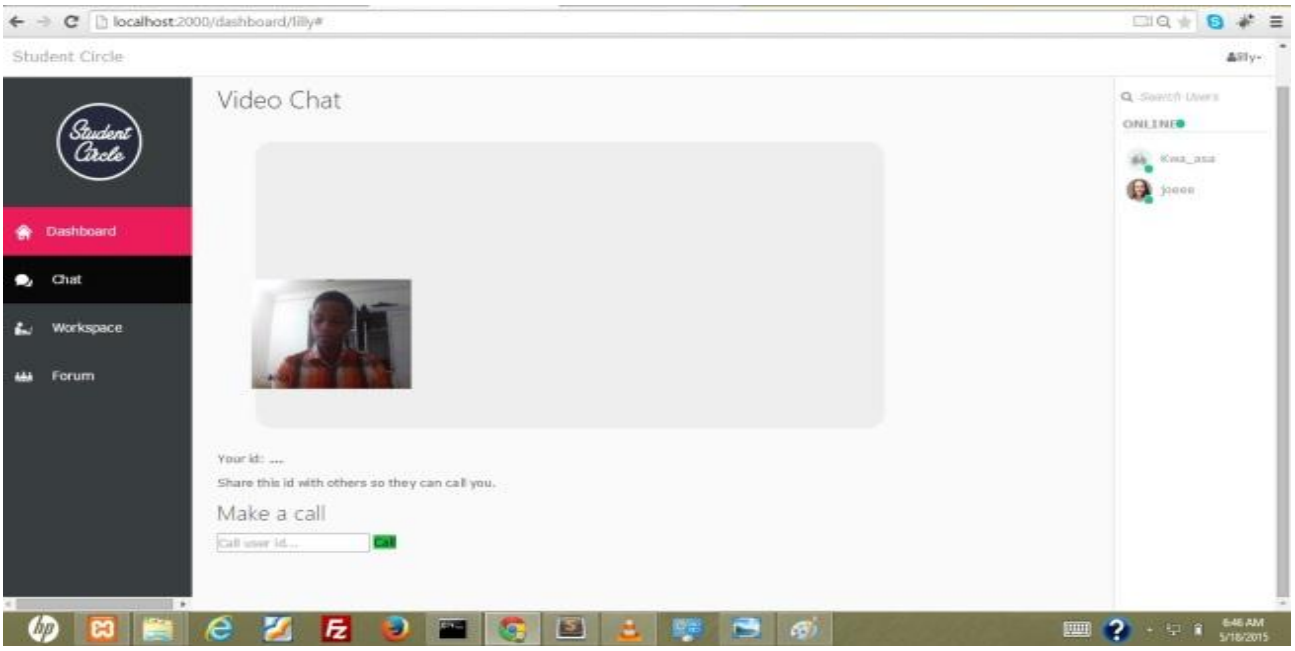


Figure 7: Voice Calling

This shows a user making a video call using the generated id by the system.

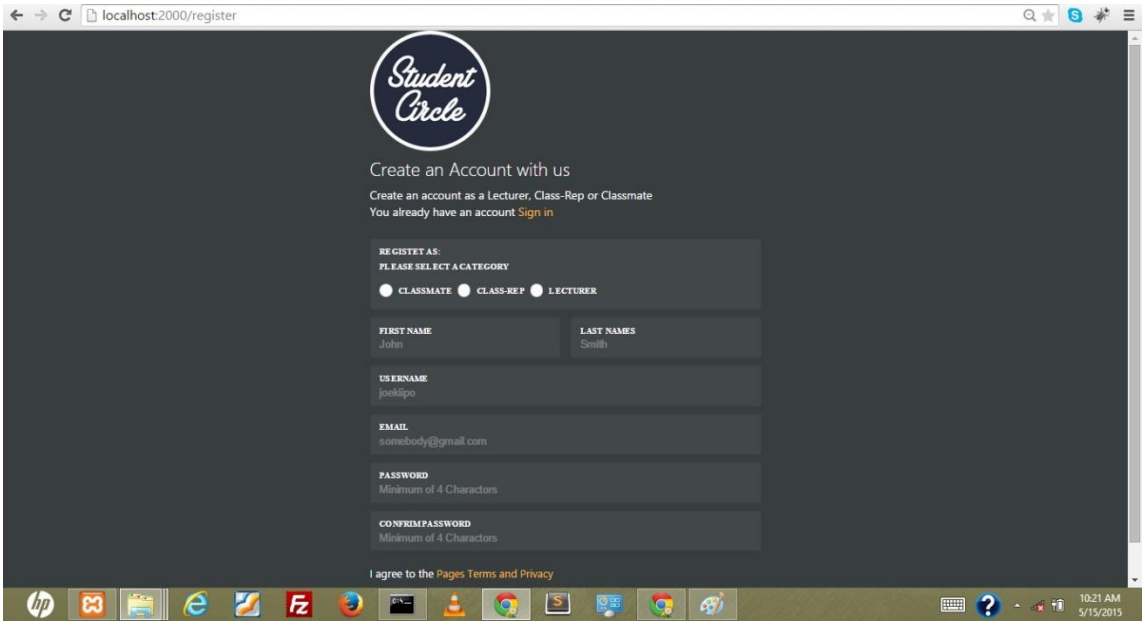


Figure 8: Register Page

This page is for new users to create an account to allow them access to the system.

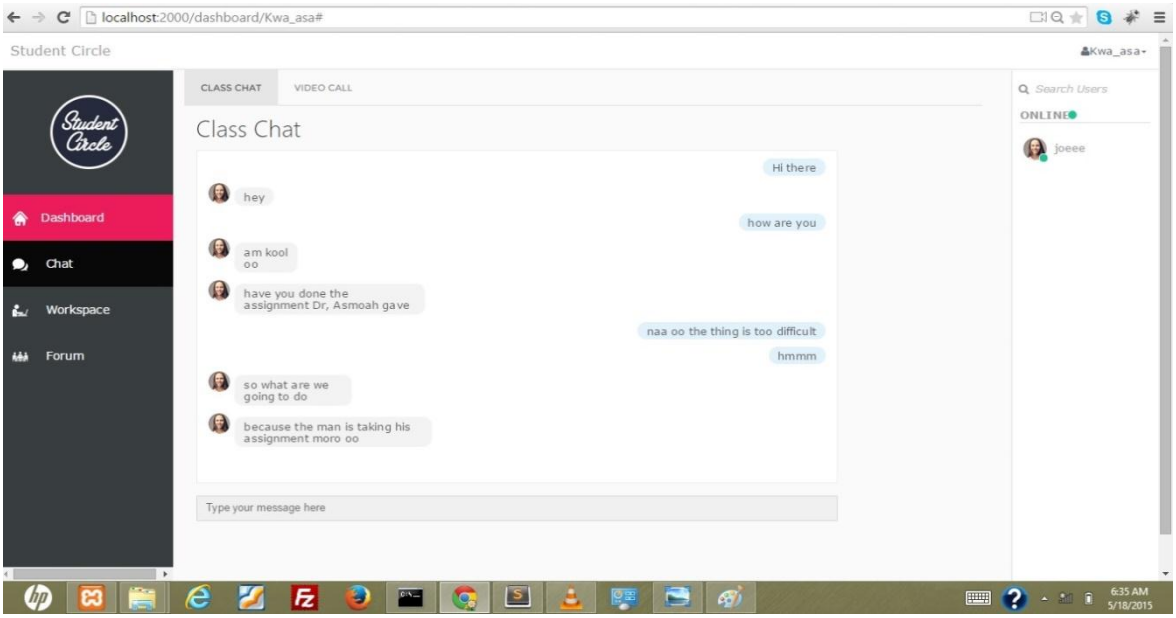


Figure 9: Group Chat

This page shows multiple users chatting in a group created.

4.2IMPLEMENTATION OF SOLUTION

This is a realization of a technical specification or algorithm as a program, software component through computer programming and deployment. At this stage the system constructed is implemented to solve the problems in the existing systems. Also we convert the system specification into an executable system.

4.3 MAPPING LOGICAL DESIGN UNTO PHYSICAL PLATFORM

Implementation may involve developing programs in high- or low-level programming languages or tailoring and adapting generic, off-the-shelf systems to meet the specific requirements of an organization.

This stage focuses on mapping the various logical modules unto the physical modules. The logical design represents the inherent nature of the system and only changes if the system changes, which is less volatile.

The physical implementation is ultimately based on available technology and as such changes dynamically and is only necessary to serve the business need.

For the Student Circle, since it is a web application, it is required that the user has the latest version of the web browser installed that is supported by the frame work being used. Such as Mozilla Firefox, Google Chrome or Internet Explorer. It runs a server side that is handled by node.js.

4.4 CONSTRUCTION

The system was constructed using the following development tools

- Node.js command prompt
- Webcams and Computer headphones, were used to enable the video and audio functions
- My sql

- Sublime text
- Express framework
- Web RTC
- Socket IO

4.5 SYSTEMATIC AND THOROUGH TESTING.

Testing is very essential in software development. It enables the developer know if the requirements have been met as well as if there are any errors in the system. When software is tested, it executes a program using artificial data. You check the results of the test run for errors, anomalies, or information about the programme's non-functional attributes.

The system, Student Circle went through a number of tests.

Unit testing, where individual program units or object classes were tested. Unit testing focused on testing the functionality of objects or methods.

Component testing, where several individual units were integrated to create composite components. Component testing focused on testing component interfaces.

System testing, all of the components in a system were integrated and the system was tested as a whole. System testing focused on testing component interactions.

System testing checks that components are compatible interact correctly and transfer the right data at the right time across their interfaces.

Beta testing, here a number of users (computer science students, 2015 batch) were made to use the system with the aim of identifying and fixing errors.

CHAPTER FIVE

SYSTEM EVALUATION AND CONCLUSION

5.1 PROJECT EVALUATION

This chapter provides a brief overview of the extent to which the project has been a success in relation to the stated requirements. It is the systematic and objective assessment of an ongoing or completed project, the aim of which is to determine the relevance and level of achievement of project objectives, development, effectiveness, efficiency, impact and sustainability. It throws more light on the entire architecture of the project, how effective the project has been ,how it has solved the problems proposed in the previous chapters and the methodologies used in tackling them.

5.1.1 EVALUATING THE PROJECT

The evaluation approach enables the assessment of the worth of the project and identifying areas for improvement.

In comparing the proposed project to the existing project, I can say the project is successful as all the proposed features have been met, even with additional functionalities.

The creation of an intuitive graphical user interface for easy usage of the software has also been achieved successfully. This proposed system apart from being easy to deploy also provides all these functionalities in one simple system which is non-existent. Also with the advent of android OS, there is a feature that makes use of that.

I can confidently say that Student Circle has been a success as the aims and objectives have been met with respect to the stated functional and non-functional requirements.

The system efficiency, effectiveness, security, availability and scalability are some of the system objectives that have been met.

5.1.2 EVALUATING THE SOLUTION

The system can easily be accessed by having internet access and a web browser installed. The software has been developed to be compatible with windows operating system as well as other operating systems like Mac and Linux. Security issues have also been handled through proper authentication and the role of HTTP.

5.1.3 EVALUATING THE METHODOLOGY

The methodology used here is the incremental approach .This enables changes be made to the functional and non-functional requirements if need be in the course of development. The system is developed as a series of versions

(Increments), with each version adding more functionality to the previous version. The use of the incremental method provided the following benefits:

- The cost of accommodating changing customer requirements is reduced. The amount of analysis and documentation that has to be redone is much less.

- It is easier to get customer feedback on the development work that has been done. Customers can comment on demonstrations of the software and see how much has been implemented. Customers find it difficult to judge progress from software design documents.

- More rapid delivery and deployment of useful software to the customer is possible, even if all of the functionality has not been included. Customers are able to use and gain value from the software early.

In all the incremental approach was best suited for the type of system we developed and provided the needed effect.

5.2 SUMMARY OF PROBLEMS

The software Student Circle has the following limitations as follows;

1. The use of webrtc made the application browser dependent. Some of its api was not supported by some browses like Firefox (Voice recognition API), Safari and Opera mini.

2. Another limitation is limited online resources for node and webrtc since it is a new way for creating web apps.

3. Lack of a standard library: JavaScript is a language with a beautiful core, but an absolutely anemic standard library -- things that you would take for granted as part of other server-side language installs will simply not be there.

4. Most functionalities such as adding sessions, file upload requires that you download modules. These modules ended up increasing the size of the application.

5. The application was a one page application and in real world will take some time to load since it has to load all its resources on the go.

6. The application was developed using JavaScript and will fail to run in browsers that have JavaScript disabled.

5.3 SUMMARY OF ACHIEVEMENTS

A lot has been achieved through the development of this project. They include:

1. Using voice recognition to dictate notes.

2. Limiting the use of server since most commutation between users was done peer to peer without the server sending or receiving request or response respectively.

3. Being able to create an environment which enhanced students to students learning

4. Being able to create an application which works by being real time. Since real time is all that is needed in communication.

5.4 SUMMARY OF CHALLENGES

The challenges faced during the development of the project include;

1. Lack of sufficient resources online since the language used in developing the software is a new language.

2. Time constraint in developing the application, as the language was new coupled with the limited time frame for development.

5.4 FUTURE DEVELOPMENTS

Despite the successes there is still room for improvement. This could be in the following ways:

1. Future development of the application with the same language should be developed in such a way that is not browse dependent.
2. The application should be built to prevent it from being a one page app. Placing other functionalities on different pages will facilitate a fast load time of the app.
3. The Application should be developed to be able to detect if browse's JavaScript has been disabled then alert user to enable it or show user how to enable it.
4. The high dependence on NPM modules should be minimized, to help decrease the general size of the app.
5. The application should be built to include more functionalities that will help make student communication and academics easier.

5.5 RECOMMENDATION

A further improvement of the Student Circle is highly recommended. Hence incorporating the above mentioned future developments will further improve the system. It is recommended that the system be employed in the various tertiary institutions to improve teaching and learning, especially in the distance learning sector.

5.6 CONCLUSION

Student circle is web based chatting application tailored to enhance teaching and learning for tertiary students. It provides a cost effective means of communication. It transcribes audio messages into words.

It incorporated the use of node.js, web RTC, and Socket IO, which is a modern technology in web development.

Finally it provides a window of opportunity to improve distance learning.

REFERENCES

[1] Software engineering ,9th edition by I.Sommerville, Pearsons, 2011, page 83,Page 87.

[2] Jump Start Node.js by Don Nguyen,1stEdition, page 4.

[3] Building the Realtime user Experience by Teb Ronden 1st Edition, page 2.

[4] Socket.io Real-time Web application development by Rohit Rai 1st Edition, page 16.

[5] www.Nodejs.com

[6] www.Socketio.com

[7] www.wikipedia.com

APPENDIX

Student Circle: This is the code name of the project

WebRTC: It is an open source project aiming to enable the web with real time Communications capabilities via simple APIs

Whatsapp, Facebook, Viber and Skype: These are social networks found on the internet and perform text, audio and video media functions.

API : It is abbreviation of application Programming Interface, is a set of routines, protocols, and tools for building software applications by specifying how the software components should interact.

Web Server: These are computers that deliver web pages

HTTP: Hypertext Transfer Protocol is an application protocol for distributed, collaborative hypermedia information systems.

Web Socket: This is a protocol providing full duplex communication channels over a single TCP connection.

Spoofing: This is the process of creating Internet Protocol packets with a source IP address, with the purpose of concealing the identity of the sender or impersonating another computer system

Distance learning: This is a mode of delivering education and instruction often on an individual basis, to students who are not physically present in a traditional setting such as a classroom.

Voice recognition: This is the translation of spoken words into text.

Man in the middle attack: This is an attack where the attacker secretly relays and possibly alters the communication between two parties who believe they are directly communicating with each other.