

# GOOGLE APPS FOR EDUCATION

## CASE STUDY: KAMPALA INTERNATIONAL UNIVERSITY

By

BIT/31737/102/DU

Kibet Lonyakou Jonas

Email: jonas.kibet@stud.kiu.ac.ug

&

BIT/31655/102/DU

Joshua Eyaru

Email: joshua.eyaru@stud.kiu.ac.ug

A Project Report Submitted to the  
College of Applied Science and Technology,  
In partial fulfillment of the requirements for the award of  
Bachelors Degree in Information Technology of  
Kampala International University

September, 2013

### Declaration

I Kibet Lonyakou Jonas & I Joshua Eyaru, declare that this is our original work and has not been submitted to any University for any award of a bachelor's degree in Information Technology.

Kibet Lonyakou Jonas  
BIT/31737/102/DU

Sign.....

Date.....

Joshua Eyaru  
BIT/31655/102/DU

Sign: .....

Date: .....

### Approval

This graduation proposal has been submitted with the approval of the following supervisor.

Dr. Mumtaz Sohag

Sign: .....

Date: .....

### **Dedication**

We sincerely dedicate this project to our parents, the Google University Programs Team Sub-Saharan Africa, Google Student Ambassadors Sub-Saharan Africa 2012 -2013, Google Club of Kampala International University 2012 -2013, and the Computer Studies Graduating Class of 2013

### **Acknowledgement**

We would like to give thanks to God, for giving us the grace to undertake this project successfully until we have completed it as planned.

We take this opportunity to recognize the support offered to us by the following individuals; Our Supervisor Dr. Mumtaz Sohag Deputy Principal Research and publications Kampala International University who kept encouraging and directing us all through the project, Mr. Cyrus Barigye the ICT Director Kampala International University for giving us the platform to do numerous testing of our project and availing us with all the necessary materials we needed during this project. We also recognize Mr. Emmanuel Lusenji Web Admin Kampala International University for his endless support and technical consultation towards the success of this project.

We extend our sincere gratitude to our fellow coursemates for their encouragement during the early stages of this project.

We sincerely thank the JEMMO Group for their support, not to forget the class coordinator Ms. Katutu Peace for her regular coordination.

Last but not least, we would like to thank Lonyakou's Family and Eyaru's Family for their financial, emotional, spiritual and social support throughout the development and implementation of this project, God will always reward you.

## List of Figures

<b>Figure 4.1</b> how cloud computing works .....	20
<b>Figure 4.2</b> Cloud Computing Architectural Framework “how it works” .....	21
<b>Figure 4.3</b> a use case diagram of campus email system .....	27
<b>Figure 4.4</b> a use case diagram of campus collaboration system .....	27
<b>Figure 4.5</b> sequence diagram showing how a user logs into a mail system .....	28
<b>Figure 4.6</b> campus email system login interface conceptual design .....	29
<b>Figure 5.1</b> showing access to the system with a student domain .....	30
<b>Figure 5.2</b> showing the user log in panel .....	31
<b>Figure 5.3</b> Showing the authentication of the user .....	31
<b>Figure 5.4</b> showing the users inbox section after authentication .....	32
<b>Figure 5.5</b> showing Google Drive, customized to KIU for cloud storage and sharing of information	32
<b>Figure 5.6</b> customized sites for the school domain .....	33
<b>Figure 5.7</b> a Calendar with one day view .....	33
<b>Figure 5.8</b> showing a week’s view of the calendar .....	34
<b>Figure 5.9</b> showing a month view of calendar .....	34
<b>Figure 5.10</b> showing the display of users active on the domain .....	35
<b>Figure 5.11</b> showing the domain setup page where services allowed to use are activated from ...	35
<b>Figure 5.12</b> showing the security management tools for the email accounts .....	36
<b>Figure 5.13</b> showing the reports section about the users interaction in a given time frame .....	36

## **List of Tables**

<b>Table 4.1</b> Use case 1 accessing campus mail system .....	26
<b>Table 4.2</b> Use case 2 accessing collaboration system .....	26
<b>Table 5.1</b> Showing Use case testing of the campus email system .....	37
<b>Appendix: Table 1</b> .....	47
<b>Appendix: Table 2</b> .....	48

**List of Appendices**

**Appendix A: Interview Guide ..... 46**

**Appendix B: Project Time Frame ..... 47**

**Appendix C: Budget ..... 48**



## TABLE OF CONTENTS

Declaration.....	i
Approval.....	ii
Dedication .....	iii
Acknowledgement .....	iv
List of Figures .....	v
List of Tables .....	vi
List of Appendices .....	vii
CHAPTER ONE .....	1
1.1 Introduction.....	1
1.2 Background Information .....	1
1.3 Statement of the problem .....	1
1.4 Objectives.....	1
1.4.1 Main objective: .....	1
1.4.2 Specific objective .....	2
1.5 Scope of the project.....	2
1.5.1 System Scope .....	2
1.5.2 Geographical Scope .....	2
1.6 Significance of the project .....	3
CHAPTER 2.....	5
LITERATURE REVIEW .....	5
2.0 Introduction .....	5
2.1 Information System .....	5
2.1.1 Types of Information System.....	6
Transaction Processing Systems .....	6
Management Information Systems .....	6
Decision Support Systems.....	6
2.2 Management Information System.....	7
2.3 Historical Perspectives .....	7
2.3.1 The “effect” paradigm. ....	8

2.3.2	The “conditions” paradigm.....	8
2.3.3	The “interactions” paradigm.....	9
2.3.4	The “computer supported” paradigm.....	9
2.4	Relationship to Other Concepts .....	10
2.5	Development of Collaboration Skills .....	10
2.6	Challenges in Collaboration .....	11
2.7	Future Prospects.....	11
CHAPTER THREE .....		13
METHODOLOGY .....		13
3.1	Introduction .....	13
3.1.1	Waterfall Model.....	13
3.1.2	Weakness of waterfall model .....	13
3.1.3	Agile Methodology.....	13
3.1.4	Comparison between Agile and Waterfall.....	14
3.1.5	Reasons for using Agile Methodology .....	14
3.2	Google Apps Deployment Methodology .....	15
3.3	Mobile Application Development Methodology V3.....	15
3.4.1	Observation.....	16
3.4.2	Brain-Storming Sessions .....	16
3.4.3	Captured System Requirements .....	16
3.5	System Design Techniques.....	16
3.5.1	Storyboarding .....	17
3.5.2	UML techniques .....	17
3.6	Design.....	17
3.6.1	User Interface Overview .....	17
UI Element Description .....		17
3.7	Application Building .....	18
3.7.1	Programming Technique Used .....	18
3.8	Application Testing Techniques .....	18

3.8.1 White box Testing .....	18
3.8.2 Blackbox Testing .....	18
3.9 Conclusion .....	19
CHAPTER 4.....	20
SYSTEM REQUIREMENT ANALYSIS AND DESIGN .....	20
4.1 System Architecture.....	20
<b>Figure 4.1 how cloud computing works .....</b>	<b>20</b>
4.1.1 Understand the Architecture.....	21
4.1.2 App container model .....	21
4.1.3 Programming model .....	21
4.1.4 App lifecycle at a glance.....	22
4.1.5 Security model .....	22
4.2 System Requirements .....	22
4.2.1 Functional Requirements.....	22
4.2.2 Non-Functional Requirements.....	23
4.2.3 Hardware Requirements.....	23
4.2.4 Software Requirements .....	24
Platform .....	24
Languages .....	24
Browsers .....	24
Desktop .....	25
Mobile and tablet.....	25
4.3 Google Apps Use Case diagram.....	26
4.3.1 Explanation of the above use cases.....	28
4.4 Sequence Diagram of how to log into campus email system .....	28
4.6 User Interface Design .....	29
4.7 Summary .....	29
CHAPTER 5.....	30
THE CAMPUS EMAIL SYSTEM.....	30

5.1 Screen shots for Campus Email system user interface.....	30
5.1.1 Screen shots of our project interface .....	30
Login panel for students .....	31
User authentication .....	31
Drive .....	32
Customized sites to School domain .....	33
Calendars .....	33
One day view calendar.....	33
A Weeks view calendar .....	34
A month view of calendar.....	34
5.2 Screen shots Email system admin panel.....	35
Domain users display page .....	35
Setup page .....	35
Security settings page .....	36
Reports page .....	36
5.3 Use Case Test Results.....	37
5.4 Requirement Test Results.....	37
CHAPTER 6.....	39
CONCLUSION AND RECOMMENDATIONS .....	39
6.0 Introduction.....	39
6.1 Achievements.....	39
6.2 Limitations.....	39
6.3 Challenges .....	40
6.2 Conclusions.....	40
6.3 Recommendations .....	40
REFERENCES .....	42
APPENDICES .....	46

## **CHAPTER ONE**

### **1.1 Introduction**

Google apps for education is a free suit of hosted email and collaboration applications, exclusively for schools and universities that leverages the power of internet and cloud computing. This offers institutions and organizations cutting edge software that is used to create customized email accounts so as to integrate communication systems within institutions and organizations to allow smooth, timely and fast convergence of information as compared to old methods of information communication that exists currently in Kampala International University.

### **1.2 Background Information**

Communication is the major element that creates unity among various organs of any organization or entity in order for it to realize a common goal. Many institutions around the world have been faced by this eminent factor and it has led to their operations taking longer to commence and also lead to unclear breakdown of responsibilities and tasks to be undertaken taking long to effect due to its slow communication convergence within various organs of the organization

### **1.3 Statement of the problem**

The major cause for the researchers' to come up with project is the fact that Kampala International University has for long not been able to have a good communication system that can promote collaboration between the staff and students.

Due to the above problems, it is better for both the staff and students to collaborate through the Google Apps for Education so as to help improve communication between staff and students.

### **1.4 Objectives**

The main idea behind this application is come up with a system that will aid collaboration between the staff and the students in the area of academics at Kampala International University.

#### **1.4.1 Main objective:**

The main objective is to self-deploy, migrate staff emails and create student emails to the Google Apps for education platform which supports cloud computing. This will consist of both staff and student domain targeting on collaboration between them.

#### **1.4.2 Specific objective**

The following are the list of objectives used by researchers' to produce this system:

- To come up with a system that will aid collaboration between the staff and students of Kampala International University.
- To come up with a system that will integrate with other systems to facilitate the online result system
- To come up with a system that will integrate with other systems to facilitate the online clearance system.
- To come up with a system that will aid the students registration.
- To come up with a system that will promote e-Learning at Kampala International University using the concept termed as "virtual classrooms".

### **1.5 Scope of the project**

#### **1.5.1 System Scope**

The system scope of this project was defined by having a staff domain and a student domain with the following features as deliverables

- Ability to allow students to communicate to any staff member of the University without any hindrance.
- Ability to allow both students and staff to search for any contact of any registered person on the two domains.
- Ability to get daily updates the University to both students and staff through the system administrator.
- Ability to allow student to get results online.
- Ability to allow students to get their clearance ledgers online.

#### **1.5.2 Geographical Scope**

The geographical scope of this project was defined by having a staff and student domain with the following coverage features.

- Ability to allow both staff and students to access information related to the University any where in the world through internet access.

- Ability to allow the system administrator to administer the system anywhere in the world.
- Ability for users of the system to save their data online “cloud computing”.

### 1.6 Significance of the project

Cloud computing is triggering a fundamental shift in the way people experience computing, before people used different kind of storage medias to store data and perform collaboration within different departments and branches of the organization which were limited to only a few functionalities. Today, cloud computing makes it easier to integrate all the different functionalities ranging from storage of data to collaboration between in an organization.

These days Email systems can give place to various applications coming from various genres.

These include;

Social networking applications like Google+, facebook, twitter.

Chatting applications like gtalk, msn, yahoo chat.

Data collection application from users like use of Google forms embedded in the Email system.

These applications carry a lot of importance too as they have opened a new dimension of communication as well as new dimension in accessing and supplying information.

The significances of this project are as follow:

1. Ability to share information by students and staff.
  - Students and staff of Kampala International University will use Google documents embedded in the KIU Google apps for education to share the same document. This can give them the ability to edit the same document which they are sharing from anywhere in the world.
2. Flip classrooms and e-Learning.
  - Students can now have flip classrooms where they can have an advance lecture streamed online using the Google+ application and the lecturer only shows up to make few remarks and respond to questions.  
The e-Learning ability can also be supported since all students have the user accounts.

3. Data collection.

- The Heads of departments can also use the embedded forms to collect student's information like semester registration.
- The forms can also be used to apply to the University by new students and any kind of application including applications for new guild union candidates.

4. Free SMS.

- The system has the ability for users to have free sms from via the gtalk to supported mobile services.

5. Video conferencing.

- The staff and student's can use the video conferencing function "hangouts" to conduct online meetings and virtualizing classes.

6. Student's documents.

- Different staff ranging from administrative to teaching staff can use the Email system to deliver student's documents like finance clearance, semesters results and class test mark.

7. Online storage.

- Students and staff are able to backup their data online using the Drive feature which enables them to store lecture notes, results, ledgers, certificates, transcripts, photos and any kind of important document. These documents can be retrieved from anywhere in the world by the owner.

8. Online presentations.

With this system, no more virus stress on the side of students and staff who intend to go for a presentation, the system can support online presentations where you store your presentation and retrieve it from anywhere and carry on with the presentation without the need of moving with it in flash disk or laptop.



## CHAPTER 2

### LITERATURE REVIEW

#### 2.0 Introduction

##### Theoretical Perspectives

Collaborative learning is broadly defined as “a situation in which two or more people learn or attempt to learn something together,” and more specifically as joint problem solving (Dillenbourg, 1999, p. 1). Roschelle and Teasley define collaboration more specifically as “mutual engagement of participants in a coordinated effort to solve a problem together,” (as cited in Dillenbourg et al., 1996, p. 2).

In recent years the market has seen a rapid development in document collaboration tools. Primitive document collaboration used email, whereby comments would be written in the email with the document attached. The problem was that this was not a document-centric solution.

Today, the best document collaboration tools are more document-centric. These systems provide a user with a document-centric collaboration experience because they allow users to tag the document and add content specific comments, maintaining a complete version history and records and storing all comments and activities associated around a document. For this reason, an increasing number of firms are using email less and file sharing and document collaboration tools more.

This chapter provides critical review on tools and systems set up to help multiple people work together on a single document or file to achieve a single final version of that desired document or file.

Based on the several literature review conducted, several key entities that affect collaboration of information systems have been identified

#### 2.1 Information System

Information systems are the software and hardware systems that support data-intensive applications such as data transformation of information into knowledge and desired by an individual or group users in organization.

The journal Information Systems publishes articles concerning the design and implementation of languages.

Information systems are implemented within an organization for the purpose of improving the effectiveness and efficiency of that organization.

### **2.1.1 Types of Information System**

1. Transaction processing system
2. Management Information System
3. Decision Support System

#### **Transaction Processing Systems**

TPS processes business transaction of the organization. Transaction can be any activity of the organization. Transactions differ from organization to organization. For example, take a railway reservation system. Booking, canceling, etc are all transactions. Any query made to it is a transaction. However, there are some transactions, which are common to almost all organizations. Like employee new employee, maintaining their leave status, maintaining employee's accounts, etc.

This provides high speed and accurate processing of record keeping of basic operational processes. These include calculation, storage and retrieval.

Transaction processing systems provide speed and accuracy, and can be programmed to follow routines functions of the organization.

#### **Management Information Systems**

These systems assist lower management in problem solving and making decisions. They use the results of transaction processing and some other information also. It is a set of information processing functions. It should handle queries as quickly as they arrive.

#### **Decision Support Systems**

These systems assist higher management to make long term decisions. These type of systems handle unstructured or semi structured decisions. A decision is considered unstructured if there are no clear procedures for making the decision and if not all the factors to be considered in the decision can be readily identified in advance.

These are not of recurring nature. Some recur infrequently or occur only once. A decision support system must be very flexible. The user should be able to produce customized reports by giving particular data and format specific to particular situations.

## **2.2 Management Information System**

Management Information System (M.I.S.) is basically concerned with processing data into information, which is then communicated to the various Departments in an organization for appropriate decision-making.

## **2.3 Historical Perspectives**

Much of the research on collaborative and cooperative learning is rooted in the work of Piaget and Vygotsky (Dillenbourg et al., 1996). For example, socio-constructivists borrow Piaget's system of developmental stages describing children's cognitive progress, as well as ideas related to cognitive conflict, which refers to the sense of dissonance experienced when one becomes aware of a discrepancy between one's existing cognitive framework and new information or experiences.

According to the socio-constructivist approach, cognitive conflict is critical in triggering growth. Social interactions help to facilitate such conflict to the extent that students interact with peers at more advanced developmental levels. Within this school of thought, group heterogeneity is an important consideration, as group mates are expected to possess different knowledge, different knowledge representation schemes, and different reasoning mechanisms (as reviewed in Dillenbourg et al., 1996).

Since the late 1990s, a new strand of research on collaborative learning focusing on new technologies for mediating, observing, and recording interactions during collaboration has emerged (Kreijns et al., 2003). This new strand of research, commonly called computer-supported collaborative learning (CSCL), typically uses online networks for facilitating and recording online interactions among two or more individuals who may be geographically and/or temporally dispersed.

Much of this research has grown in parallel to new technologies for supporting distance interactions, such as email, chat, instant-messaging capability and more recently, resources for synchronous video conferencing (such as Skype).

These different historical perspectives have led to different research paradigms. For example, Dillenbourg et al. (1996) characterize research stemming from the Piagetian, Vygotskian, and shared cognition approaches as the "effect" paradigm, the "conditions" paradigm, and the "interactions" paradigm, respectively. The latest variant of the interactions paradigm might be called the "computer-supported" paradigm. Each of these paradigms is explored separately below.

### **2.3.1 The “effect” paradigm.**

Those working in the “effect” paradigm tend to examine outcomes of collaboration rather than the collaborative process itself, comparing group performance with individual performance. Research in this tradition suggests that collaborative classroom structures can have powerful effects on student learning and performance. For example, seventh-grade students working in groups of 3–4 on computational math problems earned significantly higher scores working in groups than equivalent-ability students working individually (Webb, 1993).

Saner et al. (1994) report the results of a study that administered hands-on science tasks to students in grades 5 and 8. Students completed the first portion, answering short-answer questions about relevant content, individually (Part 1).

The second portion of the task included hands-on science activities, such as observing, recording data, and carrying out experiments. This portion was completed in pairs (Part 2). The third portion of the task was again completed individually, and entailed interpreting results obtained from the group portion of the task and applying those results to an unfamiliar context (Part 3). Saner et al. (1994) conclude that at both grade levels, higher- and lower-ability students were affected differently by the collaboration.

Higher ability students who performed well as individuals on Part 1 tended to perform similarly well on Part 3. In addition, the best predictor of Part 3 scores for higher-ability students was their own score on Part 1 of the task. In contrast, lower-ability students exhibited a carry-over effect of the collaboration on their Part 3 scores, which tended to be higher than their Part 1 scores. In fact, the best predictor of Part 3 scores for lower-ability students was their score on Part 2. Thus, collaboration tends to benefit lower-ability students, whereas there appears to be no carry-over effect for higher-ability students.

### **2.3.2 The “conditions” paradigm.**

Researches working in the “conditions” paradigm generally attempt to determine the conditions moderating the effectiveness of collaboration on learning, such as individual characteristics of group members, group heterogeneity and size, and task features (Dillenbourg et al., 1996). For example, Webb (1991) found significant differences in the collaborative learning experiences of boys and girls. Boys were more likely than girls to give and receive elaborated explanations, and their explanations were more likely to be accepted by group

mates than girls' explanations. Boys were also more likely to receive responses to requests for help, perhaps because they asked direct and specific questions more frequently.

### **2.3.3 The "interactions" paradigm.**

The "interactions" paradigm developed in response to the complexities associated with the former paradigm and attempts to identify mediating mechanisms between collaboration and learning outcomes (Dillenbourg et al., 1996). In particular, this strand of research attempts to isolate characteristics and processes of interactions through which collaboration effects learning. For example, one proposed mediator of the effect of collaboration on learning is the extent to which social interactions produce elaborated explanations (Dillenbourg et al., 1996).

Mercer (1996) argues that interactions producing elaborated explanations enable students to learn the principles underlying practical procedures and strategies, which can result in learning that is more generalizable and transferrable to new situations.

### **2.3.4 The "computer supported" paradigm.**

Finally, research emerging from the computer-supported collaborative learning paradigm has generally attempted to determine whether the theoretical benefits of collaborative learning in face-to-face settings can be realized through computer-mediated or computer-assisted interactions that are limited to asynchronous, text-based interactions. For example, Kreijns et al. (2003) caution that, in contrast to face-to-face interactions, social interaction should not be taken for granted just because the technology to support interaction is there. With respect to asynchronous, text-based interactions, research suggests that there are a few differences between face-to-face collaborations and those conducted at a distance.

First, certain interaction patterns may be more or less evident in one medium versus the other. For example, Curtis and Lawson (2001) found that in the online medium, there were fewer exchanges in which students challenged one another and more exchanges related to planning. The authors speculate that because students did not know one another prior to interacting, they may have felt less comfortable challenging the ideas of others.

In addition, the online medium appeared to have made planning activities more important for coordinating work. Despite these differences between traditional and technology-enabled collaboration, the authors conclude that successful student collaboration in an online medium is possible.

## **2.4 Relationship to Other Concepts**

Collaborative learning approaches are related to a number of other, so-called 21st century skills, including critical thinking, metacognition, and motivation. For example, a number of researchers have linked collaborative learning to the development of critical thinking (Bailin et al., 1999; Bonk & Smith, 1998; Heyman, 2008; Nelson, 1994; Paul, 1992; Thayer-Bacon, 2000). Definitions of critical thinking vary widely, but common elements of most definitions include the following component skills:

- Analyzing arguments (Ennis, 1985; Facione, 1990; Halpern, 1998; Paul, 1992);
- Making inferences using inductive or deductive reasoning (Ennis, 1985; Willingham, 2007; Paul, 1992; Facione, 1990);
- Judging or evaluating (Case, 2005; Ennis, 1985, Facione, 1990; Lipman, 1988; Tindal & Nolet, 1995); and
- Making decisions or solving problems (Ennis, 1985; Halpern, 1998; Willingham, 2007).

## **2.5 Development of Collaboration Skills**

This section reviews the limited theoretical and empirical literature on development of collaborative capacities, including how collaboration skills first appear and develop over time.

According to Tudge (1992), early work by Piaget and Vygotsky is informative with respect to development of collaboration skills in young children. From a Piagetian perspective, children younger than 7 may lack the developmental skills to benefit from collaboration because they have not reached the concrete operational stage, or the stage at which logical reasoning first appears and children begin to apply mental operations to concrete problems such as conservation tasks. Once children have attained this stage, however, they appear to benefit from collaboration.

Dillenbourg et al. (1996) similarly observe that Piaget's theory leads to specific expectations for development of collaboration skills. For example, pre-operational children may lack the ability to benefit from collaboration because they cannot de-center from their own perspective, suggesting they may have difficulty recognizing the views of others. Similarly, preschool-age children may lack the ability to sustain discussions of alternative hypotheses.

## **2.6 Challenges in Collaboration**

Collaborative learning, because it occurs in a group, can obscure individual contributions, making it difficult to isolate individual student scores. Often teachers will assign a single score to a group based on completion of a group product, and this group score in turn is assigned to each individual group member. However, to the extent that individual student contributions are not comparable or equivalent, this does not necessarily reflect individual effort, knowledge, or skills (Race, 2001; Saner et al., 1994; Webb, 1995).

Research shows that group-level assessments may not yield scores that are predictive of individual-level ability, even when individual students turn in separate products. In particular, scores from group work tend to over-estimate individual performance, and exhibit both ceiling effects and range restriction (Webb, 1993). Fall et al. (1997) demonstrate that the effect of collaboration on performance is evident even with very limited interaction (e.g. a 10-minute discussion regarding themes in a reading passage). Even when individual group mates turn in separate products, scores between group members tend to be correlated at very high levels (.66–.83), suggesting that they are not independent of one another (Saner et al., 1994).

In addition, there is a carry-over effect, with scores on subsequent individual work affected by previous group collaboration, particularly for low achievers (Saner et al., 1994; Webb, 1998). If this higher performance reflects real learning, then group assessment scores are not necessarily invalid. However, if students obtain higher scores from group assessments simply because their more able group mates complete work for them, then using group scores as indicators of individual student learning is problematic.

## **2.7 Future Prospects**

To date, only 5% of businesses have installed Web 2.0 collaboration applications. However, document collaboration based in the cloud looks set to grow rapidly this decade, as cloud based services begin to really take-off. Analysts give different projections for the growth of the cloud computing sector.

Frost & Sullivan predicts that the cloud industry is set to grow by 39% every year until 2015 when it will be worth \$5.8 billion.

Forrester is even more optimistic, predicting that Software as a service and Web 2.0 applications will be worth \$22 billion by the end of this year 2013.

New innovative firms are joining the industry as cloud based document collaboration gets adopted more and more by businesses around the world. It is for this reason that document collaboration has become an area of IT services industry analysts are watching for the future.



## CHAPTER THREE

### METHODOLOGY

This chapter provides an overview of the methodology for the system that aims to meet the Objectives provided for this project (see Chapter 1). The design provided here was the basis for the implementation of the system produced.

#### 3.1 Introduction

Methodology is a set or system of methods, principles, and rules for regulating a given discipline or is a procedure, technique, or way of doing something, especially in accordance with a definite plan. In the ICT field there are number of methodologies that can be used to manage projects. These methodologies can be categorized as Non Agile Methodology like Waterfall Model and Agile Methodology like FDD.

##### 3.1.1 Waterfall Model

Waterfall is a non agile methodology.

According to (Cobb, 2011) Waterfall model is a simplified representation of a software development process, presented from a specific perspective that separate and distinct phases of specification and development.

Figure 3.1 Showing diagram of waterfall Model adopted from (Cobb, 2011)

##### 3.1.2 Weakness of waterfall model

The drawback of the waterfall model is the difficulty of accommodating change after the process is underway

- This makes it difficult to respond to changing customer requirements therefore, this model is only appropriate when the requirements are well-understood

##### 3.1.3 Agile Methodology

Agile software development is a group of software development methodologies based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It promotes adaptive planning, evolutionary development and delivery, a time-boxed iterative approach, and encourages rapid and flexible response to change. It is a conceptual framework that promotes foreseen

interactions throughout the development cycle.

According to (Highsmith, 2002), he explained about different types of Agile Methodologies like:

- Agile Modeling

- Agile Unified Process (AUP)
- Dynamic Systems Development Method (DSDM)
- Essential Unified Process (EssUP)
- Extreme Programming (XP)
- Feature Driven Development (FDD)
- Open Unified Process (OpenUP)
- Scrum
- Velocity tracking

(Highsmith, 2002), defined Agile methodology as an approach to project management, typically used in software development. It helps teams respond to the unpredictability of building software through incremental, iterative work cadences, known as sprints.

#### 3.1.4 Comparison between Agile and Waterfall

Waterfall, as a project management methodology, has been criticized for not being able to cope with constant changes in software projects, and with the inability of customers to specify adequate requirements before they have tried prototypes. Also, the huge overhead of specifying requirements before starting the actual development may cause long delays of a project.

The iterative nature of agile makes it an excellent alternative when it comes to managing development projects, especially software projects. (Agile Management, 2012).

#### 3.1.5 Reasons for using Agile Methodology

Most of the methods that obey the Agile Methodology promote development, teamwork, collaboration, and process adaptability throughout the life-cycle of the project. Other reasons are as listed below:

An agile method breaks tasks into small increments with minimal planning and does not directly involve long-term planning.

Team composition in an agile project is usually cross-functional and self-organizing, without consideration for any existing corporate hierarchy or the corporate roles of team members. Team members normally take responsibility for tasks that deliver the functionality iteration requires. They decide individually how to meet iteration's requirements.

Agile methods emphasize face-to-face communication over written documents when the team is all in the same location.

### 3.2 Google Apps Deployment Methodology

There are two kinds of Google Apps deployment methodology: self deployment and guided development. Self deployment involves the app administrator determining the following options:

- Domain setup - the administrator will determine whether to have one Google Apps account for staff and students (one domain and one Google Apps instance) or have one Google Apps account for staff and students but have a separate student domain within the Google Apps account (multiple domains in one Google Apps instance) or two Google Apps accounts, one for faculty and staff and another for students (multiple Google Apps instances)
- Mail flow - the administrator will determine whether email will have dual delivery through legacy server (mail is delivered to both the legacy server and the Google servers via the legacy server) or dual delivery through Google (mail to be delivered to both the legacy server and Google server via Google servers) or split delivery through Google servers.
- User provisioning - the apps administrator has to determine how new user accounts on Google Apps will be created; either by uploading a CSV (comma separated values) file containing the list of new user accounts to be created or by setting up a Google Apps Sync Directory (GADS)
- Organisational structure - Organizational units allow the administrator to segment the user population (i.e. he can create an organization for students and another for staff). This will allow him to turn services (email, calendar, sites etc) on or off for different organizational units. Organizational units also allow for email delivery to be restricted for certain populations. The apps administrator can configure email delivery settings so that some users can only email users within the domain (and receive emails from users within the domain).

### 3.3 Mobile Application Development Methodology V3

According to (Mobinex, 2011) Mobile Application Development Methodology V3 is intended to be used as a guide for the developing mobile applications.

Mobile Application Development Methodology V3 has the agile methodology features.

If needed. In this project the researchers has used this phase to determine the users' requirement from users / community.

Most of these requirements have been obtained through Observation, Prototyping and Brainstorming sessions as requirement elicitation techniques.

#### **3.4.1 Observation**

During observation, researchers were observing some of the way the services in the university were being handled especially communication. The visited health centres include Clinics, Pharmacies and Physiotherapy centres etc. This method is useful because it doesn't waste money and time and it gives the researchers good opportunity to see only things that are relevant to the system to be produced.

researchers was observing how activities are carried out in the visited areas. The information obtained from there was used to obtain system requirements.

#### **3.4.2 Brain-Storming Sessions**

During these sessions the researchers contacted colleagues so as to get to know in details what is required in deploying of the Google apps so as to facilitate collaboration on campus.

#### **3.4.3 Captured System Requirements**

The following are some of the features that the system has. These features are the ones obtained by researchers from requirement elicitation phase:

- **Ability to share information by students and staff.**
- **Flip classrooms and e-Learning.**
- **Data collection.**
- **Free SMS.**
- **Video conferencing.**
- **Student's documents.**
- **Online storage.**
- **Online presentations.**

### **3.5 System Design Techniques**

In the system design phase, researchers used design techniques such as Storyboarding, UML, and prototyping. The explanations on how researchers used these techniques are explained in sub-sections of this section.

### 3.5.1 Storyboarding

According to (Wikipedia, Storyboarding, 2012), Storyboards are graphic organizers in the form of illustrations or images displayed in sequence for the purpose of pre-visualizing a motion picture, animation, motion graphic or interactive media sequence.

The storyboard phase tells if the interactive experience of the user interface is usable to the whole team; including the developer, designer, project manager and any other 3rd parties who are involved in the project. The storyboard is mainly a grid of the application which is going to be designed. It tells about each use case and the step-by-step process of the application.

In this project, researchers used storyboarding to come up with the logic and meaning fully navigation between screens of the pages. The illustrations of the screen navigation (storyboard) is shown in chapter 4 section 4.7

### 3.5.2 UML techniques

Researchers used a number of Unified Modeling Language techniques to model of the system.

Some of the techniques used in this project are Use Cases, Use Cases Diagrams, Sequence Diagrams, and Conceptual Model.

## 3.6 Design

In this phase researchers concentrated in the Interface look of different main features of the System.

### 3.6.1 User Interface Overview

The user needs to interact with their Google app via a number of different UI's, and as described previously this is achieved through the user of generalized user-interface descriptions. Through examination of a number of user-interface types (HTML, XML, GUI's etc...), the following components are allowed within a generalized user-interface description:

UI Element	Description
Text area	presents some text as part of the UI, which may or may not be editable
Attachment	An optional component that may be presented as part of the UI or its contents may be stored by the device displaying the UI (e.g. a graphic, sound file, text, file archive etc which can be written to disk for later viewing).

### **3.7 Application Building**

This section describe in detail the building of the system based upon the high-level design

overview (see section 3.6: System Design), the tools chosen to carry out the task, the API used, design decision that have been made and any problems that arose during the implementation of the system and their solution.

#### **3.7.1 Programming Technique Used**

In the development phase the researchers used the Object Oriented Programming (OOP) concept to develop the application. The main advantage of using OOP is in managing complexity of the codes as well as enhancing reuse of classes and object.

### **3.8 Application Testing Techniques**

Researchers used a number of testing techniques to make sure the apps are working as it was intended. Techniques such as Blackbox testing and White box testing were used.

researchers. The testing was done in the system Individual units as well as system Integrated Units.

#### **3.8.1 White box Testing**

According to (Wikipedia, White-box\_testing, 2012), White box testing is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs.

researchers used white-box testing at the unit, integration and system levels of the software testing process. Though this method of test, researchers uncovered many errors or problems.

The only thing that thing testing techniques cannot do is it might not detect unimplemented parts of the specification or missing requirements.

#### **3.8.2 Blackbox Testing**

According to (Wikipedia, Black Box Testing, 2012) ,Black-box testing is a method of software testing that tests the functionality of an application as opposed to its internal structures or workings.

During this Black Box Testing, Specific knowledge of the application's code/internal structure and programming knowledge in general is not required. Researchers used some sample testers.

Testers selected by researchers were only aware of what the software is supposed to do, but not how i.e. when they enter a certain input, they get a certain output; without being aware of how the output was produced in the first place.

### **3.9 Conclusion**

In this chapter, researchers explained the way different activities were carried out during the entire deployment of the apps from requirement elicitation, system deployment, system migration and system testing.

## CHAPTER 4

### SYSTEM REQUIREMENT ANALYSIS AND DESIGN

This Chapter in detail describes the design of the system based upon high level design overview that is conceptual design, System boundary, use cases, sequence diagrams and data modeling techniques that have been used by the researcher.

#### 4.1 System Architecture

Since this project utilizes cloud computing system, it's helpful to divide it into two sections: the front end and the back end. They connect to each other through a network, usually the Internet. The front end is the side the computer user, or client, sees. The back end is the "cloud" section of the system.

1. **Computer/Phone/Tablet e.t.c. :** The front end includes the client's computer (or computer network), phone, tablet or an ipad and the application required to access the cloud computing system
2. **Cloud:** On the back end of the system are the various computers, servers and data storage systems that create the "cloud" of computing services. In theory, a cloud computing system could include practically any computer program you can imagine, from data processing to video games. Usually, each application will have its own dedicated server.

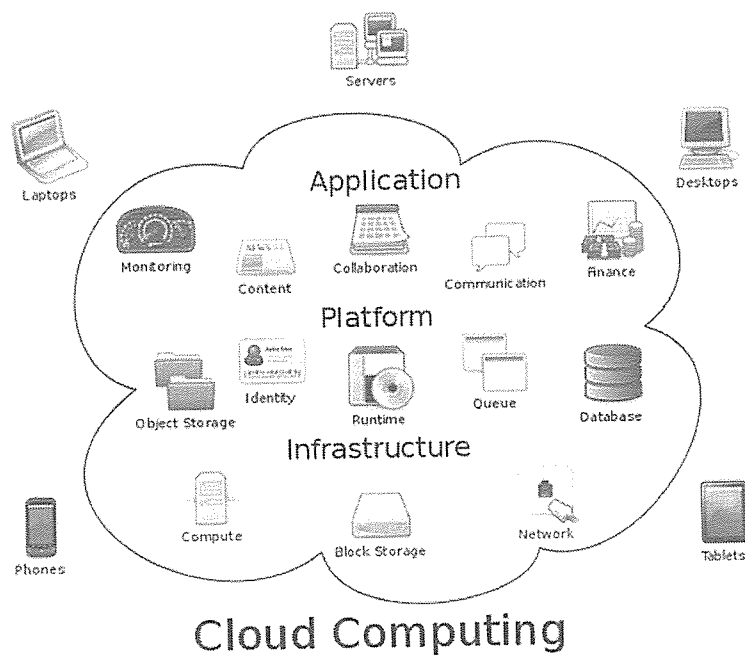


Figure 4.1 how cloud computing works



4.1.1 Understand the Architecture

Packaged apps integrate closely with a user’s operating system. They are designed to be run outside of a browser tab, to run robustly in offline and poor connectivity scenarios and to have far more powerful capabilities than are available in a typical web browsing environment. The app container, programming, and security models support these packaged app requirements.

4.1.2 App container model

The app container describes the visual appearance and loading behavior of packaged apps. Packaged apps look different than traditional web apps because the app container does not show any traditional web page UI controls; it simply contains a blank rectangular area. This allows an app to blend with “native” apps on the system, and it prevents the user from “messing” with the app logic by manually changing the URL.

Packaged apps are loaded differently than web apps. Both load the same type of content: HTML documents with CSS and JavaScript; however, a packaged app is loaded in the app container, not in the browser tab. Also, the app container must load the main document of the packaged app from a local source. This forces all packaged apps to be at least minimally functional when offline and it provides a place to enforce stricter security measures.

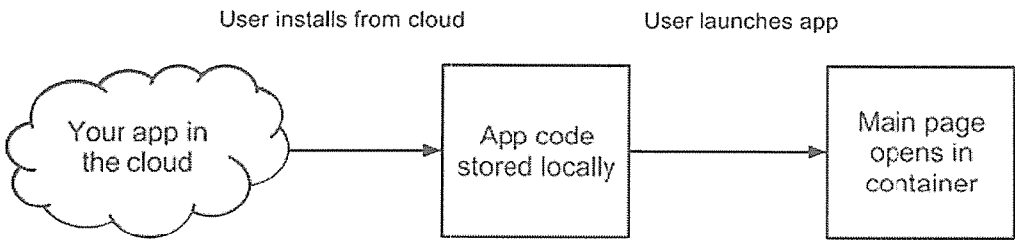


Figure 4.2 Cloud Computing Architectural Framework “how it works”

4.1.3 Programming model

The programming model describes the lifecycle and window behavior of packaged apps. Similar to native apps, the goal of this programming model is to give users and their systems full control over the app lifecycle. The packaged app lifecycle should be independent of browser window behavior or a network connection.

The “event page” manages the packaged app lifecycle by responding to user gestures and system events. This page is invisible, only exists in the background, and can be closed automatically by the system runtime. It controls how windows open and close and when the app is started or terminated. There can only be one “event page” for a packaged app.

#### 4.1.4 App lifecycle at a glance

Stage	Summary
Installation	User chooses to install the app and explicitly accepts the permissions.
Startup	The event page is loaded, the 'launch' event fires, and app pages open in windows. You create the windows that your app requires, how they look, and how they communicate with the event page and with other windows.
Termination	User can terminate apps at any time and app can be quickly restored to previous state. Stashing data protects against data loss.
Update	Apps can be updated at any time; however, the code that a packaged app is running cannot change during a startup/termination cycle.
Uninstallation	User can actively uninstall apps. When uninstalled, no executing code or private data is left behind.

#### 4.1.5 Security model

The packaged apps security model protects users by ensuring their information is managed in a safe and secure manner. Comply with CSP includes detailed information on how to comply with content security policy. This policy blocks dangerous scripting reducing cross-site scripting bugs and protecting users against man-in-the-middle attacks.

Loading the packaged app main page locally provides a place to enforce stricter security than the web. Like Chrome extensions, users must explicitly agree to trust the packaged app on install; they grant the app permission to access and use their data. Each API that your app uses will have its own permission. The packaged apps security model also provides the ability to set up privilege separation on a per window basis. This allows you to minimize the code in your app that has access to dangerous APIs, while still getting to use them.

Packaged apps reuse Chrome extension process isolation, and take this a step further by isolating storage and external content. Each app has its own private storage area and can't access the storage of another app or personal data (such as cookies) for websites that you use in your browser. All external processes are isolated from the app. Since iframes run in the same process as the surrounding page, they can only be used to load other app pages. You can use the object tag to embed external content; this content runs in a separate process from the app.

### 4.2 System Requirements

Given the objectives specified in chapter one section 1.4.2 a number of requirements for this project were captured. With regard to the design, implementation and functionality of this project, the list of requirements obtained were categorized into Functional Requirements and Non-Functional Requirements as shown in section 4.2.1 and section 4.2.2 respectively.

#### 4.2.1 Functional Requirements

The following are the Functional requirements obtained by researcher. These are the requirements that have been used to develop the final system.

- i) To have a system that will allow user to register to the system and obtain email address that will enable the user access other services within the system
- ii) To have a system that will allow a user to view his/her information that has been entered into the system and can allow editing of the information incase of erroneous information
- iii) To come up with a system that will aid collaboration between the staff and students of Kampala International University.
- iv) To come up with a system that will promote e-Learning at Kampala International University using the concept termed as “virtual classrooms”.
- v) To come up with a system that will integrate with other systems to facilitate the online result system
- vi) To come up with a system that will aid collaboration between the staff and students of Kampala International University.
- vii) To come up with a system that will integrate with other systems to facilitate the online clearance system.

#### **4.2.2 Non-Functional Requirements**

A number of Non-Functional Requirements included in the system are as listed below:

- i) Ability to validate data before submitted to the server
- ii) Easy of data filling techniques using features such us date picker, drop down list and auto complete lists.
- iii) Ability to notify user in case of anything (Success / Failure / unexpected situation)
- iv) Easy of Navigation from screen to screen
- v) User Friendly
- vi) Portable in Web application. Platform free. Run on various operating systems.

#### **4.2.3 Hardware Requirements**

The system requires running and performing of its required tasks properly should be on hardware facilities. Therefore, the researchers carried out research on recommended hardware that would enable the system run well as expected by the users. The following were the hardware requirements that were proposed for the better running of the system:

- The Hard disk should be not less than 40GB
- The computer monitor screen must be 14 INCH +
- The RAM should range from 1GB+
- The processor speed must be at least 1000MHZ

- The peripherals can take any format but most desirable those with USB compatibilities.
- Computer servers “cloud” to host the project online

#### 4.2.4 Software Requirements

##### Developing software's

Below are some of the software's used in the development of this project, they vary from the platform that was used to develop to the languages used and finally the software's used to utilize the finished project.

##### Platform

This project solely utilizes Google Cloud Services and also gives an option for offline saving of information of which ultimately will be synced to the cloud drive the moment internet is available or connection is made between the offline version and the online version, and updates are automatically inserted into the existing primary document.

##### Languages

- Python
- C++
- Java

##### Browsers

Google Drive on the web supports the two most recent versions of the following browsers:

- Chrome
- Firefox
- Safari (not supported on Windows)
- Internet Explorer

**Note:** Other browsers might work with Google Docs, Sheets, and Slides. However, we can't guarantee that features will work as expected.

No matter what browser you use, you should enable cookies and JavaScript.

If you're working in Internet Explorer 8 and would like to create or edit a Google drawing or presentation, you may need to use another supported browser. If you use Google Apps, you may be unable to upgrade your browser on your own. In this case, please contact your administrator or your IT support for further assistance.

## **Desktop**

### For Windows

- Windows Vista
- Windows XP
- Windows 7
- Windows 8

### For Mac

- Lion (10.7)
- Snow Leopard (10.6)

### For Linux

Google Drive isn't currently available for the Linux operating system. Linux users can still access Google Drive on the web – [drive.google.com](https://drive.google.com) or through the Google Drive mobile app.

## **Mobile and tablet**

### For Android phones or tablets

- Eclair and up (Android 2.1+)

### For iPhone or iPad

- iOS 3.0+

### 4.3 Google Apps Use Case diagram

The following use case diagram has been used to describe the functionality of campus email system with collaboration systems and how to access information from it.

Use case 1 User accesses the campus mail system	
Description	A user validates himself/ herself with the right credentials to be able to login into the mail and be able to leverage mail services
Actor(s)	Users or students with University mails
Pre-condition	The user should have a computer, phone, a tablet or any electronic device that will allow him/her to access internet and should make sure he/she is connected to the internet
Post -Condition	The cloud services are up and running
Scenario	The user authenticates himself / herself into the system, the user is accepted

Table 4.1 Use case 1 accessing campus mail system

Use case 2 User accesses the Collaboration System	
Description	A user validates himself/ herself with the right credentials to be able to login into the collaboration system to leverage its services
Actor(s)	Users or students with University mails
Pre-condition	The user should have a computer, phone, a tablet or any electronic device that will allow him/her to access internet and should make sure he/she is connected to the internet
Post -Condition	The cloud services are up and running
Scenario	The user authenticates himself / herself into the system, the user is accepted

Table 4.2 Use case 2 accessing collaboration system

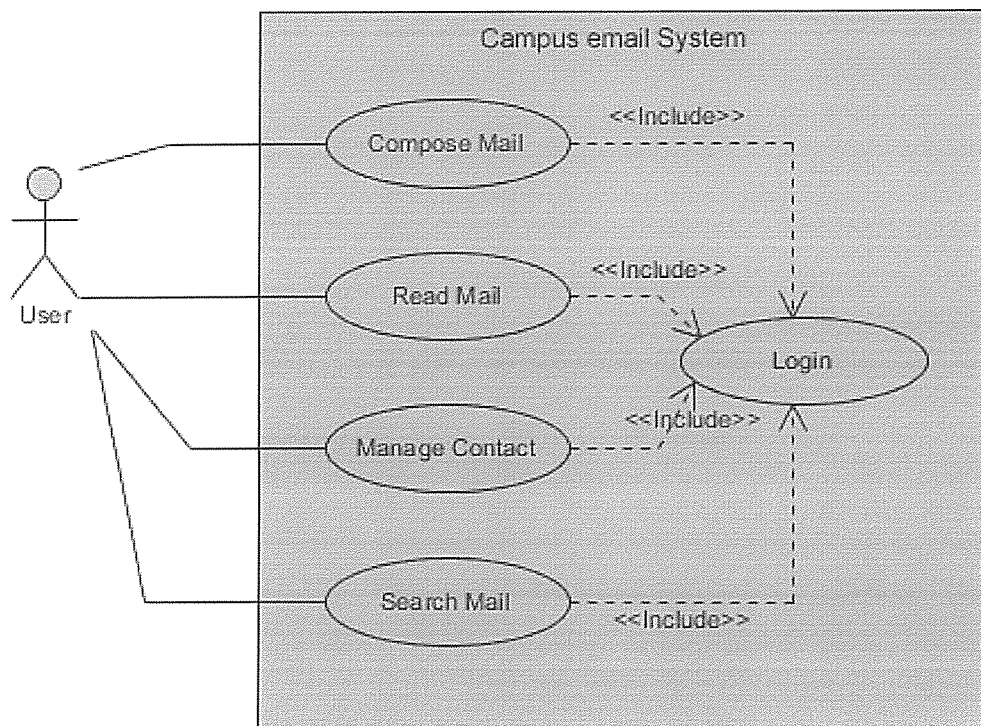


Figure 4.3 a use case diagram of campus email system

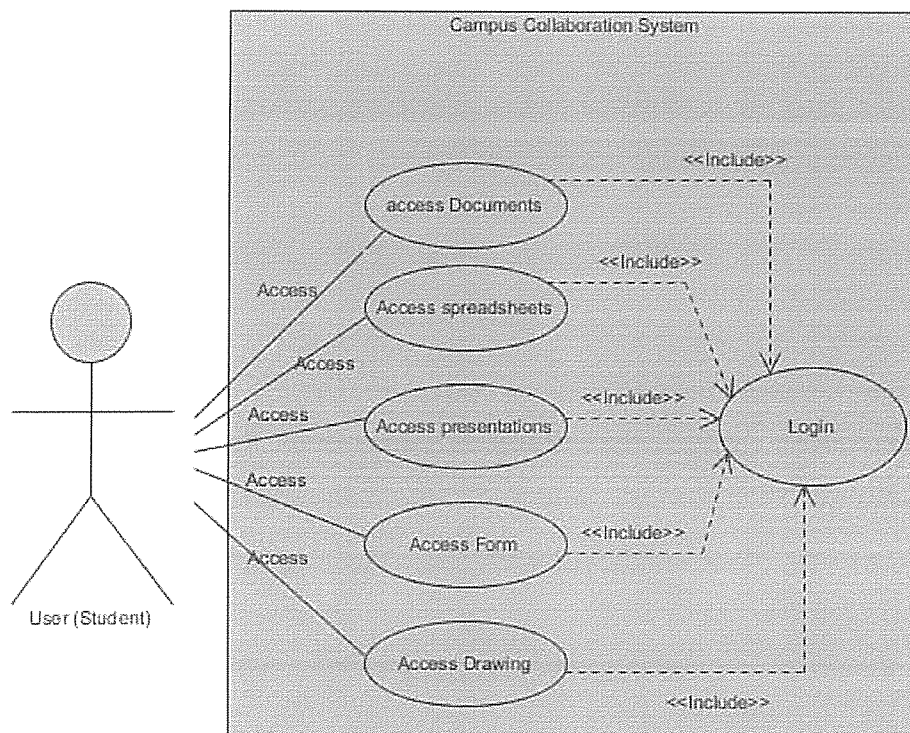


Figure 4.4 a use case diagram of campus collaboration system

#### 4.3.1 Explanation of the above use cases

Before the system is initiated a user must login into the system to enjoy the services the system offers. These services include composing mail, reading mail, managing contacts, searching mails within a user's mailbox among others.

For the effective use of these services one must have the university personalized account. Since without a university personalized email accounts one cannot be able to collaborate.

#### 4.4 Sequence Diagram of how to log into campus email system

Below is a sequence diagram showing how users in our context students will do in order to log into the campus mail system

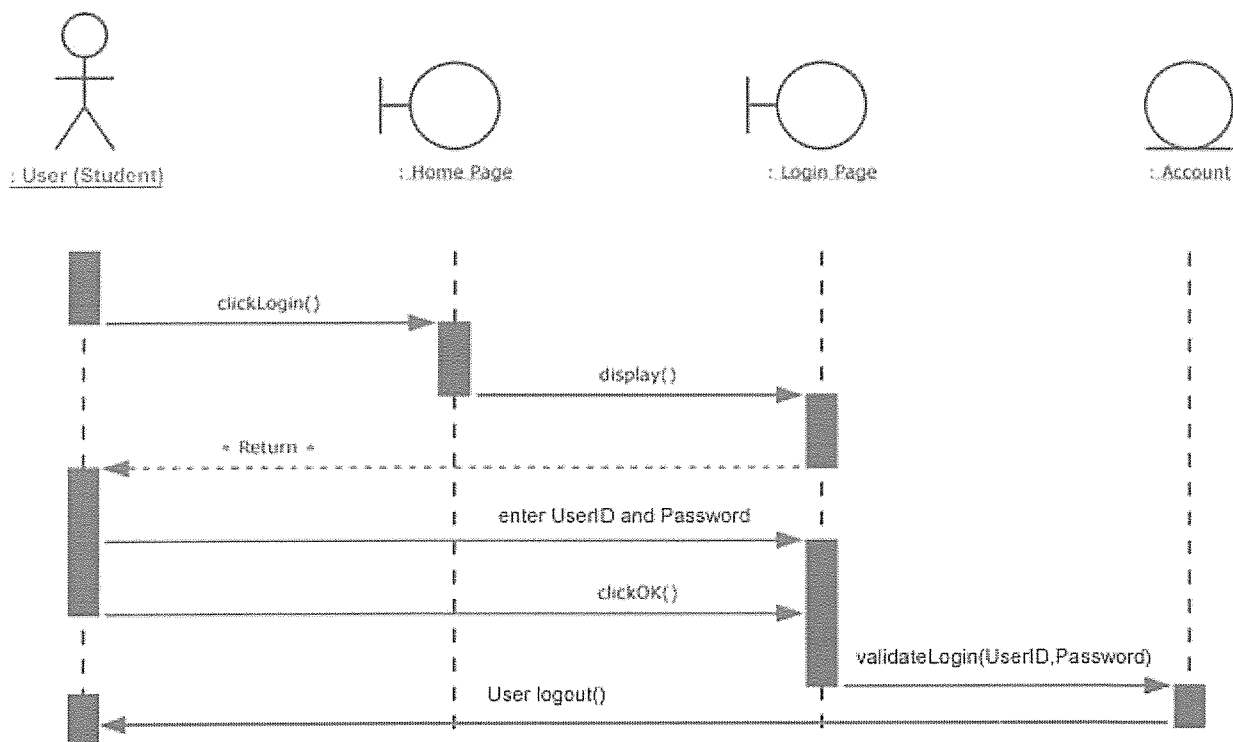


Figure 4.5 sequence diagram showing how a user logs into a mail system

#### Explanation of the above sequence diagram

1. Only students with university mails can access the system
2. A student only needs one email address and a password to access all the services the system offers right from checking mails to being able to participate in collaboration environment
3. Several users can be able to utilize one resource at ago.



## 4.6 User Interface Design

Below is the login user interface design of the campus e-mail system.

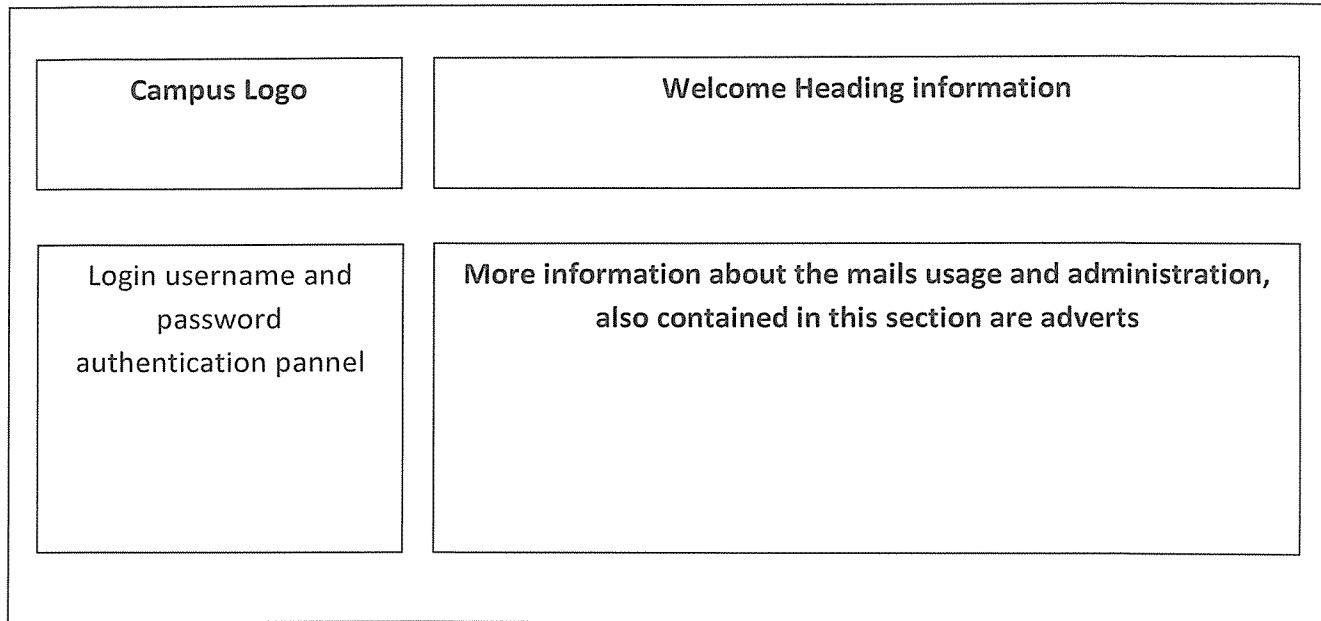


Figure 4.6 campus email system login interface conceptual design

## 4.7 Summary

The greatest achievement of this project is the development of a fully functioning campus email and collaboration system that is currently being adopted by the administration and students of Kampala International University. The application offers quick and easy access to students accounts information; easy access to online research, the proposed system will increase speed to access campus facilities like announcements, access to financial statements hence productivity within the various arms of the university to discharge their duties. The administrator has login passwords to be able to log into the system, hence improving security against unauthorized users.

## CHAPTER 5

### THE CAMPUS EMAIL SYSTEM

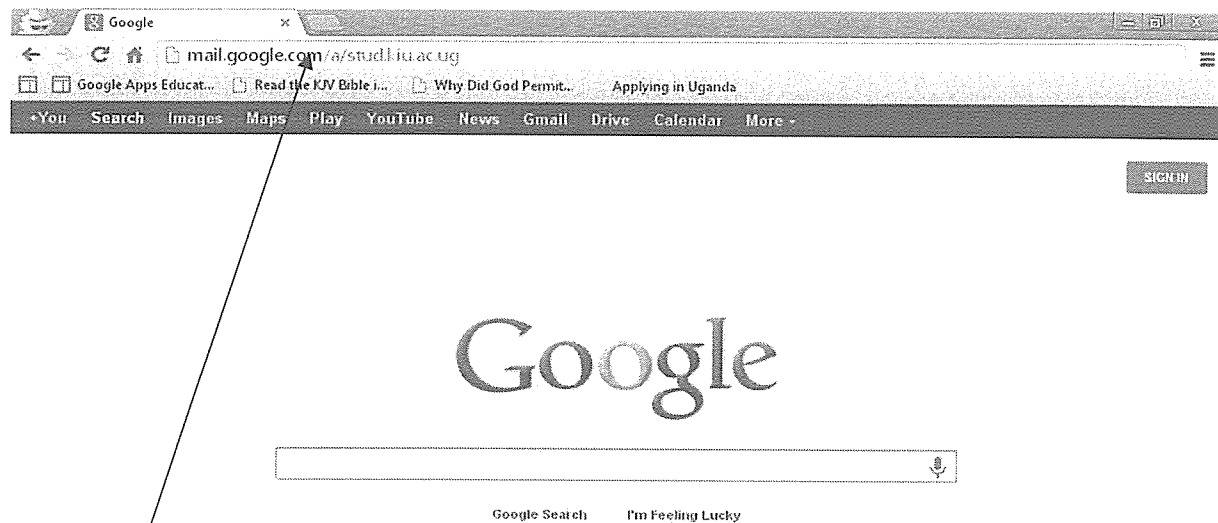
This chapter aims at providing the results of the project. The screen shots illustrating the evidence of the Campus Email system are as indicated in section 5.1 for the Campus Email system user interface and 5.2 for the Campus Email system admin panel.

#### 5.1 Screen shots for Campus Email system user interface.

Here researcher illustrated some important parts of the system basing on the design of the Google apps for education edition. These will show the main features that users interact with.

##### 5.1.1 Screen shots of our project interface

Domain for students accounts



mail.google.com/a/stud.kiu.ac.ug

Figure 5.1 showing access to the system with a student domain

## Login panel for students

Sign in to your account at  
**Kampala International University**

Username:

Password:

☐ Stay signed in

[Can't access your account?](#)

Less spam, plenty of space and access from anywhere.

Welcome to your email for Kampala International University, powered by Google, where email is more intuitive, efficient and useful.

- Keep unwanted messages out of your inbox with Google's powerful spam blocking technology
- Keep any message you might need down the road, and then find it fast with Google search
- Send mail, read new messages and search your archives instantly from your phone

**Download Chrome to get a faster & enhanced Google Apps experience**  
Chrome is a secure and stable browser that offers the fastest and most feature-rich Google Apps experience, including offline support for Gmail, Docs & Calendar.  
[Deploy Chrome company-wide via the MSI](#) or [download Chrome for your personal use.](#)

©2013 Google [Privacy Policy](#) - [Terms of Service](#) Powered by Google

user enter password

user enter user name

Figure 5.2 showing the user log in panel

## User authentication

Loading joshua.eyaru@stud.kiu.ac.ug...

The user is authenticated

Figure 5.3 Showing the authentication of the user.

## Inbox view for user account

## University logo

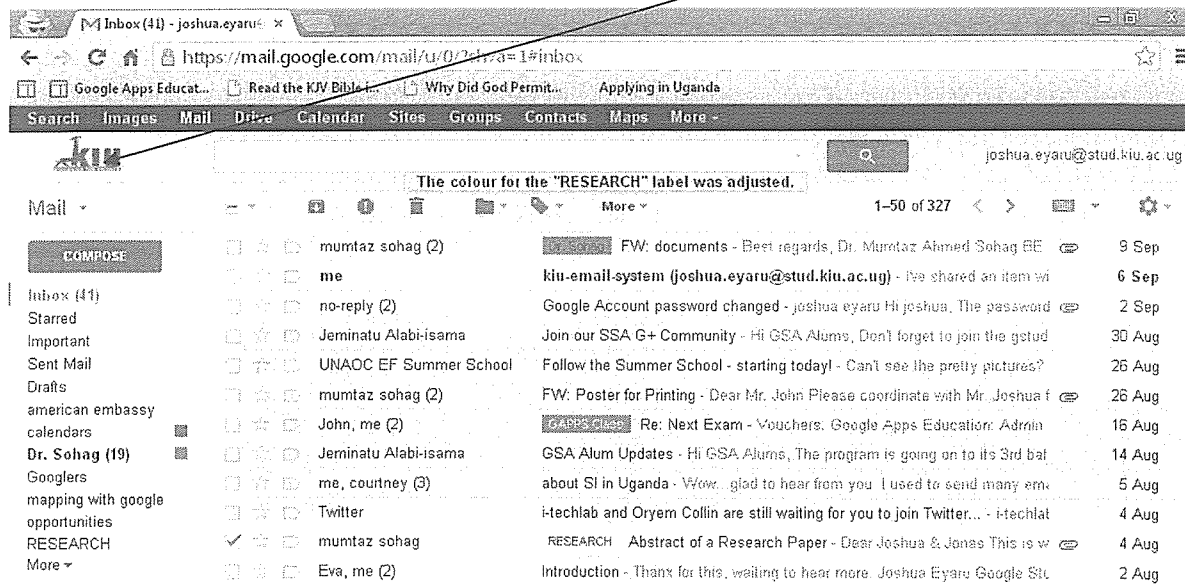


Figure 5.4 showing the users inbox section after authentication.

## Drive (storage and sharing of documents, forms, spread sheets, presentations and files)

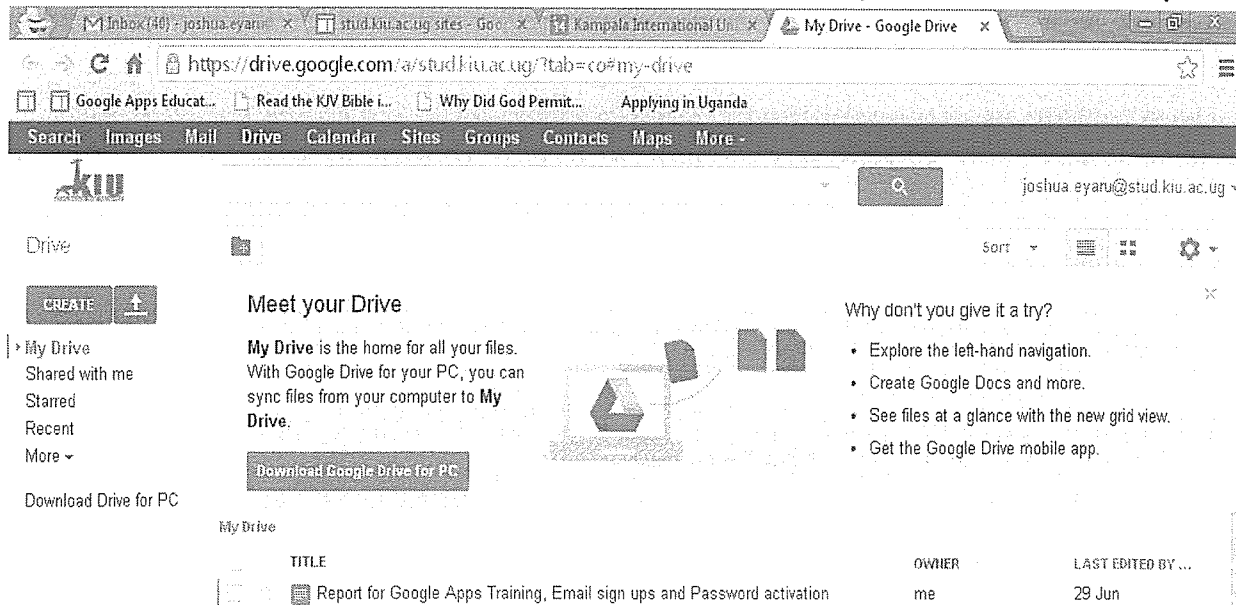


Figure 5.5 showing Google Drive, customized to KIU for cloud storage and sharing of information.

## Customized sites to School domain

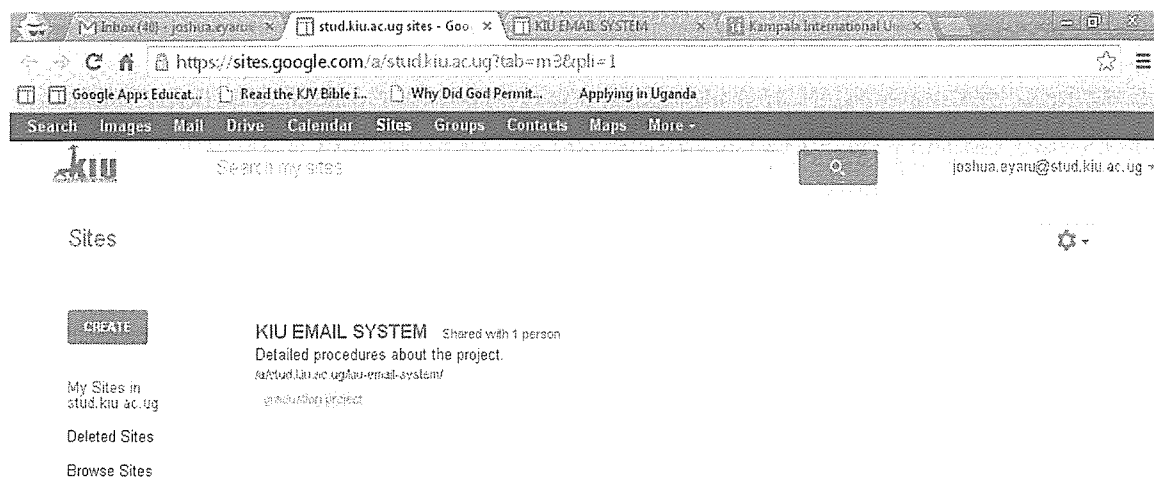


Figure 5.6 customized sites for the school domain

## Calendars

### One day view calendar

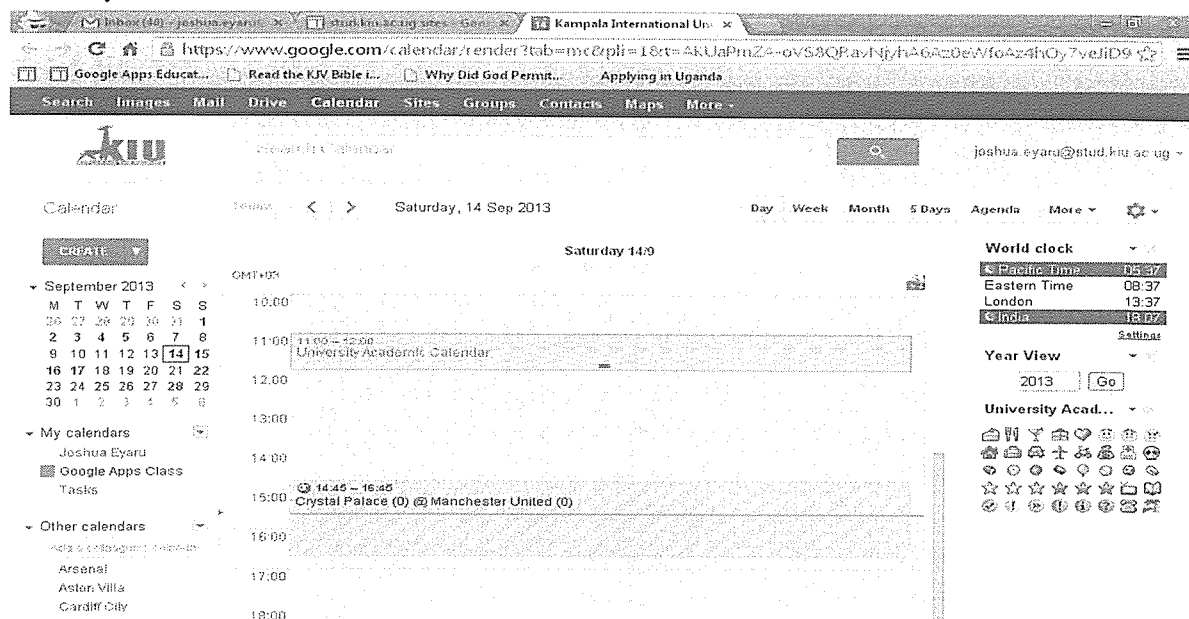


Figure 5.7 a Calendar with one day view.

## A Weeks view calendar

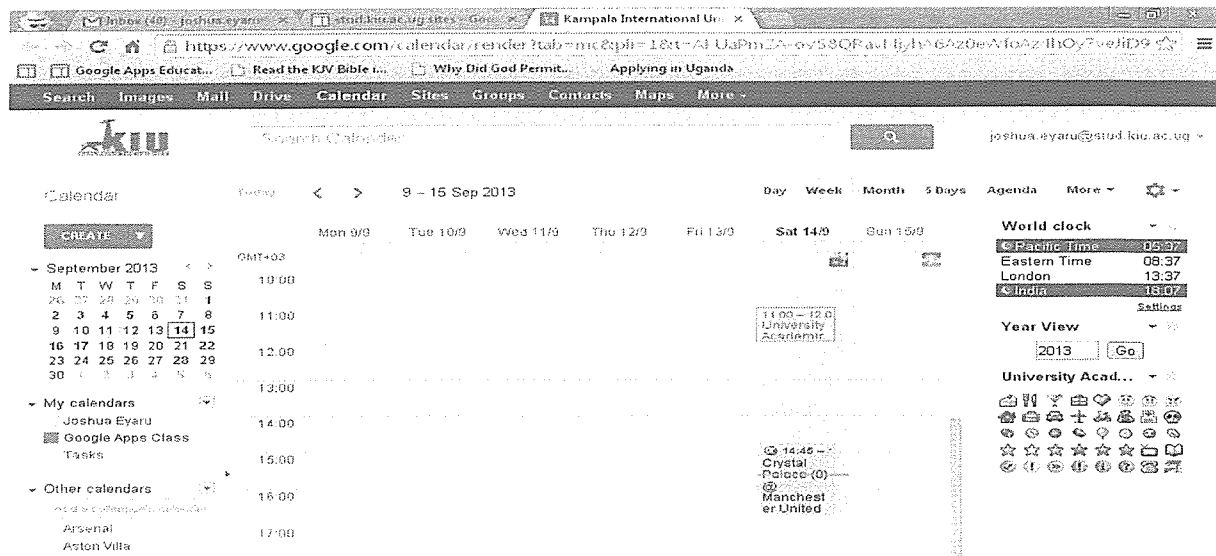


Figure 5.8 showing a week's view of the calendar

## A month view of calendar

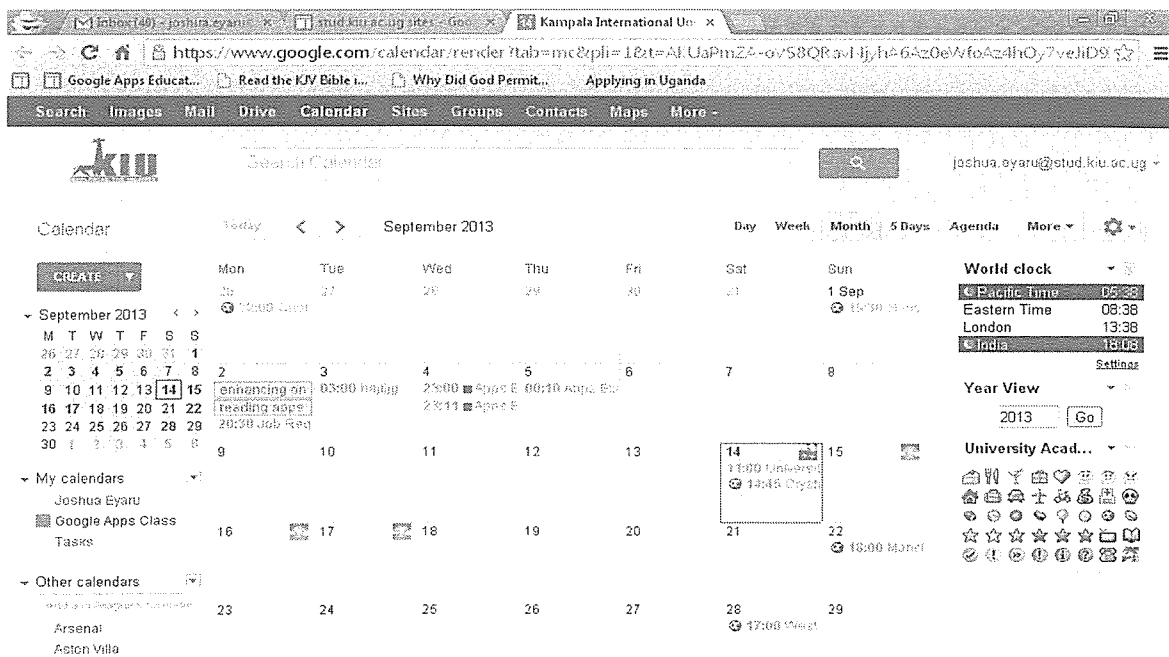


Figure 5.9 showing a month view of calendar

## 5.2 Screen shots Email system admin panel.

### Domain users display page

The screenshot shows the Google Admin console for the domain kiu.ac.ug. The 'Users' page is active, displaying a list of users. The table below represents the data shown in the screenshot:

Name	Email	Status	Storage used	Total storage	Last signed in
Abdullahi Mohamed Hussein	abdullahi.mohamedhussein@stud.kiu.ac.ug	Newly created	0 GB	30 GB	Never logged in
Abel Mwanje	abel.mwanje@kiu.ac.ug	Newly created	0 GB	30 GB	Never logged in
Abraham Maluk Abiot	abraham.maluka@stud.kiu.ac.ug	Newly created	0 GB	30 GB	Never logged in
Abraham Mabior Nhial	abraham.nhialmabior@stud.kiu.ac.ug	Newly created	0 GB	30 GB	Never logged in
Ahmed Abdirahman M	ahmed.abdirahmanm@stud.kiu.ac.ug	Newly created	0 GB	30 GB	Never logged in
Ahmed Ali Mohamed	ahmed.alimohamedahmed35003@stud.kiu.ac.ug	Newly created	0 GB	30 GB	Never logged in

Figure 5.10 showing the display of users active on the domain

### Setup page

The screenshot shows the Google Admin console for the domain kiu.ac.ug, specifically the 'Setup' page. The page displays the 'Google Apps setup wizard' with a list of services to be activated. The services listed are Welcome, Users and groups, Set up your apps, Gmail, Calendar, Drive, Sites, Mobile access, Extend and customize, and Training and support. The 'Drive' service is highlighted, showing details about Google Drive and its features.

Figure 5.11 showing the domain setup page where services allowed to use are activated from.

Security settings page

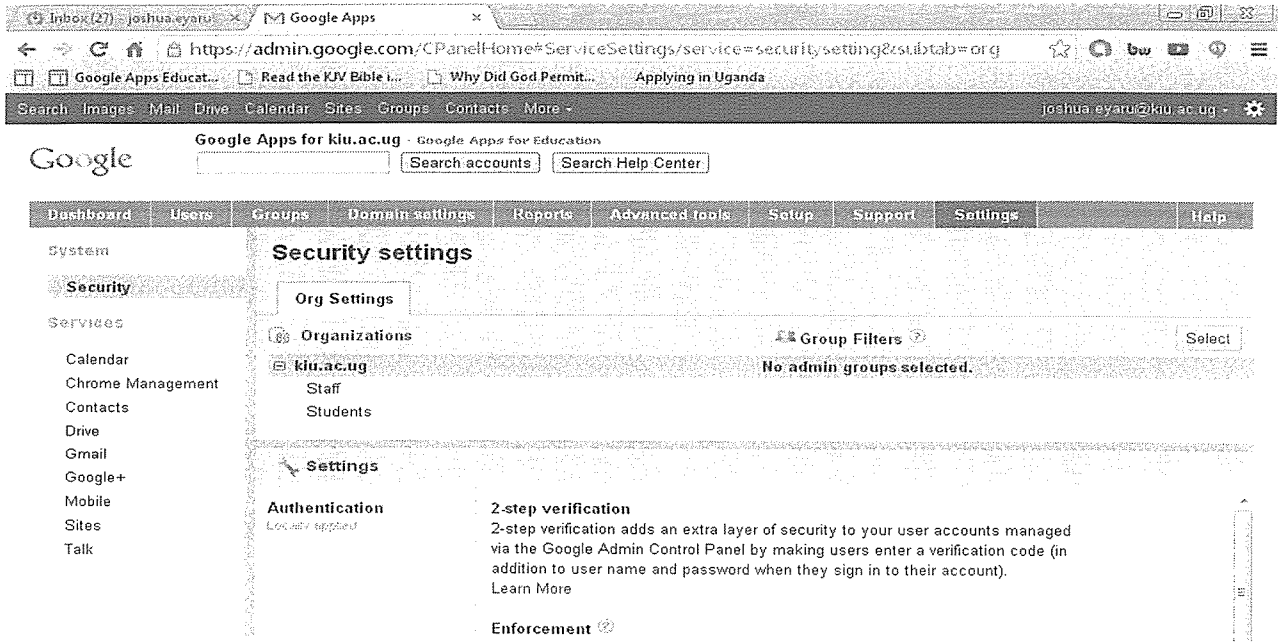


Figure 5.12 showing the security management tools for the email accounts.

Reports page



Figure 5.13 showing the reports section about the users interaction in a given time frame.



### 5.3 Use Case Test Results

These tests set out to establish if the desired functionality of the system matches the functionality of the implemented system. In this project, researcher tested the use cases described in chapter 4 to see if the functionality of the system has been implemented.

Use Case	Description	Implementation
1	Users register and obtain email addresses	✓
2	Users access mail services	✓
3	Users collaborate with each other	✓
4	Users conduct e-learning	✓
5	Users can intergrate several technologies	✓

**Table 5.1 Showing Use case testing of the campus email system**

### 5.4 Requirement Test Results

Testing that the requirements of the specification have been met requires that evidence is produced to support this claim that each requirement has been met. The evidence takes either the form of documented design details that correlate with the implemented system.

**Requirement 1:** Ability to share information by students and staff.

**Evidence:** Students and staff of Kampala International University can now use Google documents embedded in the KIU Google apps for education to share same piece of information. This can give them the ability to edit the same document which they share from anywhere in the world.

**Requirement 2:** Flip classrooms and e-Learning.

**Evidence:** Students can now have flip classrooms where they can have an advance lecture streamed online using the Google+ application and the lecturer only shows up to make few remarks and respond to questions. The e-Learning ability can also be supported since all students have the user accounts.

**Requirement 3:** . Data collection.

**Evidence:** The Heads of departments can also use the embedded forms to collect student's information like semester registration. The forms can also be used to apply to the University by new students and any kind of application including applications for new guild union candidates.

**Requirement 4:** Free SMS.

**Evidence:** The system has the ability for users to have free sms from via the gtalk to supported mobile services.

**Requirement 5:** Video conferencing

**Evidence:** The staff and students can use the video conferencing function “hangouts” to conduct online meetings and virtualizing classes.

**Requirement 6:** Student’s documents.

**Evidence:** Different staff ranging from administrative to teaching staff can use the Email system to deliver student’s documents like finance clearance, semesters results and class test mark.

**Requirement 7:** Online storage

**Evidence:** Students and staff are able to backup their data online using the Drive feature which enables them to store lecture notes, results, ledgers, certificates, transcripts, photos and any kind of important document. These documents can be retrieved from anywhere in the world by the owner.

**Requirement 8:** Online presentations.

**Evidence:** With this system, no more virus stress on the side of students and staff who intend to go for a presentation, the system can support online presentations where you store your presentation and retrieve it from anywhere and carry on with the presentation without the need of moving with it in flash disk or laptop. Therefore all the requirements for this application have been met.

## CHAPTER 6

### CONCLUSION AND RECOMMENDATIONS

#### 6.0 Introduction.

This chapter outlines the success of this project, limitations, challenges the recommendations of the project and also the future prospects of this project that should be researched and added into the existing system.

#### 6.1 Achievements

This project has shown a number of achievements so far and what impact they have contributed to the entire project.

- Successfully created mails for all the students within the university
- Successfully created an online collaboration systems to monitor student performance on coursework's
- Successfully managed to use the mails to distribute ledgers ( financial statements) to students for exam seating verification
- Successfully created sites to enhance student learning through more learning materials to download online
- Successfully created students group mails to keep specific groups updated with what is happening within the specific groups.

#### 6.2 Limitations

Besides the success achieved above this system has also limitations, below are some of the limitations experienced in the system and these are as follows;

- Space available for hosting student mails only reaches a capacity of 30,000 students and incase of a big university having more students than this the system might not cater for all of them.
- Most of the services offered by these project need internet connection. Although some of the applications can work offline until internet is on, the major factor driving it is internet. Thus any institution planning to deploy this project to work on their system should do so with the knowledge that they also should have internet connection to fully utilize the services it comes about with.
- Registration of students take long since an administrator has to create for them email accounts in accordance with university policy and to avoid jamming the system by unnecessary students signing for more accounts than one. Hence slowing the whole process of getting students into the system within a short period of time
- Resetting email passwords requires an administrator to do so thus making the process not user friendly.

### **6.3 Challenges**

During implementation of this project, Researcher faced some challenges. Most of the challenges faced by researcher were in requirement elicitation. Requirements play a major rule in the entire process of application development.

In most cases requirements obtained from users and community are not straight forward. Since requirements gathered were not clear, researcher had to validate and verify them before incorporated them into the system requirements documentation.

### **6.2 Conclusions.**

In conclusion therefore, having a complete system in place and its documentation was not easy. This is because it involved organizing and compiling the whole project required skills in several languages and many editing software such as Python, Java, C++ & PHP.

In comparison with current system, the online campus email system has had numerous components that make it applicable and desirable. These include; User Authentication, account login tracking, online and offline capabilities, inbuilt research supportive application, user smart search of information among many others.

### **6.3 Recommendations**

We would like to thank the university and administration of Kampala International University as a whole for adopting this system to manage its day to day operations and to improve the productivity of the key and vital areas in the university like the finance, the faculties and education at large because students now can do their coursework online and be able to be marked and given the results online without them physically coming into contact with the students. The other encouraging point is that some of the lectures have already gone ahead to create Google sites that enables them issue course notes and students can be able to download them and read them at a later time

It is with this regard that I would like to urge other universities to employ the same mechanism we have in Kampala International University. Truly we are exploring the heights.

We recommend that the features below should be added to the system in the future:

- (i) Merging the online result system to work in line with the campus mails to allow students have access to their results while in their mails.
- (ii) Incorporating the long distance learners in the system so that they do not require to travel for studies but utilize flip classrooms and be able at the same time to access their exams online.

We therefore urge that campus email collaboration system should be fully implemented and utilized to the maximum in Kampala International University and any other university that is willing to upgrade from the old office offline system to the online cloud system.

## REFERENCES

- Bailin, S., Case, R., Coombs, J. R., & Daniels, L. B. (1999). **Conceptualizing critical thinking.** *Journal of Curriculum Studies*, 31(3), 285–302.
- Bonk, C. J. & Smith, G. S. (1998). **Alternative instructional strategies for creative and critical thinking in the accounting curriculum.** *Journal of Accounting Education*, 16(2), 261-293.
- Bossert, S. T. (1988). **Cooperative activities in the classroom.** *Review of Research in Education*, 15(1988-1989), 225–250.
- Broussard, S. C., & Garrison, M. E. B. (2004). **The relationship between classroom motivation and academic achievement in elementary school-aged children.** *Family and Consumer Sciences Research Journal*, 33(2), 106–120.
- Case, R. (2005). **Moving critical thinking to the main stage.** *Education Canada*, 45(2): 45–49.
- Cross, D. R. & Paris, S. G. (1988). **Developmental and instructional analyses of children's metacognition and reading comprehension.** *Journal of Educational Psychology*, 80(2), 131–142.
- Dillenbourg, P. (1999). **What do you mean by 'collaborative learning?'** In P. Dillenbourg (Ed.), *Collaborative-learning: Cognitive and Computational Approaches* (pp.1–19). Oxford: Elsevier.
- Dillenbourg, P., Baker, M., Blaye, A., & O'Malley, C. (1996). **The evolution of research on collaborative learning.** In E. Spada & P. Reiman (Eds.), *Learning in humans and machine: Towards an interdisciplinary learning science* (pp. 189-211). Oxford: Elsevier.
- Ennis, R. H. (1985). **A logical basis for measuring critical thinking skills.** *Educational Leadership*, 43(2), 44–48.
- Facione, P. A. (1990). **Critical thinking: A statement of expert consensus for purposes of educational assessment and instruction.** Millbrae, CA: The California Academic Press.
- Fall, R. Webb, N., & Chudowsky, N. (1997). *Group discussion and large-scale language arts assessment: Effects on students' comprehension.* CSE Technical Report 445. Los Angeles, CRESST.

- Gottfried, A. E. (1990). Academic intrinsic motivation in young elementary school children. *Journal of Educational Psychology*, 82(3), 525–538.
- Halpern, D. F. (1998). Teaching critical thinking for transfer across domains: Dispositions, skills, structure training, and metacognitive monitoring. *American Psychologist*, 53(4), 449–455.
- Hennessey, M. G. (1999, March). Probing the dimensions of metacognition: Implications for conceptual change teaching-learning. Paper presented at the annual meeting of the National Association for Research in Science Teaching, Boston, MA.
- Heyman, G. D. (2008). Children's critical thinking when learning from others. *Current Directions in Psychological Science*, 17(5), 344–347.
- Hidi, S. & Harackiewicz, J. M. (2000). Motivating the academically unmotivated: A critical issue for the 21st century. *Review of Educational Research*, 70(2), 151–179.
- Kramarski, B. & Mevarech, Z. R. (2003). Enhancing mathematical reasoning in the classroom: The effects of cooperative learning and metacognitive training. *American Educational Research Journal*, 40(1), 281–310.
- Kuhn, D. & Dean, D. (2004). A bridge between cognitive psychology and educational practice. *Theory into Practice*, 43(4), 268–273.
- Lipman, M. (1988). Critical thinking—What can it be? *Educational Leadership*, 46(1), 38–43.
- Martinez, M. E. (2006). What is metacognition? *Phi Delta Kappan*, 87(9), 696–699.
- McLeod, L. (1997). Young children and metacognition: Do we know what they know they know? And if so, what do we do about it? *Australian Journal of Early Childhood*, 22(2), 6–11.
- Mercer, N. (1996). The quality of talk in children's collaborative activity in the classroom. *Learning and Instruction*, 6(4), 359–377.
- Nelson, C. E. (1994). Critical thinking and collaborative learning. *New Directions for Teaching and Learning*, 1994(59), 45–58.
- Palincsar, A. S. (1987, April). Collaborating for collaborative learning of text comprehension. Paper presented at the annual meeting of the American Educational Research Association, Washington, D.C.

Paris, S. G. & Winograd, P. (1990). Promoting metacognition and motivation of exceptional children. *Remedial and Special Education*, 11(6), 7–15.

Paul, R. W. (1992). Critical thinking: What, why, and how? *New Directions for Community Colleges*, 1992(77), 3–24.

Race, P. (2001). *A briefing on self, peer, and group assessment. Assessment Series Number 9*. York, UK; Learning and Teaching Support Network.

Roschelle, J. (1992) Learning by collaborating: Convergent conceptual change. *Journal of the Learning Sciences*, 2, 235–276.

Roschelle, J. & Teasley, S. D. (1995). The construction of shared knowledge in collaborative problem-solving. In C.E. O'Malley (Ed.), *Computer-supported collaborative learning* (pp. 69–97). Berlin: Springer-Verlag.

Salomon, G. & Globerson, T. (1989). When teams do not function the way they ought to. *International Journal of Educational Research*, 13(1), 89–100.

Saner, H., McCaffrey, D., Stecher, B., Klein, S., & Bell, R. (1994). The effects of working in pairs in science performance assessments. *Educational Assessment*, 2(4), 325–338.

Schraw, G., Crippen, K. J., & Hartley, K. (2006). Promoting self-regulation in science education: Metacognition as part of a broader perspective on learning. *Research in Science Education*, 36, 111–139.

Schraw, G. & Moshman, D. (1995). Metacognitive theories. *Educational Psychology Review*, 7(4), 351–371.

Thayer-Bacon, B. J. (2000). *Transforming critical thinking: Thinking constructively*. New York: Teachers College Press.

Tindal, G. & Nolet, V. (1995). Curriculum-based measurement in middle and high schools: Critical thinking skills in content areas. *Focus on Exceptional Children*, 27(7), 1–22.

Tudge, J. R. H. (1992). Processes and consequences of peer collaboration: A Vygotskian analysis. *Child Development*, 63(6), 1364–1379.

Turner, J. C. (1995). The influence of classroom contexts on young children's motivation for literacy. *Reading Research Quarterly*, 30(3), 410–441.



Van Boxtel, C., Van der Linden, J., & Kanselaar, G. (2000). Collaborative learning tasks and the elaboration of conceptual knowledge. *Learning and Instruction, 10*(4), 311–330.

Webb, N.M. (1991). Task-related verbal interaction and mathematical learning in small groups. *Research in Mathematics Education, 22*(5), 366–389.

Webb, N. M. (1993). Collaborative group versus individual assessment in mathematics: Processes and outcomes. *Educational Assessment, 1*(2), 131–152.

Webb, N. M. (1995). Group collaboration in assessment: Multiple objectives, processes, and outcomes. *Educational Evaluation and Policy Analysis, 17*(2), 239–261.

Webb, N. M., Nemer, K. M., Chizhik, A. W., & Sugrue, B. (1998). Equity issues in collaborative group assessment: Group composition and performance. *American Educational Research Journal, 35*(4), 607–651.

Willingham, D. T. (2007). Critical thinking: Why is it so hard to teach? *American Educator, 31*(2), 8–19.

