EDUEASY – SMART LEARNING ASSISTANT SYSTEM

Project ID: 2020-009

Project Proposal Report

IT117112192 – Jayasuriya J.A.T.P.

IT17407458 – P.D. Gallage

IT17153096 – J.M.S.D. Jayasundara

IT17106634 – L.A.P.Y.P Nuwanjaya

B.Sc. (Hons) Degree in Information Technology

Department of Information Technology

Sri Lanka Institute if Information Technology

Sri Lanka

February 2020

DECLARATION

We declare that this is our own work and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Name	Student ID	Signature
Jayasuriya J.A.T.P.	IT17112192	
P.D. Gallage	IT17407458	
J.M.S.D. Jayasundara	IT17153096	
L.A.P.Y.P. Nuwanjaya	IT17106634	

The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

Signature of the supervisor Date

....

(Dr. Anuradha Karunasena)

Abstract

Usage of smart learning concepts has been increased rapidly all over the world recently as better teaching and learning methods. Most of the educational institutes such as universities are experimenting those concepts with their students.

As university students, there are many educational challenges and difficulties we face on a daily basis. EduEasy - Smart Learning Assistant System is designed to provide solutions for some major problems we have identified during our time in the university. Those problems can be commissioned as difficulty of note taking in a university lecture, difficulty of searching online references and questions related for lectures and difficulty of referring a lecture note and lecture slides at the same time.

Since this is a pretty extensive project, the whole system will be separated into four different components as Note Taker Component, Reference Finder Component, Question Presenter Component and Slide Matcher Component. Each of these components will be developed by different members using different tools and technologies.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Background & Literature Survey

EduEasy is a smart learning assistant system. It is designed as a web based E-learning application. This application is designed to provide solutions for some major difficulties university students experience during lecture room environment. As university students, we experience these difficulties regularly. EduEasy application is basically providing solutions for the four problems which are mentioned below.

1. Students experience that note taking is hard to do in a university lecture.
2. Students believe that searching for references for lectures from internet is too much time consuming.
3. Students finds that finding questions related to their lessons from the internet is hard.
4. When referring a lecture note, students like to refer their lecture slides with the note. Some students find this process is difficult to conduct.

Since there is no existing product or research similar to the EduEasy Smart Learning Assistant System as a whole product, literature studies had to be carried out component wise. The Note Taker component is one of the four major components of the system. Writing a note in a lecture room environment is difficult because of several reasons. Sometimes the lecturers are speaking too fast. Sometimes students are focused on trying to understand the lecture so they miss to write the note. So as a solution, a component named Note Taker is implemented in the smart learning assistant system.

Note taker component has two basic parts. In the first part the component, voice of the lecturer is transcribed to text. There are some previous researches done on this voice transcription part [2]-[3]. In those researches, they use different methods to transcribe audio into text. The next part of the component is the text summarization. Automatic text summarization is a complex task which contains many sub-tasks in it. Every subtask has an ability to get good quality summaries. There are also researches done on automatic text summarization, text summarization machine learning

algorithms etc. [4]-[6]. They use methods like machine learning algorithms and clustering & extraction to perform text summarizations.

According to those research papers there are basically two methods of text summarization. First one is Extractive summarization and the second one is Abstractive summarization. Extractive summarization is basically creating a summary based on strictly what you get in the text. It can be compared to copying down the main points of a text without any modification to those points and rearranging the order of that points and the grammar to make more sense out of the summary. Abstractive summarization techniques tend to mimic the process of 'paraphrasing' from a text than just simply summarizing it. Texts summarized using this technique looks more human-like and produce more condensed summaries.

Reference Finder component finds exact important places in relevant various references according to a keyword or a key phrase. It is great to support students who are struggling with understanding lessons when the exams are closer. Reference Finder component has two main processes. First, it has to identify a keyword or a key phrase from a summarized note which is stored in the application's cloud database [10]. Second, it has to automatically search for the reference materials from the internet and list down the results for referring. Finally, it classifies the search results. As the first process, the reference finder has to identify keywords or key phrases from the summarized note. Keyword search algorithms are the most suitable methods for identifying keywords. Keyword search is a proven and widely accepted mechanism for querying in textual document systems and the World Wide Web [12]. Many pieces of research have been implemented in order to search for keywords.

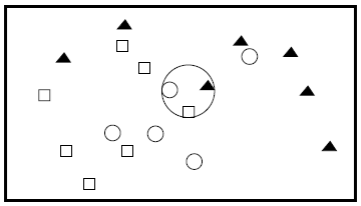

Figure 1.1 - mCK query for three keywords obtained from placemarks

In a research paper about keyword searches, the authors have come up with an innovative query called the "mclosest keywords (*m*CK) query" [13]. When the user

provides m keywords, the *m*CK query focus on finding the closets tuples that match those keywords. Their main aim is to apply mCK query for documents.

Figure 1.1 shows the dimensional distribution of three keywords that are obtained from placemarks in some mapping applications. This scenario can be done by using *m*CK query on the three keywords. The measure of closeness for a set of *m* tuples is defined as the maximum distance between any two of the tuples [13]. To implement *m*CK query the authors have used five main algorithms for keyword searching.

Algorithm 1 — Finding *m* Closest Keywords
Algorithm 2 — SubsetSearch: Searching in a Subset of Nodes
Algorithm 3 — SearchInOneNode: Searching in One Node
Algorithm 4 — SearchInMultiNodes: Search in Multiple Nodes
Algorithm 5 — Enumerate: Enumerate All Possible Candidates

The Question Presenter component finds exact important questions online relevant to different parts of the lectures. Using this application student can get relevant questions using key phrases. This is an automatically Google search by using the question presenter component. Existing evaluations of relational keyword search systems are4ad hoc5with little standardization. Webber [29] summarize existing evaluations with regards to search effectiveness. Although Coffman and Weaver [22] developed the1benchmark that their work does not include any performance evaluation. Baid et al. [18] assert that many existing keyword search techniques have5unpredictable performance due to unacceptable1response times or fail to produce results even after exhausting memory. The7results particularly the large memory footprint of the systems confirm4this claim. A number of relational keyword search systems have been published5beyond those included in this2evaluation. Chen et al. [21] and Chaudhuri5and Das [20] both presented tutorials on keyword search in databases. Yu et al. provides an excellent overview of relational keyword search techniques.

Schema based methods support keyword search over relational database (such as SQL) using SQL commands [24] to execute. Such methods incorporate edges and vertices with tuples and keys (primary and international key). Each tuple used in the database

defines interdependence among tuples as the vertex and edges. In the case of RDBMS, keyword searching is performed using the Schema Based Approach using SQL. Schema Based4approach working is divided into the two main steps:

i) Determine how to create and generate SQL queries in order to find out the structures among tuples.
ii) Determine how to evaluate the queries which are generated in step 1 efficiently.

There are certain methods that exist for solutions focused on schemas. The search requirement for the text data cannot be met for the same existing keyword search method. These strategies concentrate on the performance and efficiency of keyword query searches [23]. They propose a new ranking formula using existing information retrieval techniques. Major importance of this 2 technique is works on large scale real databases (Eg. Commercial application which is Customer Relationship Management) using two popular RDBMS effectiveness and Efficiency. For the ranking function it uses a Top-k Join algorithm which includes two efficient query processing algorithms.

Slide Matcher component matches the relevant lecture slide content according to the generated summarized note. It means If student wants to refer the lecture note which is published by the lecturer, can navigate to the lecture note directly using slide matcher tool. Student has to click on the sub topic of summarized note and system will displays the exact place where the lecture content is.

There are many functions in NLP which helps to re-process the data. The research [25] has explained about the main steps and clear solutions to overcome the problems if only string matching algorithms are used to compare the similarity of two sentences. Data cleaning is a function about to clean the data. Get the data in clean and standard format for further analysis is very important. This function can remove the unwanted spaces, commas and etc. After getting the data cleaned, system cannot match the whole document at once. So there is an another function called Tokenization to reduce the complex of sentences. Tokenization can break a complex sentence into words, understand the importance of each word to the sentence and produce a structured description on an input sentence.

Next function is Stemming. Stemming does normalize words into its base form or root form, cuts the prefixes and suffixes. Changing affected, affective to affect is an example. There is an another function called Lemmatization which groups together different infected forms of a word called Lemmas and outputs a proper word. Using Lemmatization, the words ‘go’, ‘going’ and ‘gone’ will turn into ‘go’. Chuncking is an another function which is picking up individual pieces of information and grouping them into bigger pieces such as nouns, verbs and etc. POS tags and Named Entity Recognition are some other functions in NLP which helps to reprocess the data.

Another research [26], published by Aselsan research center in Turkey described about the types of text similarity and how it helps to find the most similar sentences. It describes about mainly two types in text similarity. Lexical similarity and Semantic similarity are the main two types. Lexical similarity is about surface closeness and semantic similarity is about the meaning of the sentence. When we measure the similarity, both similarities should be noticed. An article [27] explained about this problem with an example. There are two sentences called “The cat ate the mouse” and “Mouse ate cat’s food”. If the word level similarity is only concerned, these two phrases appear very similar as 3 of the 4 unique words same. But when we consider the meaning, it is totally different. To overcome this problem there is an algorithm called Jaccard similarity which is explained in an article [27].

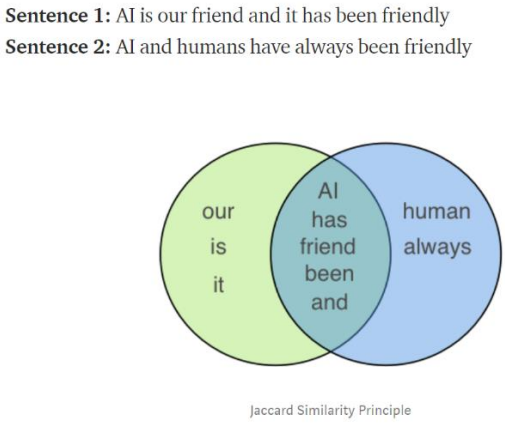

Figure 1.2 – Jaccard similarity principle

Above figure explans about Jaccard similarity. There is an another problem may occur when comparing two sentences. Both sentences have the same meaning but the words used are exactly different. Then when we compare the similarity, both sentences do not match with each other. For exam a research paper [28] explained it with an example. “President greets the press in Chicago” and “Obama speaks in Illinois” are those two sentences and to identify the comparison of those two sentences that research paper [4] explains an algorithm called K-mean algorithms and Hierarchical Clustering Dendrogram. Here these two sentences should be converted into vectors first and there are methods explained “Count Vectorizer method” and “Word Embeddings”.

Following table is an existing product comparison similar to EduEasy Smart Learning Assistant System. Although there are no existing products similar to the EduEasy system as a whole product, there are some products can be compared to individual components of the system.

	LiguaKit	SlidePlayer	CSUN	GoConar	CopyLeaks	Proposed Solution
Search in any place on the internet	✗	✗	✗	✗	✗	✓
Show user targeted recommendations	✗	✓	✓	✓	✗	✓
Search every type of reference materials	✗	✗	✗	✗	✗	✓
Search every type of questions(MCQ/Essay)	✗	✗	✗	✗	✗	✓
Can summarize audio transcripts	✗	✗	✗	✗	✗	✓
Can format the note under sub topics	✗	✗	✗	✗	✗	✓
Can generate synonyms for selected key phrases	✗	✗	✗	✗	✓	✓
Free of charge	✓	✓	✓	✓	✓	✓
Provide Web platform	✓	✓	✓	✓	✓	✓

Table 1.1 – Comparing Existing Products

1.2 Research Gap

According to the available research papers and resources there are several systems developed to Smart Learning System. So far, there's no simple, cohesive concept of smart learning. The definition of smart learning is constantly discussed by multidisciplinary scholars and educational professionals. In general, the atmosphere for intelligent learning is productive, efficient and engaging. In this section, the research gap between previous researches/products and our research will be discussed component wise and as a whole product.

Note Taker component is one of the four components of the EduEasy smart learning assistant system. What this particular component basically does is transcribe the lecturer's voice summarize it. Then the summarized note can be uploaded to the online database so that students can refer it later.

There are many previous researches done on voice transcriptions [2]-[3]. In those researches, they use different methods to transcribe audio in to text. There are also researches done on automatic text summarization, text summarization machine learning algorithms etc. [4]-[5]. They use methods like machine learning algorithms and clustering & extraction to perform text summarizations.

But there is no research or product which is a combination of voice transcription and text summarization. So in this note taker component, those two features above mentioned will be combined and create a full functioning speech-to-text & text summarization feature in the EduEasy smart learning assistant system application.

In the suggested reference finder component has designed for search every type of reference material all over the internet according to the identified keyword from the summarized note. Not only that reference finder classifies the target results according to the material type. Users can experience all those benefits at once from only using EduEasy application.

Using the Question Presenter this application, student can get relevant questions using key phrases. This is an automatically Google search by using question presenter component. These things make this component unique. From the students can select the

key phrases from the schedule lectures and can get questions using internets easily. It will have done by systems automatically. Learning is conventionally defined as the process of acquiring competence and understanding. It results in a new ability to do something, and an understanding of something that was previously not understood.

All those existing smart learning apps basically give few questions for the students. Those are not useful all of the students. Using this question presenter component student can get lot of questions regarding their lecture topics. This is the special feature of this component. In our question presenter component system is capable for increase the answering the questions skill level. It's very useful for students to study.

Slide Matcher component compares the selected topic or sub topic with the lecture notes which are stored in cloud database. Normally the system does that the words or letters which are in the selected topic are taken to an array and try to match with the most similar words from the database. There are some researches which are done about string matching algorithms to compare the similarity of two words and two sentences. If the system takes word by word in a sentence and stores those words in an array and try to find the similarity, it might take too much time and will produce wrong output because of Stemming and Lemmatization as discussed in literature survey [25].

As a whole product, EduEasy Smart Learning Assistant System is a unique complete system which has been never created before.

1.3 Research Problem

Students who study smart instead of study hard, can have good success in their education. But studying smartly is a challenge without a correct support. University students get used to study their lessons in various ways. Students who do not go lectures regularly may not have good lecture notes. Those who don't have a proper lecture note may feel uncomfortable near an exam. EduEasy is an E-learning application for students to effectively learn and revise lectures done at university. EduEasy application has mainly four components. They are note taker, reference finder, question finder and slide matcher.

Students who don't have proper notes, may feel uncomfortable during the exam period because without having a good note it is very difficult to understand the theories and learn them. Then they waste their time to find out good notes. EduEasy application generates a summarized short note by recording the lecturer's voice and converting it into a text document. Then using some algorithms unwanted data will be removed and generates a summarized note. After approving the summarized note by the lecturer, it will be uploaded to the system and student can refer it.

While referring the summarized note students may want to refer some questions or regarding to the relevant part. Then students should have to find sample questions by manually searching in search engines. Then the students time will waste and sometimes they might not find sample questions. EduEasy application has a method to find the questions directly. Question finder component directly navigates to the websites which has sample questions regarding to the relevant lecture part.

Referring the relevant reference books is very important as undergraduates. But most of the students skip to refer the reference books. Lecturer can't teach everything in the lecture time. Students should search more information by themselves. Our proposed system has a tool to navigate to the relevant references directly from the summarized note, it will help students to find references easily.

Normally students try to find the lecture notes by manually when it comes near to an exam. Sometimes they don't have lecture notes with them or the system which is maintained by university might be removed lecture content from the system. But the Slide Matcher tool can find the correct lecture note navigate directly to the relevant content. It will save the time of students and students can study smartly by comparing summarized note and lecture note.

The proposed system will help to overcome those problems which are explained above. Students can study easily without wasting time to find short notes, sample questions, references, lecture notes.

2. OBJECTIVES

2.1 Main Objective

Develop an E-learning application for students to effectively learn and revise lectures done at the university.

2.2 Specific Objectives

2.2.1 Note Taker Component

Capture the lecturer's voice using a voice recorder and automatically store it in the computer. Develop a method to create transcript of the voice recording. Create an algorithm to identify the important parts of the voice record and create a summarized lecture note by removing unnecessary parts, then store in the database.

2.2.2 Reference Finder Component

Retrieve the summarized notes from the database and scan/read it. Develop an algorithm to identify important keywords/phrases from the notes and create a method to search for online references using those keywords/phrases. Create a database to store online reference documents for offline use.

2.2.3 Question Presenter Component

Find questions from online resources related to the summarized lecture note. Develop an algorithm to identify suitable keywords/phrases suitable for questions & create a method to find relatable questions from online resources. Create interfaces to display questions in a clear way and arrange them accordingly.

2.2.4 Slide Matcher Component

Match the lecture slide content according to the summarized lecture note. Create an algorithm to identify unique keywords/phrases which are common to both lecture slides & the summarized note and create a method to match lecture slides with the particular area of the note.

3. METHODOLOGY

3.1 System Architecture

The proposed EduEasy Smart Learning Application has four main functionalities.

1. Note taker
2. Slide matcher
3. Question presenter
4. Reference finder

According to figure 3.1, it demonstrates an overview of the system of EduEasy web application. The process begins with the note taker component by recording the lecturer's voice to the computer. Then using a transcription algorithm (Google API), audio is converted into text format. This ends the audio transcription part of the component. Then the summarizing part begins. The transcribed text is converted into a summarized note using summarization algorithms discussed in the Literature Survey. The summarized note will be opened in the text editor so that the lecturer can edit the note manually if they desire. Then the lecturer uploads the summarized note into the cloud database, so students can refer to it from their computers.

Secondly, the student can refer to lecture slides by using the slide matcher component. The summarized note is available in cloud database and student can choose the relevant note. There are sub topics in each summarized note and when a student wants to navigate to the lecture note from a sub topic, students should click on the bullet point which is placed each and every sub topic. Then the system selects the sentence which is connected with the bullet. Once the bullet mark is clicked, the system takes the main topic of the summarized note and starts finding the exact lecture note from the cloud database. Then System takes all words in the selected line into an array and re-processes those data using NLP algorithms, and word similarity algorithms such as Jaccard similarity, K-mean algorithms. Then the system tries to find the most appropriate words from the lecture notes.

Then the system displays the results and Students can refer to the relevant lecture slide easily.

If the student wants to refer to the reference materials, the student can easily access the reference finder component. In the Reference Finder component first, the student has to select a summarized note from the cloud database. Then he must select a topic. Here the system identifies the topic as a keyword or a key phrase. It is very suitable to use keyword search algorithms to implement identifying the topics process [12]-[14]. Then the system automatically google search for references according to the selected keyword. For the automatic web search process, the system has to use keyword search algorithms [15]. Then the system classifies the search results according to the types of reference materials. For the results classifying process, the system uses classifying algorithms and meta search engines [17]. Finally, the student can refer to the results as his wish.

Finally, the student can refer to the question presenter component. Here the student has to select a summarize note from the cloud database which we stored previously. After that students can select key phrases. It means those are topics in lecture slides. And also check synchronism for given keywords. This is an automatic Google search by using “machine learning” algorithms. Finally, the student can select the most suitable questions by using web links.

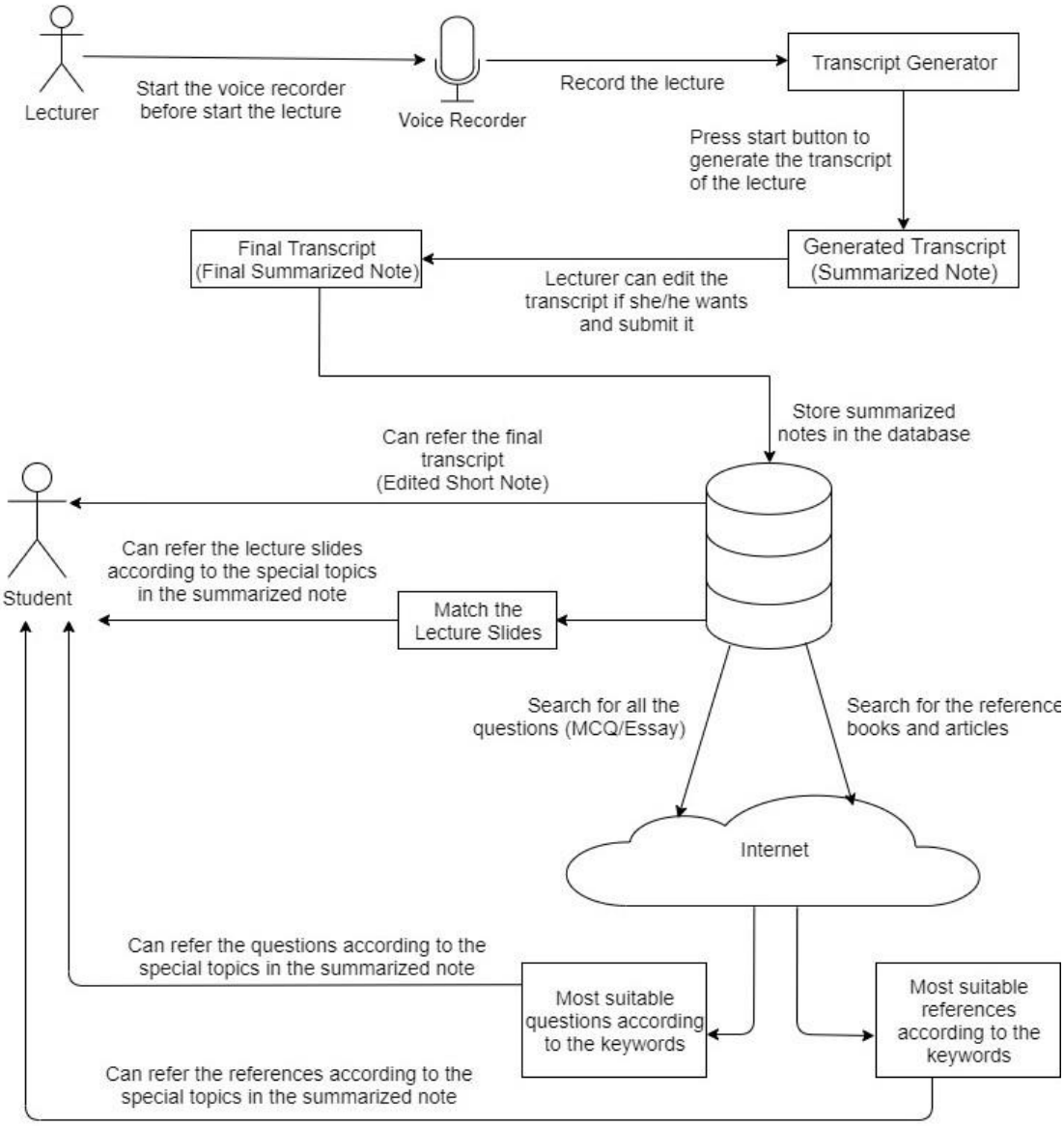

Figure 3.1 – System Architecture Diagram

3.2 Tools and Technologies

Hardware

- Google Cloud Server
- Voice recorder on laptop

Software and Tools

- Java
- Python
- Springboot Framework
- Eclipse IDE
- Apache Tomcat application server
- MySQL Database

4. DESCRIPTION OF PERSONAL AND FACILITIES

Name	Function
Jayasuriya J.A.T.P. IT17112192	• Capture the lecturer's voice using a voice recorder and automatically store it in the computer. • Develop a method to create transcript of the voice recording. • Create an algorithm to identify the important parts of the voice record and create a summarized lecture note by removing unnecessary parts, then store in the database.
P.D. Gallage IT17407458	• Retrieve the summarized notes from the database and scan/read it. • Develop an algorithm to identify important keywords/phrases from the notes and create a method to search for online references using those keywords/phrases. • Create a database to store online reference documents for offline use.
J.M.S.D. Jayasundara IT17153096	• Find questions from online resources related to the summarized lecture note. • Develop an algorithm to identify suitable keywords/phrases suitable for questions & create a method to find relatable questions from online resources. • Create interfaces to display questions in a clear way and arrange them accordingly.
L.A.P.Y.P. Nuwanjaya IT17106634	• Match the lecture slide content according to the summarized lecture note. • Create an algorithm to identify unique keywords/phrases which are common to both lecture slides & the summarized note and create a method to match lecture slides with the particular area of the note. • Develop additional interfaces and UX/UI improvement components.

Table 4.1 – Individual Functions

5. BUDGET AND BUDGET JUSTIFICATION

The budget is a detailed (unthinkable) picture of the costs related to the proposed commitment. The budget motivation contains more details about the expenses behind the details and sometimes does not clarify the use of the assets. The cost estimates should be as accurate as possible to cover the costs proposed in the commitment. The reviewers will take note of the over-and under-estimates.

The following cost estimate is to develop and test the accuracy of the proposed system project.

Task/Description	**Cost (in LKR)**
Cloud Server Cost	10,000.00
Printing and Stationary charges	2,000.00
Internet Charges	3,500.00
Miscellaneous Expenses	3,000.00
Total	**18,500.00**

Table 5.1 – Budget

We are marketing the EduEasy Smart Learning Assistant System as a fully functional smart learning application. We hope to directly contact with the Sri Lanka Ministry of Education to promote the product among government and private universities. Although our main target is university students and lecturers.

We are planning to introduce our product as packages. For time being there will 3 main packages named Sliver package, Gold package, and the Platinum package with different functionalities for different price ranges.

REFERENCES

[1] J. van der Meer, “Students' note-taking challenges in the twenty-first century: Considerations for teachers and academic staff developers”, *Researchgate.net*, February 2012. [Online]. Available: https://www.researchgate.net/publication/233335744_Students'_note_taking_challenges_in_the_twenty-first_century_Considerations_for_teachers_and_academic_staff_developers. [Accessed Feb. 14, 2020].

[2] T. Kawahara, H.Nanjo & S. Furui, “Automatic transcription of spontaneous lecture speech”, *IEEE Workshop on Automatic Speech Recognition and Understanding, 2001. ASRU '01.*, Madonna di Campiglio, Italy, 2001, pp. 186-189. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/1034618. [Accessed Feb. 14, 2020].

[3] C. Munteanu, G. Penn, R. Baecker, “Web-Based Language Modeling for Automatic Lecture Transcription”, *isca-speech.org*, August 2007. [Online]. Available: https://www.isca-speech.org/archive/archive_papers/interspeech_2007/i07_2353.pdf. [Accessed Feb. 15, 2020].

[4] K. S. Thakkar, R.V. Dharaskar, M.B. Chandak, “Graph-Based Algorithms for Text Summarization”, *3rd International Conference on Emerging Trends in Engineering and Technology*, Goa, 2010, pp. 516-519. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5698380. [Accessed Feb. 15, 2020].

[5] P. Zhang and C. Li, "Automatic text summarization based on sentences clustering and extraction," *2009 2nd IEEE International Conference on Computer Science and Information Technology*, Beijing, 2009, pp. 167-170. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5234971. [Accessed Feb. 15, 2020].

[6] J. N. Madhuri and R. Ganesh Kumar, "Extractive Text Summarization Using Sentence Ranking," *2019 International Conference on Data Science and Communication (IconDSC)*, Bangalore, India, 2019, pp. 1-3.

[7] V. Gupta, G.S. Lehal, “A survey of text mining techniques and applications”, *Journal of Emerging Technologies in Web intelligence*, VOL 1, NO 1, 6076, August 2009.

[8] G. Erkan, D.R. Radev, “LexRank : graph based centrality as salience in Text summarization”, *Journal of Artificial intelligence Research*, Re-search, vol.22, pp. 457-479 2004.

[9] CAE Team, “What is Smart Learning and why does it interest educational centers?” cae.net, 2020. [Online]. Available: https://www.cae.net/what-is-smart-learning-and-why-does-it-interest-educational-centers/. [Accessed: Feb. 21, 2020].

[10] Margaret Rouse, “cloud database,” whatis.com, Feb. 03, 2017. [Online]. Available: https://searchcloudcomputing.techtarget.com/definition/cloud-databasehttps://searchcloudcomputing.techtarget.com/definition/cloud-database. [Accessed: Feb. 23, 2020].

[11] V.P.fernoaga, G.A.Stela, and C.Gavrila, “Intelligent Education Assistant Powered by Chatbots,” In the International Scientific Conference eLearning and Software for Education, Bucharest, April, 2017.

[12] J.Feng, G.Li, and J.Wang, “Finding Top-k Answers in Keyword Search over Relational Databases Using Tuple Units,” In IEEE Transactions on Knowledge and Data Engineering, Dec, 2011, pp.1781-1794.

[13] D.Zhang, Y.M.Chee, A.Mondal, A.K.H.Tung, and M.Kitsuregawa, “Keyword Search in Spatial Databases: Towards Searching by Document,” In 2009 IEEE 25th International Conference on Data Engineering, Shanghai, China, April, 2009.

[14] L.Li, H.Wang, J.Li, and J.Luo, “Efficient Top-k Keyword Search on XML Streams,” In 2008 the 9th International Conference for Young Computer Scientists, Hunan, China, Nov, 2008.

[15] X.Somg, J.Feng, G.Li, and Q.Hong, “LCA-based Keyword Search for Effectively Retrieving “Information Unit” from Web Pages,” 2008 the Ninth International Conference on Web-Age Information Management, Zhangjiajie Hunan, China, July, 2008.

[16] V.Raval, and P.Kumar, “SEReleC (Search Engine Result Refinement and Classification) - a Meta search engine based on combinatorial search and search keyword based link classification,” In IEEE-International Conference On Advances In Engineering, Science And Management (ICAESM -2012), Nagapattinam, Tamil Nadu, India, March, 2012.

[17] A.D.Patel, and V.N.Pandya, “Web Page Classification based on Context to the Content Extraction of Articles,” In 2017 2nd International Conference for Convergence in Technology (I2CT), April, 2017

[18] A. Baid, I. Rae, J. Li, A. Doan, and J. Naughton “Toward Scalable Keyword Search over Relational Data” Proceedings of the VLDB En-dowment, vol. 3, no. 1, pp. 140–149, 2010.

[19] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan, “Keyword Searching and Browsing in Databases using BANKS,” in Proceedings of the 18th International Conference on Data Engineering, ser. ICDE ’02, February 2002, pp. 431–440.

[20] B. B. Dalvi, M. Kshirsagar, and S. Sudarshan “Keyword Search on External Memory Data Graphs,” Proceedings of the VLDB Endowment, vol. 1, no. 1, pp. 1189–1204, 2008.

[21] J. X. Yu, L. Qin, and L. Chang, Keyword d Browsing in Databases using BANKS,” in Proceedings of the 18th International Conference on Data Engineering ser. ICDE ’02, February 2002, pp. 431–440.

[22] J. Coffman and A.C. Weaver, “A Framework for Evaluating Database Keyword Search Strategies,” in Proceedings of the 19th ACM Inter-national Conference on Information and Knowledge Management,ser. CIKM ’10, October 2010, pp. Search in Databases, 1st ed. Morgan and Claypool Publishers, 2010.

[23] Y. Luo, X. Lin, W. Wang, and X. Zhou, “SPARK: Top-k Keyword Query in Relational Databases,” in Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, ser. SIGMOD ’07, June 2007, pp. 115–126.

[24] A.Baid, I. Rae, J. Li, A. Doan, and J. Naughton “Toward Scalable Keyword Search over Relational Data,” Proceedings of the VLDB En-dowment, vol. 3, no. 1, pp. 140–149, 2010.

[25] L. Logeswaran, H. Lee, D. Radev “Sentence Orderingand Coherence Modeling Using Recurrent Neural Networks” Department of Computer Science & Engineering, University of Michigan 2Department of Computer Science, Yale University, The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18).

[26] P. Shravanthi, Dr. B. Srinivasu “Semantic similarity between sentences” M.tech Scholar Dept. of Computer Science and Engineering, Stanley College of Engineering and Technology for Women, Telangana- Hyderabad, India, International Research Journal of Engineering and Technology.

[27] A. Sieg “Text Similarities: Estimate the degree of similarity between two texts”. Available (https://medium.com/@adriensieg/text-similarities-da019229c894).

[28] X. Yang, J. Liu, “Semi-supervised learning of dialogue acts using sentence similarity based on word embeddings” department of electronic engineering, Tsighua university Beijing100084, China.

[29] W. Webber, “Evaluating the Effectiveness of Keyword Search,” IEEE Data Engineering Bulletin, vol. 33, no. 1, pp. 54–59, 2010