Due 28th June, 2020

Research by Ekalaji Lynn

INTRO

Below are short summaries that I made regarding our project choice. Linked are the pages where I found the original material. For more information, please click on them.

[Student Supervision System](https://www.slideshare.net/DhrutiRanjanBag/student-supervision-system) (by Gaurav Sharma)

Gaurav Sharma describes his project as a solution to ‘digitize’ the manual way of doing things which posed a myriad of problems such as it being time consuming and slow. He proposes a system that will reduce this and have additional functions as well. His project is divided into 7 modules with the main ones being: student, supervisor, second marker and admin modules.

This project is web based and uses PHP and Microsoft SQL for the back-end and html/js for the front end.

The method used is the SDLC method with an estimated completion within 6 weeks if the work is divided between 3 people.

The methodology adopted was the Waterfall model

CON

This project seems basic, there is no smart element to it.

PRO

Doable

[Recommender Engine](https://www.researchgate.net/publication/335109398_A_Simple_Recommender_Engine_for_Matching_Final-Year_Project_Student_with_Supervisor) (by Mohammad Hafiz Ismail, Tajul Rosli Razak, Muhamad Arif Hashim, Alif Faisal Ibrahim)

These 4 students decided to approach the same problem by incorporating a recommender engine which used Euclidean distance score to demonstrate.

The language of choice was Java.

The methodology used isn’t clear but they basically divided it into 4 phases: initial prep, data gathering, data modelling and the recommender engine.

They used questionnaires as the preferred data collection method. These were circulated across the IT and CS department.

They limited the project scope to 5 major areas: Multimedia, Web application, Network, Artificial Intelligence and Mobile Application. This is where one’s project of choice would fall (broadly).

CON

I did not fully understand the doc, mostly the algorithms used

[Automated matching](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiv7_zymKTqAhWlzIUKHX7vCN0QFjAOegQIAxAB&url=https%3A%2F%2Fdl.acm.org%2Fdoi%2Fpdf%2F10.1145%2F3294016.3294030%3Fdownload%3Dtrue&usg=AOvVaw2F2JG7CXx7soW6ZKXFje5_) (by Gintare Zemaityte and Kasim Terzić)

This pair thought of automating the process altogether. Using a variety of resources at their disposal, they were able to rank supervisors and match them to students.

Some of the algorithms used are the:

Rake algorithm – used to extract keywords during the scraping process and also when students upload short proposals.

Semantic matching algorithm (simplified) – used to search for keywords associated with available supervisors. Several heuristics are applied to ensure intensive and extensive searches.

To rank supervisors, the term frequency – inverse document frequency is calculated.

The pair makes use of the NLTK library in their project.

CONS

The doc wasn’t thorough as to describe what exactly was used and how it was used.

PRO

I saw slight hints of artificial intelligence although not directly mentioned

[E-research Engine system](https://books.google.co.ke/books?id=wZRdDwAAQBAJ&pg=PA137&lpg=PA137&dq=%22system%22+to+Matching+supervisor+to+student&source=bl&ots=2mzOljwQRv&sig=ACfU3U2ehM54sznmc4XaKccuRU4EiAAR0Q&hl=en&sa=X&ved=2ahUKEwjApMjsg6XqAhW6A2MBHdyqCTAQ6AEwDnoECAkQAQ#v=onepage&q=%22system%22%20to%20Matching%20supervisor%20to%20student&f=false) (Regional Conference on Science, Technology and Social Sciences book edited by Nor Azizah Yacob, Nur Asmaliza Mohd Noor, Nor Yuziah Mohd Yunus, Rahmah Lob Yussof, Shaikh Abdul Karim Yamani Zakaria)

This is a web-based system created to assist in saving time, cost and effort in the process of finding and matching project titles and the expert area between students and supervisors. On top of that, the system guarantees more awareness of availability of supervisor.

What I found interesting is that it also provides a list of project titles for students who don’t have a proposal, possibly one that is set to capture the interest of a supervisor. The project also sheds light on how students decide on a project title (mostly through search engines) and how 8 of 10 times they get rejected by their supervisor of choice.

Methodology used: SDLC

Data collection method involved questionnaires with about 15 questions

Some improvements mentioned were:

Integration with the student portal

Extending the system to cover other departments

[Another web dev project](https://ukdiss.com/examples/student-project-allocation-monitoring-duplication.php#_Toc519976872)

This system manages the database and maintains a list of all students registered in the department. Allocation of a project is done by lecturers registered in this system through a list of topics in the department database organized by the department board.

The system design is based on three-tier architecture. The three-tier (layer) is a client-server architecture in which the user interface, registration process and data storage and access are developed and maintained as independent modules or most often on separate platforms.

From what I have gathered, the system is controlled by the admin, who has to have an account. She/he then uploads lists of supervisors and students and saves them to the database. Once this is done, students and supervisors are randomly matched through predefined functions. A list of project titles selected by supervisors is uploaded and assigned to the students.

Methodology: more or less SDLC

Tools: bootstrap (html, css, js – jquery and ajax), php and mySql

CONS

The system is not hands-on for the students, everything is done by the system. This may result in students ending up with topics they have no interest in and supervisors they don’t get along with.

Recommendations suggested in the doc:

1. Group student project allocation
2. Allowing access to various file types not only Microsoft excel
3. Better arrangement of tabs

BONUS: <https://github.com/amodig/Matching-System>