**Functional requirements.**

1. **Students** are able to upload their **proposal** to the system and the system finds the most suitable **supervisor**.
2. In case a student doesn't have a ready proposal, he/she can submit keywords of his/her project idea.
3. The system can send the matching results to the student’s personal email.
4. **Lecturers** can signup or register.
5. Lecturers can login into the system (authentication).
6. Lecturers can update their profile information.
7. Lecturers can upload their work. These include project proposals, projects, academics. papers and notes on units they are teaching.
8. Lecturers can change their password.
9. The system accepts proposals and lecturer's work in word,PDF or text file (txt).
10. The system is able to extract keywords from the student's proposal, dropping words that do not provide any meaningful information.
11. The system transforms the keywords to their lemma before it is stored in the database.
12. The system captures the term frequency and the lecturer's id on each document.
13. The system creates and stores the keywords in a hash table. The format should be {"keyword": [lecturer\_id, tf]}
14. Keywords are stored locally.
15. System should provide a list of lecturers in the order of the most appropriate. (Ranking)

**Non functional requirement**

1. The system should be sent with a latency of no greater than 5 minutes.
2. The user steps should be as minimal as possible.
3. Availability of the system should be 99.99%
4. System should be able to uphold confidentiality. Encryption of the passwords that even the system admin cannot read.
5. The system should ensure integrity.
6. The system should be compatible to most of the systems. (Web based)

**Lecturer requirements**

1. They should be able to sign up for an account.
2. They should be authenticated during login
3. They should be able to update their profile information
4. They should be able to upload their work.
5. They should be able to delete their work.
6. They should be able to change their password
7. They should be able to recover their account when they lose their credentials.

# DESIGN

## SYSTEM ARCHITECTURE

User experience

1. Lecturer
2. Student
3. Administrator

Main actions

1. a) Sign up for an account.

b) Update their profile information.

c) Upload their work.

d) Delete their work.

e) Change their password.

f) Recover their accounts

g) Login in the system

### Lecturer Registration

A lecturer would need to have an account. If they don’t then they will need to go through signup first. If they have an account, they will need to log in first to get to the dashboard to upload their work and update their profile information

1. For a lecturer to sign up, they will need to go to the website, and click on lecturer sign up at the home navigation bar.
2. Upon clicking ‘Sign Up’, they will be redirected to the lecturer signup page.
3. They will then be asked to provide a number of personal information (details).
   1. Their name.
   2. Their school email. This is to verify that the lecturer is really associated with the school.
   3. Their phone number.
   4. Their office addresses.
   5. Department they belong to.
   6. Profile picture if they want to upload it. This is for easy identification of lecturer by the student.
   7. They will be required to confirm their email addresses, whereby an email with a OTP confirmation code will be sent to them.
   8. After the confirmation of their email is successful, they will be logged in and redirected to the homepage. An account Icon will also be made available on the navigation bar, from where, they will be able to do account management functions.



### Lecturer Login

A login button present at the home navigation bar initiates the login process.



### Lecturer Upload

