

End Term Paper

Released: 18th April, 2022

Deadline: 28th April, 2022

Instructions

- You will be writing a research paper with 15% weightage instead of End semester exam.
- This has been designed to give you a nice exposure to the world of research and help you get familiar with software systems research principles.
- You will be writing a classic academia style research paper in a 2-column LATEX document, named as rollnumber.pdf. You can use the IEEE template [here](#) or the CVPR one [here](#). Note that you will be submitting the tex files as well. You can also create your own template.
- This is an individual assignment and you are free to choose whatever problem statement you like. In case you want to work on your own problem statement, contact Vikrant once for approval. For any clarifications in the given statements, please ping the concerned TA.
- Plagiarism detectors will be run on all submissions, so please do not copy from anyone. If found, you would be given a straight 0.

1 IIIT Expenses Tracking App

Concerned TA: Vikrant Dewangan

Statement:

You wake up and suddenly realise, you are ordering too much from outside. You decide to track and control your spending, and build an app which does this comprehensively. It should also duplicate as bill-tracking amongst friends. You can integrate some functionalities from any UPI-based app. Some ideas could be -

- **Simplify Debts** - Global simplify debts based on FB or IIIT CAS network integration.
- **Spending Visualization** - Display monthwise, and category-wise spending, based on it. Have a cut-off for few categories - food, travel, and send alerts when it crosses the cut-off.
- **Split Taxes** - Calculate taxes based on type of item, and split it in the ratio of original order.

While some of these features may be there on PayTM/Splitwise Pro, you need to come up with your own solution to tackle the problems. Be creative and come up with innovative solutions for this problem!

Resources:

1. [Splitwise Case Study/ Redesign](#)
2. [Software Requirements Specification for SplitPay](#)
3. [Splitwise: a UX case study](#)

2 Multimodal Search Engine for text, images and videos

Concerned TA: Bhavyajeet Singh

Statement:

The idea is simple, all you need to do is to come up with a practical and functional design for a search engine which can support multimodal inputs in the form of text, images and videos. Some of the already solved technical roadblocks can be considered as blackboxes (with references) and the main objective is to design the software system around them with all possible creative features and functionalities.

Resources:

1. [Reverse image search](#)
2. [Deepstyle](#)
3. [Wikipedia](#)

3 Attendance!

Concerned TA: Kanish Anand

Statement:

Let's move back to the most memorable activity of offline semester: the sudden increase in the class strength as Badri Bhaiya arrives and as soon as he leaves, the class strength drops like a roller-coaster.

Professors are quite fed up from this situation. They want students to listen to them during whole class and not leave in between. They don't want to lock the doors because someone may need to go out due to some emergencies. Attendance policy is also not working here because students just come, mark the attendance and then go back. So you need to provide a solution which can store when students entered, left the class and attendance will be marked based upon this data. Along with this, attendance will be shown instantly (just after the class ends) on some software system to the students so that students can plan their leaves accordingly, as well as the professors so that they can monitor which class was more boring.

You can explore the field of Computer Vision or maybe Robotics and come up with great ideas for this. All this can be treated as a black-box and main objective is to design the software system around it which would store attendance data, provide features to students to monitor their attendance and professors to monitor class attendance for various topics, show reasons for marking someone as absent.

P.S: This is not going to be implemented in reality :xD So give your ideas freely.

Resources:

1. [YoLo](#)
2. [FaceNet](#)

4 IIIT Cab Service

Concerned TA: Kartik Garg

Statement:

Lets say, there is a college 'TIII' in some metropolitan city called 'Bhagyanagar' and it is managed by someone named 'ydder'. Somehow 'ydder' takes over the city 'Bhagyanagar' and 'TIII' seeing this as a great opportunity expands its campus which was earlier considered to be a small campus. Now, the whole city enjoys their strictly followed policy of "offline classes". But there is one challenge that 'ydder' didn't think of which was how to conduct the examinations in this new expanded campus where students and faculty, enjoying the benefits of a large campus, are now residing in far away parts.

So, 'ydder' is asking you to help him design a cab service which can be used by the different Actors of this situation. The system should be efficient, should be able to accommodate a large population using the service in a given short time interval. Try to include as many as features you can.

PS: Any sort of resemblance to real life is totally unintentional and coincidental.

Resources:

The problem is fairly simple, the key will be to just start googling about System Design.

5 The IIIT Dating App

Concerned TA: Alapan Sau

Statement: Rohan have had enough of being single. But he does not want a long distance relationship. He is very hopeful and wants to find his "perfect one" in the campus. He wants to help himself and the community with a IIIT Dating Application with an excellent Recommendation system. Your task is to help him design this Dating Application. Make it innovative and catered towards IIIT. Rohan never got a single match on Tinder or Bumble and is very pissed with them. So, refrain from making another clone of either of them.

Some of the essential requirements of the application are given below. You have to explain how you are going to realize them.

Scalability: You have to make a scalable and efficient system.

Recommendation Engine: This is the heart and soul of the application. Refer to existing SoTAs and come up with your ideas of implementing a robust recommendation engine to help Rohan find his "perfect one".

Others: Login using OAuth (FB), Swipes (Left, Right), Matching, Chat, push notification, super likes etc. Feel free to add any other functionalities.

6 Recording Management in Offline Classes

Concerned TA: Pavani Babburi

Statement:

As offline classes start, the days of missing classes and binging through recordings are gone. The parliament manages to convince the authorities that a proper recording system would be beneficial. After the previous fiasco of offline recording, the students now have the responsibility to design a cheap and scalable end-to-end system for recording offline classes with minimal human intervention.

Some of the required features of the application are

1. System of smart cameras to track both the professor and the board
2. Ability to start and stop recording given the class timings
3. Automatic upload to cloud
4. Proper access for students, TAs and professors
5. A Clean UI to access recordings

You can implement any additional features such as transcribing lectures, OCR for text on board, etc.

7 Course Allocation System

Concerned TA: Abhijeeth Singam

Statement:

In IIIT, up to 3rd year, course allocation is a breeze as students don't need to pick their courses, but from 3rd year onwards, students have to choose their own electives.

During course registration, students are given a choice to select only one preference per elective. Due to the limited number of seats in each course many students don't end up with the courses they selected. Hence, they are forced to take up other electives of the same course type (which they don't like) in order to satisfy the credits requirement. This process currently involves multiple rounds and is usually done manually by the academic office and is tedious for students, faculty and administration.

You need to design an efficient course allocation software system accessible to the students, faculty and admins. You can include features like Add/Drop, student priorities from faculty, multiple preferences for each elective etc. The goal is to have maximum number of students content with the allocated courses while reducing the burden on the administration.

Resources:

1. [Student Course Allocation with Constraints](#)
2. [Course Allocation via Stable Matching](#)
3. [Centralized Course Allocation](#)

Major Sections

Your research paper must comprise of 6 sections, namely the Abstract, Introduction, Literature Review, System Architecture, Conclusion and References. We honestly don't care about how much you write, you should design your system well and convince the evaluator that you've done your research. 2.5 to 3 pages is a decent amount of content. Even if you write 5 pages, but if you don't design your system efficiently with a good architecture, then you'll lose marks.

Abstract

Just a 3-4 sentence description of your solution to the problem in hand. Something like "This is a major problem that people face and we propose XYZ, a novel software system application that will take as input yada yada yada."

Introduction

Explain briefly about your problem and the difficulties that people face because of it. You must convince the reader that the problem is indeed a major one and that coming up with a solution is paramount.

Literature Review

This is where you have to do some research and read papers and scientific articles online. You must talk about the ongoing research in the field of your problem statement and the existing technologies out there that try to solve a similar problem that you're working on. You should explore existing products in the solution space, maybe study the existing data and convince the reader that you are well versed about the ongoing research related to your problem.

System Architecture

This is the significant part of your research paper, where you design the software system and explain its overall architecture. You should explain the pipeline of your system, about the various interactions between the model, view and the controller, how the database is being updated etc. We expect a neat design that eloquently explains your system. Use classic software design principles for this, listing down the use cases and making use of UML sequence and class diagrams. You should present a nice blueprint of your system and convince the reader that it's an effective system and would have a high probability of succeeding in the real world. You don't have to code it up, maybe you can do that over the summer!

Conclusion and Future Work

Conclude well and summarise your solution in 3-4 sentences. Talk about future work and how things can be improved in the coming times.

References

Whatever you read online - research papers, scientific journal articles etc. - please cite them

Deliverables

Submit a single <RollNumber>.zip file containing all the latex files, and the necessary images/diagrams/figures needed for compilation. In case, if the images are too big while zipping, you may put them in a drive folder and provide the link in the pdf.

Please submit in your respective submission portal.