



Contribution/project goal:

The goal of our project is to implement live monitoring of people density, with a specific focus on the university library. We aim to provide accurate and real-time data to students, enabling them to make informed decisions about available seating areas within the library. By offering a fast search experience and real-time reporting. Our project aims to optimize the student experience and maximize their productivity during study sessions.

Methods/algorithms/Alternatives or Design

Considerations:

During our planning we originally planned to use raspberry pies with Wi-Fi antennas to count Wi-Fi signal transmitted by the mobile devices of the student in the libraries, but due to security features added in both Android and IOS it was impossible, as all devices send probe requests using randomized MAC addresses.

Another method is by using CCTV cameras with combination of image processing in order to determine the number of people in the filmed area, but due to costs and scaling complexity it was discarded.

We also considered using IR sensors to determine the amount of people passing by them, with the direction they are heading, but we neglected it as it couldn't provide data with the accuracy in the tables level.

Introduction:

Our project focuses on implementing live monitoring of people density, with a specific emphasis on the university library. The main objective is to provide students with accurate and real-time information regarding available seating areas within the library, enabling them to make informed decisions and optimize their study time. By developing a user-friendly website interface, students can scan QR codes placed on tables and fill in their expected duration of stay. This information is processed and stored in a secure database. The system utilizes Flask in the backend, and JavaScript and HTML in the frontend. Our solution aims to enhance the student experience by offering real-time reporting and efficient utilization of library spaces. With the ability to quickly identify free tables and the number of people occupying them, students can minimize the time spent searching for available seats and focus on their academic pursuits.

Selected Approach:

Our selected approach for live monitoring of people density in the university library involves creating a user-based system, in which the students can update their stay time and at which table they are sitting via a QR code and form submission. There is a user-friendly website interface that integrates the backend and frontend components, through which the students can get real time data about the amount of people in the different locations of the library, and decide where to sit in order to optimize their studies. We utilize Flask in the backend for efficient communication with the database and JavaScript/HTML in the frontend for a responsive user interface. A relational database is used for data storage. Additionally, an automated alerting system is implemented to notify users when their allotted time is about to end. This approach aims to provide an efficient, secure, and user-friendly system for real-time monitoring of seating availability in the university library.



For full code in github



Visit out website

Solution Description:

Data Collection: To gather data on people density, we implemented a system where students can scan QR codes placed on tables and fill out a form indicating their expected duration of stay. This data is crucial for accurately determining the availability of seating areas.

Database Management: We utilized a relational database to store and manage the collected data. This allows for efficient retrieval and processing of information, ensuring real-time reporting and quick decision-making for students.

Backend Implementation: The backend of our system was built using the Flask library in Python. Flask provides a lightweight and flexible framework for handling requests from the frontend and interacting with the database.

Algorithm for Alerting: We developed a Python script that periodically checks the database for users with less than 15 minutes remaining in their allocated time. When such users are identified, the script sends email alerts to notify them and provides an option to extend their stay if needed.