# Kathmandu University

## Department of Mathematics

## Dhulikhel, Kavre



## A Project Report

## on

“Kata Khaane”

## **Submitted by:**

GROUP F

Shambhav Rayamajhi, 18

Ruby Shrestha, 23

Yogesh Sapkota, 19

## **Submitted to:**

## Dr. Prakash Poudyal

## Department of Computer Science and Engineering

Submission Date: November 29, 2022

**Abstract:**

To avoid lunch time rush hour at KU by showing us the least crowded places one can have their lunch at. By implementation of queues, arrays and sorting algorithms, our project aims to show its users the best possible place to eat if they are looking to save time. We were facing problems with long wait times at the places to eat in and around KU, this was the main inspiration for our project. We have compiled this project through the help of Python programming language and we have done so by implementing various data types to reach our objective. Our project tackles the above-mentioned problem by giving the users real-time updates about the number of vacant seats available at the closest eateries with respect to their precise location. This helps the users spend the least amount of time for eating their food and increases their time needed for other academic activities.

**Background**: The foodservice industry is the largest retail employment segment in the world. Some

of these services available at KU are quick service, bakeries, educational services.

**Purpose**/**Aim**: Our aim through this project is to minimize the time wasted during the process of eating.

**Procedure** **and** **Method**: We aim to do so by giving our users real-time updates about the number of

seats at each of the eateries in and around KU.

**Expected Outcome**: This program will prove to be very efficient in saving a significant amount of time of the user because s\he can go to a relatively empty eatery right before leaving.

**Conclusion**: We were successfully able to finish our project the way we had planned.

**Recommendation:** We recommend that you use this program in order to tackle the problem of long

wait times at eateries.

**Acknowledgement:**

We would like to express our special thanks to our course instructor Dr. Prakash Poudyal Sir for his time and efforts he provided throughout the year. Your useful advice and suggestions were really helpful to us during the project’s completion. In this aspect, we are eternally grateful to you.

I, Shambhav Rayamajhi, would like to take this opportunity to express my gratitude to all of my group members Ruby Shrestha and Yogesh Sapkota. We three equally worked on our designated parts to bring this project to completion. The project would not have been successful without their cooperation, input and hard work. We would also like to thank our colleague Mr. Ayam for helping us with our project.

Thank you.

Shambhav Rayamajhi, Roll No: 18

Yogesh Sapkota, Roll No: 19

Ruby Shrestha, Roll No: 23

**Table of Contents:**

|  |  |
| --- | --- |
| TITLE | PAGE NUMBER |
| Acknowledgement | 2 |
| Abstract | 3 |
| Chapter 1: Introduction  1.1: Background  1.2: Objectives  1.3: Motivation and  Significance | 5  5  5  6 |
| Chapter 2: Related Works | 7 |
| Chapter 3: Design and  Implementation  3.1: System requirement Specs  3.1.1: Software  specifications  3.1.1.1: Interpreter  3.1.1.2: OS  3.1.1.3: Python Package  3.1.2: Hardware  Specifications | 8  10  10  10  10  10  10 |
| Chapter 4: Discussions on the  Achievements | 11 |
| Chapter 5: Conclusion and  Recommendation  5.1: Limitations  5.2: Future Enhancements | 12  12  12 |
| References | 13 |

**Chapter 1: Introduction**

* 1. **Background**

The foodservice industry is the largest retail employment segment in the world. In the United States, sales of food and beverages are over $700 billion and in excess of $1.2 trillion when combined with retail food supermarkets. There are many different areas of foodservice that focus on the diverse needs of customers. Some of these segments available at KU are:

1. **Quick service industry:** Fast food restaurants or normal diners are quick service operations that provide a limited menu of convenience foods requiring minimal cooking expertise. These are mostly serving typical Nepali lunch and dinner during the morning and evening (i.e. daal bhaat) along with many varieties of fusion cuisines or dishes native to different cultures and places of Nepal.
2. **Bakeries and patisseries:** Bakeries and patisseries are small single-store operations to large wholesale production facilities that serve supermarkets and other retail operations. Baker’s hours vary but often start at midnight or early mornings.
3. **Education:** Schools, colleges, and universities provide diverse dining options for students, employees, and visiting guests. They include fast food, cafeteria-style food lines, meal plan options served buffet-style, or full-service dining. Educational institutions also have catering divisions that service a wide variety of special events.

Though the places outside the KU gates claim to be quick service industries, they tend to be the exact opposite of 'quick service'. These places have really tedious service at best and this is what all college kids are trying to avoid.

It’s a pretty frustrating problem to have when your favorite place or just any place for that matter can't satiate your appetite. And there haven’t been any developments in this matter to reduce the wait times directly as such. So we've taken it upon ourselves to solve this problem the only way we know how. By avoiding such places altogether.

* 1. **Objectives**
* To avoid long lines at the slow serving diners.
* To save time by going to a faster serving diner instead.
* Knowing where to go before you even leave the comfort of your room.
* Can be extended to all types of service industries.
  1. **Motivation and significance**

We three in general have had terrible luck when it comes to wait times at these diners. And the project came to fruition as we were walking for our dinner guessing which place would have the least crowd so we could eat quickly because you don’t want to be waiting for a plate of food for too long. So we thought what if we knew which places would be crowded even before we left our rooms.And we looked if there was already such an application. When we didn’t find any we decided we would take a crack at this daunting endeavour and see what we could come up with and this project is the result of that very question, "where should we eat".

So there is no actual way to ensure faster service time unless the restaurants increase their pace 5 6 notches. But until then, the only way is to avoid crowds as much as possible. There have been no previous attempts at this problem and this is why this appealed to us cause we have been there waiting on our plates for half an hour, and this problem really hit home.

So the rudimentary objective is to save time by avoiding crowded places. How we achieve it is we first take the location of the user and the no. of people who want to eat. Then the program displays the different eateries available sorted according to distance i.e. the place closest to the user is displayed first and so on. Then it shows how many seats are packed and how many are available so you can decide which place suits your requirements the most and book your seats.

How our project differs from other attempts out there; if there are any (not that we know of one), is that our project displays the exact number of seats available so you know what you're getting yourself into if you decide to in fact go to the place you’ve picked.

**Chapter 2: Related works**

There are no such applications or programs dedicated to finding out the state of the diners at KU. And speaking of similar projects, anything that comes remotely close to ours is the vast array to booking apps that are dedicated to hotels like Trivago or booking.com etc. None that are aimed at solving the wait time issues. There are many home delivery apps all over the world but it’s a whole different segment altogether. And we all know they never deliver in time because there is just too many variables that can go awry from how crowded the restaurant you're ordering from is, to if there is a technical or staff problem at the place would further increase delivery time or the state of the traffic on the way to your house.

So I think our project is really unique as in there is nothing like it and I can take comfort in the fact that we came up with this idea and we did our best with what little knowledge and experience we had.

**Chapter 3: Design and Implementation**

Start

a window is shown with a map and a space for entering no. of people

the user inputs their precise location by left clicking on the map

the algorithm sorts out the closest places you can dine

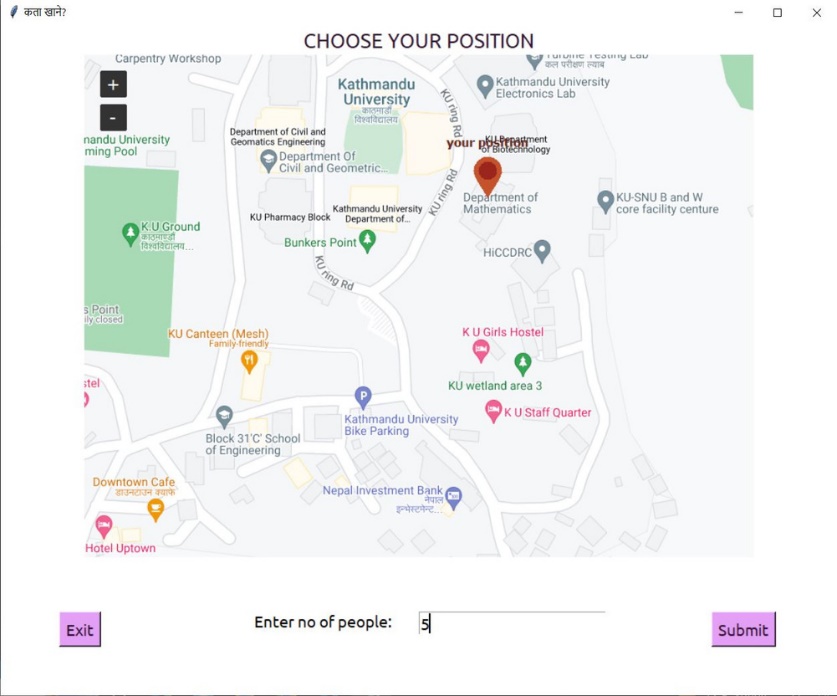
A result page is viewed with the options readily available

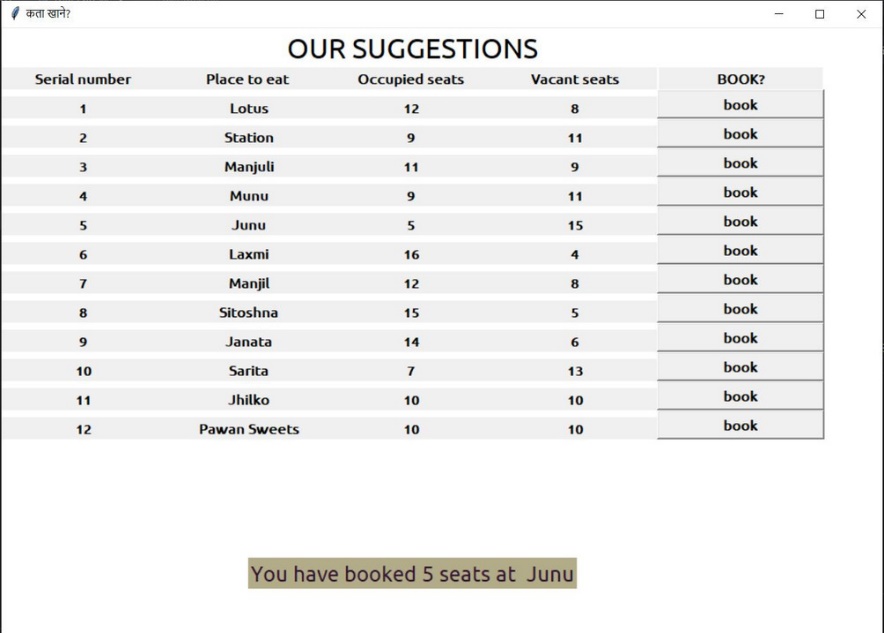
the user can choose which place to eat at and book their choice

End

**Flowchart of the program**

When we run our program, a window with a map and a box which inputs the number of user is shown. When the user clicks their position in the map, their coordinated are stored and the distance between the user's location and all the eateries are measured. Then an array is made with the help of lists that stores name, distance which was calculated before and the number of seats. The array is then sorted based on distance with the help of quick sort algorithm. The seats mentioned above is stored in a queue. Since this is just a prototype, we cannot acually enqueue the number of people in real time. So we use a random number generator to generate number of seats in the queue. This is now displayed on another page along with the name, vacant seats and the option to book. When the user books the seat, total number of seats along with the chosen location is shown.





## **System Requirement Specification**

### Software Specification

### Interpreter: Python 3 (version 3.7 recommended)

### OS: Windows XP and above or similar Linux

### Python Package: tkinter, pandas, random, numpy, queue

### Hardware Specification:

The minimum requirements of hardware needed for this program are:

**Processor:** Any Intel Core processor or similar

**Storage:** 31 MB(.exe included) of available free space

**RAM:** 500 MB or above

# **Chapter 4: Discussion on the Achievements**

Since we had to work from the scratch with no previous experience, there were lots of hurdle that we had to overcome while doing this project. A simple bug took an hour to find. Nevertheless, we had a great time and gained knowlegde both inside and outside of our course structure. We learned and understood the basics of UI using the tkinter package.

## **Features:**

1. **Starting page**: A page to select the user's position and inputs the number of people
2. **Interactive map:** If the user left clicks on a certain location of the map, the coordinates will be saved.
3. **Option to book:** A user can book among 13 places after clicking the book button.

**Chapter 5: Conclusion and Recommendation**

We basically set out to do what no one else had thought of doing before at KU, along with the fact that we wrote the codes in python which we had no previous experience of using meant we did face some problems.

**5.1. Limitations**

We initially wanted our project to be this all in one 'find a place to eat' app or interface where one could get live updates on what was happening at the places but that wasn’t possible as we tried and tried to add a refresh button to our interface but several errors impeded our attempt. So we decided to ditch that.

Also we tried to make an application for mobile ecosystem but due to lack of time and our lack of knowledge on building mobile applications, we were severely hindered.

And as in such limited timeframe, we couldn’t properly research every diner and find out the exact no. of seats and whatnot so we have fed random numbers on the vacant and occupied seats with the default no. of seats in each place being 20.

**5.2. Future Enhancements**

The possible ways in which we could further enhance our project are:

* Add the option to cancel booking.
* Find out what dishes the eateries are serving.
* Add the option for home delivery.
* Make different profiles to sort the places according to different specifications.
* We can use our project as a reference for creating other booking service apps dedicated to hair appointments etc.

**References:**

1. *Graphical User Interface with Tk.* (2022, January 27). Python Standard Library

<https://docs.python.org/3/library/tk.html>

1. *QuickSort Algorithm* - *Geeks for geeks.* (2022, September 27). Geeksgorgeeks. <https://www.geeksforgeeks.org/quick-sort/>
2. *Python Lists - W3schools.* (2018, August 3). W3schools. <https://www.w3schools.com/python/python_lists.asp>