

information system analysis – sec.07120

Getable



Fall, 2020

Nariman Mansour

LaSalle College

Table of Contents

[GeTable Summery 2](#_Toc58472671)

[Team Members: 2](#_Toc58472672)

[Our Goal 2](#_Toc58472673)

[What We Will Do? 3](#_Toc58472674)

[How Does Our Application Works? 3](#_Toc58472675)

[Long Term Goals 3](#_Toc58472676)

[Some problems (PIECES) form other application we should avoid 3](#_Toc58472677)

[Functional Requirements: 4](#_Toc58472678)

[NonFunctional Requirements: 4](#_Toc58472679)

[Feasibility Report 4](#_Toc58472680)

[Technical : 4](#_Toc58472681)

[Operational: 5](#_Toc58472682)

[INFORMATION WE COLLECT AND USE 5](#_Toc58472683)

[HOW WE SHARE YOUR INFORMATION 6](#_Toc58472684)

[Cultural: 6](#_Toc58472685)

[Cost and benefit analysis: 7](#_Toc58472686)

[Time and scheduling: 7](#_Toc58472687)

[Costs and Benefit: 7](#_Toc58472688)

[Pert Chart: 8](#_Toc58472689)

[More Details on Costs: 9](#_Toc58472690)

[Project Design 10](#_Toc58472691)

[11](#_Toc58472692)

[16](#_Toc58472693)

[16](#_Toc58472694)

[17](#_Toc58472695)

[Conclusion: 20](#_Toc58472696)

# GeTable Summery

An online restaurant(Table)-reservation application developed by GeTable team on fall 2020 and is based in Montreal, Canada. According to the team, it provides online table reservations through various platforms.

# Team Members:

Alireza

Ghazar

Mohamad Nour Al Shaar

# Our Goal

We will introduce a new system that will solve a lot of problems for both restaurants and their clients, so that we can increase the profit of the restaurant and clients will have a better experience with this system.

* Better control on reservation
* modern presentation
* save their time
* better view of reserved table

# What We Will Do?

We will do an application that can let users reserve their favourite tables before coming to the restaurants, and they can order and pay before coming to the resultants.

# How Does Our Application Works?

We will let users crate a new account in our app, and they can search about their favorite restaurants. When they chose the restaurant, they can access to the availability of the tables in the restaurant, so they can reserve and chose their favorite table, because they can see the location and picture of the table. Secondly, users can access to the menu of the restaurant, so they can order their food by our app. When they finish ordering their food, the app will show that if they want to pay by our app or in the restaurant.

# Long Term Goals

* User can reserve a table before you come to your favorite restaurant
* User can pay with this app
* User can order from the app , and see the processing
* There is a favorite dish weekly based on the numbers of likes.

# Some problems (PIECES) form other application we should avoid

* Throughput: How many orders can our server process at the same time, so we should solve their problems.
* Data is not accessible, we will **need** our app to be live, so we should have live server.
* Costs are too high in some app such as EazyDiner , and people are complaining about Membership renewal. In general, our app will be free for **user** same as Uber.
* Too much information about policy, and users are lost, so we will write our policy clearly
* Not easy to use, so we will let users just do the process in 3 tasks which are search, chose and pay.
* There is a problem for example if the customer reserve a table, and he/she does not come, so they have already reserved the table, and they will make the table busy.

Solution: we will give the period for 1 hour or 2 hours, and the customer if he/she canaled we will we will take 10$ as fees, and this is one policy of the app.

# Functional Requirements:

* It lets users search about the restaurants.
* It will send confirmations of resaving the tables and receipt to the users.
* Cancel reservation or change reservation
* Calculate the amount of the orders, and if the users cancel the reservation, it will show fees that he/she has to pay.
* It will accept just the time for the reservation for one hour or two hours, so if the user is late, it will show automatically fees.
* It will show the favorite dish weekly based on the number of likes
* It will show the processing of the cooking

# NonFunctional Requirements:

* Use encryption to avoid boys from booking
* System should accept pyments via various paymentmethods
* Easy to use , efficient and accessible

# Feasibility Report

## Technical :

We need 2 different platform one for Restaurants and one for customers.

First of all, Costs will be the biggest challenge for us, so we have to keep down the costs. In order to avoid extra costs, we will prioritize our steps. First, we will come up with an IOS app which is more common in north America for Customers, and a Windows app for Restaurants. There are many programming languages that we can use, but the best one is Swift programming language which is a modern, fast, clear, and evolving programming language. On the other hand, we can use C# programming language for the Windows app. For sure C# is not the best language, but We choose C# because we are skilled in this language and we can keep the costs down by not hiring somebody to code in other programming languages. A year after lunching the app in the market, we will introduce the web version. Why web version because not only smartphone users can use but also other people can use it. Then we will complete our mission with the Android app. In addition, we are thinking to create an app for smartwatches for reminding the time of reservation to customers.

Of course, all these steps depend on the level of customers' acceptance of this application.

## Operational:

As I mentioned in the Technical part, we need to create two different applications one for customers and one for restaurants. So, why we must create this app for restaurants? The answer is simple. Well, depending on the situations and environment of the restaurants, we need to create this additional interface in order to be always connected with restaurants. Being connected with restaurants is very necessary for us because we have to always keep updated with busy tables ,and know that when do the tables become empty again. So, designing this interface between restaurants and our company is the base of our system and this is exactly, how this system is going to help us.

In addition, this system also will have great help to restaurants to manage their tables and other customers who have not booked a table through our application.

### INFORMATION WE COLLECT AND USE

For doing this operation we need some information from the customer. Here there are more details about all information we need for a reservation.

Personal information: your name and birthday.

Contact details: E-mail address, phone number.

Dining details: your location; current and past table reservation details; and cancellations.

Payment details: credit, debit card information.

Location information: If you use our app, we automatically collect generic location information about you (city or neighborhood)

### HOW WE SHARE YOUR INFORMATION

We always try to don't share any customers' information and respect customers' privacy. All we share with restaurants is your name and your GeTable ID for confirming your reservation.

To process **payments**. We will never share your payment information with others. If you choose to order your meal by our app then we will use your payment information in order to process the payment, otherwise, your information won't use by anyone.

When your credit or debit card account information is being transmitted to our Services, it will be protected by cryptographic protocols. So, we do not have direct control over customers information.

## Cultural:

Our culture as a company is, to simplify a table reservation for customers as much as possible. On the other hand, we don't want to damage restaurants, with potential problems like a cancellation. If you want to go out to eat you have to make reservations, sometimes couple of weeks before. Imagine that you reserved a table for a very busy restaurant 2 weeks in advance to have a steak, but on that day you no longer like to have a steak and you want another meal. So what happens to your table reservation? Will you be able to reserve another table in another restaurant for that time? We solved this problem for our customers. We would like to give freedom to our customers to make 2 or 3 reservations for a specific day.

Finally, we are going to design a system that became smart enough to handle these reservations via texts, Emails and chatbots. The system works in such a way that one day before the day of your reservation, it helps you to confirm one of your reservations and cancel the rest. Otherwise, you will lose all your reservations and if this process repeated 2 times you will be blacklisted which means you are not allowed to make a reservation anymore. By doing this not only we will give freedom of choice to our customers but also we will keep restaurants away from losing revenue. So, we can say We have no penalty cancellation policy, which is the great advantage of this app.

Also, there is another point that we have to consider as a restaurant reservation company. We have some special types of customers that we have to respect them. First people who have a disability, second people who have a vegetarian diet, and finally people who tend to smoke. In order to respect these special groups, we will provide different categories which help our special customers to easily find the right restaurant.

## Cost and benefit analysis:

* Our initial thought is to charge restaurants fixed number of money per reservation ( 2.5$ ) and completely free for their costumers.
* In the upcoming years we may charge restaurants an annual fees in addition to that.
* Our goal is to start with (300 reservations /day) on the first year(109,500/year) , targeting as much restaurants as possible in the upcoming years.
* An added value of (50000 reservations /year) is what we are going to work for, aiming to target new areas and cities.
* Based on our plan, our company will break even after 3 years with a net present value of $1,865,872.76 .

## Time and scheduling:

* Our app will be ready to use on 1-Augest-2021
* Delivered after 7 months of the starting day.

## Costs and Benefit:

Table

Description automatically generated

A picture containing timeline

Description automatically generated

## Pert Chart:

Diagram

Description automatically generated

## Diagram Description automatically generated More Details on Costs:

# Use cases and users stories:

|  |  |  |  |
| --- | --- | --- | --- |
| Use case name: User register in the app to reserve a table | | Id: 1(Mohamad) | |
| Short description: The use case describes how the account of the user will be created. | | | |
| Trigger: When the user register in this app | | | |
| Type: \_\_\_ \_\_\_ External\_\_\_\_\_\_\_\_ | | | |
| Major Inputs | | Major Outputs | |
| Description | Source | Description | destination |
| 1. Some required fields from new user  2. Email and password from | 1. by Users | 1. Acceptable message  2. Error message  3. A new account | 1. The system  2. The system |
| Major steps performed | | Information for steps: | |
| 1. The user will fill in the required personal information such as Full Name, email, and his/her favorite type of restaurant. | | The account will be created, and we need a register form | |
| 2. Validation, so if the user is already existing, it will show will not allow to create | | Error messages and with red field | |
| 3. If the system accecpt the infomration, it will show a confermation message. | | The system will send the conformation on the email of the user | |
|  | |  | |
|  | |  | |

# **2-**

|  |  |  |  |
| --- | --- | --- | --- |
| Use case name: Search for restaurants | | Id: 2(Mohamad) | |
| Short description: This use case describes how a user can  search about his/her favorite type of restaurant | | | |
| Trigger: when the user chooses the type of restaurant from the checkbox that he/she wants, or they fill the textbox by the name of the restaurant or its location, and they click to see the results | | | |
| Type: \_\_\_ \_\_\_ External\_\_\_\_\_\_ | | | |
| Major Inputs | | Major Outputs | |
| Description | Source | Description | destination |
| 1. Chose the type of the restaurant  2. Type the location or the name of the restaurants | 1. User  2. User | 1. It will show the list of the near and the far restaurants  2. It will show the restaurant with its information | 1. User  2. User |
| Major steps performed | | Information for steps: | |
| 1. If the user does select the type of the restaurant, and he/she clicks search. | | The system will show a message with red line that says “Please select the your favorite restaurant” | |
| 2. If the user writes the name or the location of the restaurant with mistakes, it not will show results | | The system will show an error message that there is no restaurant with this location or name | |
|  | |  | |

**3-**

|  |  |  |  |
| --- | --- | --- | --- |
| Use case name: reserve a table | | Id: 3 (Alireza) | |
| Short description: The user will select a restaurants and reserve a table in that restaurant | | | |
| Trigger: User must select one restaurant from his/her favorite restaurant list or by searching its name or location | | | |
| Type: External | | | |
| Major Inputs | | Major Outputs | |
| Description | Source | Description | destination |
| 1.restaurant’s name, location  2.number of seats  3.table number | 1.database, search engine  2.user  3.database | 1.confermation message  2. confirmation code | 1. app screen  2. user’s email |
| Major steps performed | | Information for steps: | |
| 1. find the restaurant | | Get all restaurants corresponding to the name | |
| 2. Show the map with tables | | Get all tables info from data base | |

**4-**

|  |  |  |  |
| --- | --- | --- | --- |
| Use case name: Reschedule | | Id: 4(Alireza) | |
| Short description: User can reschedule or edit his reservation | | | |
| Trigger: User fills new form and makes changes in the reservation. | | | |
| Type: External | | | |
| Major Inputs | | Major Outputs | |
| Description | Source | Description | destination |
| 1. Number of seat for table  2. Table place  3. Date and Time | 1. Textbox(user)  2. Map(user)  3. Pick date box(user) | 1. Confirmation message  2. show the New page  3. Send Email | 1. User  2. user  3. Email to user |
| Major steps performed | | Information for steps: | |
| 1. Request for reschedule a reservation | | Take all information about this reservation from data base by click | |
| 2. Fill the requested field like date time | | Select table on map, pick date time, and numbers of people | |
| 3. Click on button update request | | Update all info for this reservation on database | |

**5-**

|  |  |  |  |
| --- | --- | --- | --- |
| Use case name: manage reservations | | Id: 5 ghazar | |
| Short description: restaurant owners will be able to manage their reservations | | | |
| Trigger: restaurants must upgrade the availability of tables | | | |
| Type: External | | | |
| Major Inputs | | Major Outputs | |
| Description | Source | Description | destination |
| 1. table available  2. table unavailable | 1. the restaurant  2. the restaurant | 1. update the database | 1. database  2.  3. |
| Major steps performed | | Information for steps: | |
| 1. change tables availibilty | | Tables availability should be updated constantly | |
| 2. database should be synchronized with table’s availability | |  | |
| 3. | |  | |

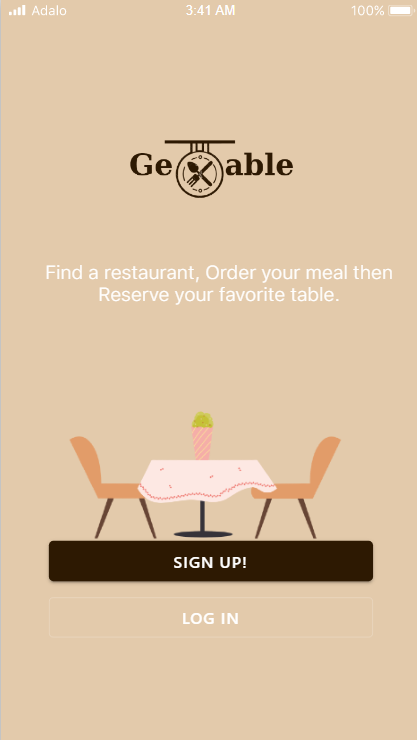
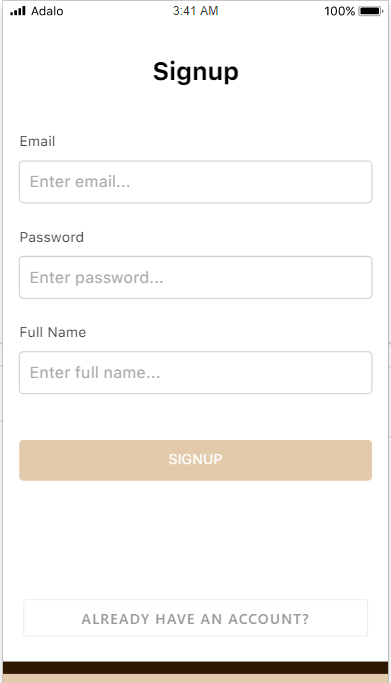
**6-**

|  |  |  |  |
| --- | --- | --- | --- |
| Use case name: add restaurants | | Id: 6 | |
| Short description: it will allow restaurant owners to add a new restaurant | | | |
| Trigger: owners must add their restaurants to update our database | | | |
| Type: \_\_\_ \_\_\_ External | | | |
| Major Inputs | | Major Outputs | |
| Description | Source | Description | destination |
| 1. restaurant name  2. restaurant location  3. menu | 1. restaurants  2. restaurants  3. restaurants | 1. update database  2.  3. | 1. database  2.  3. |
| Major steps performed | | Information for steps: | |
| 1. add new restaurants | |  | |
| 2. update the database | |  | |
| 3. catagorize the restaurants | |  | |

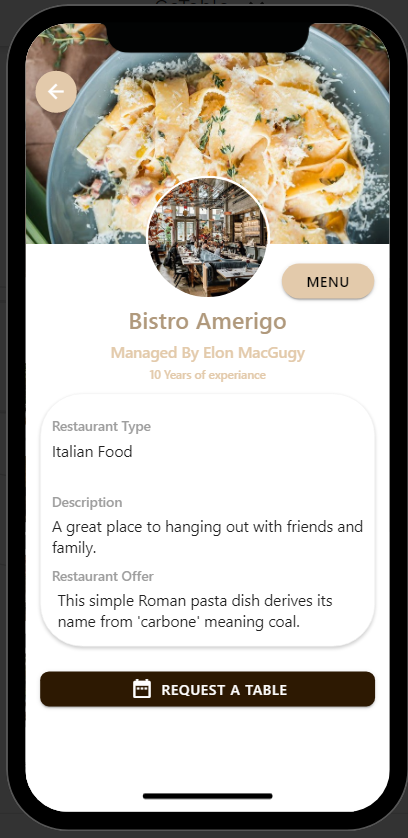
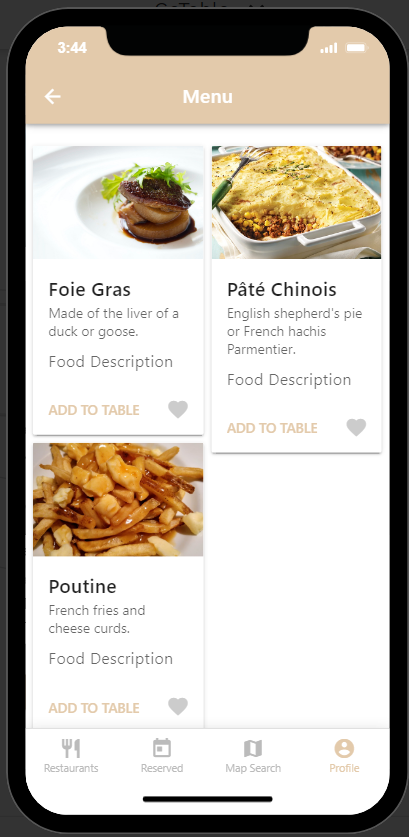
## User story:

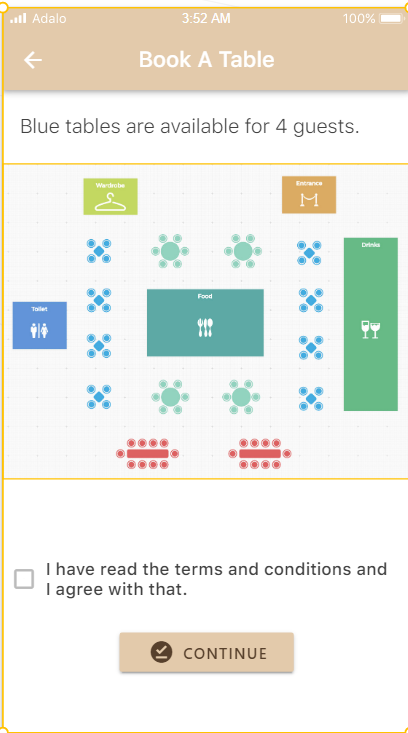
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | 1. As a user | I want to order food by the app | So, I can track the processing of cooking, before I arrive | By ordering meal, the app will direct you to page of processing | | 1. As a user | I want to reschedule my reservation | So I can change the date of my reservation and time | By Clicking on change button when you reserve , you can reschedule your reservation | | 1. As a user | I want to pay may meal in the app. | So I can have a safe payment and save my time. | By ordering meal, app will direct you to payment page. | | 1. As a user | I want to find restaurant on map | So I can find out the distance, and find nearest restaurants. | Can search by Map on the separate page. | |  |  |  |  | |  |  |  |

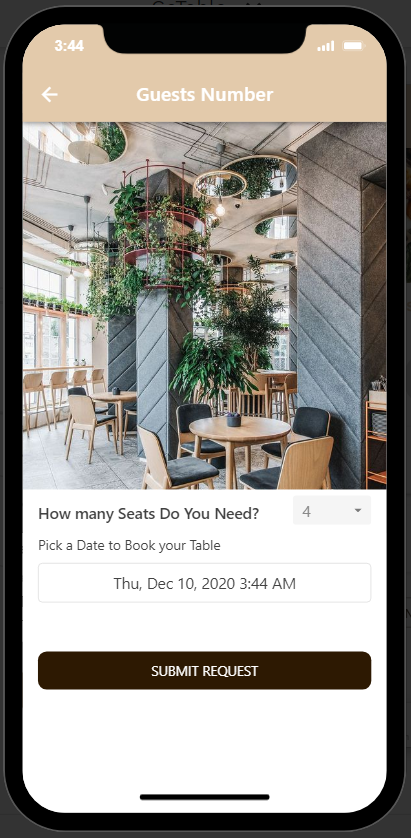
# Project Design

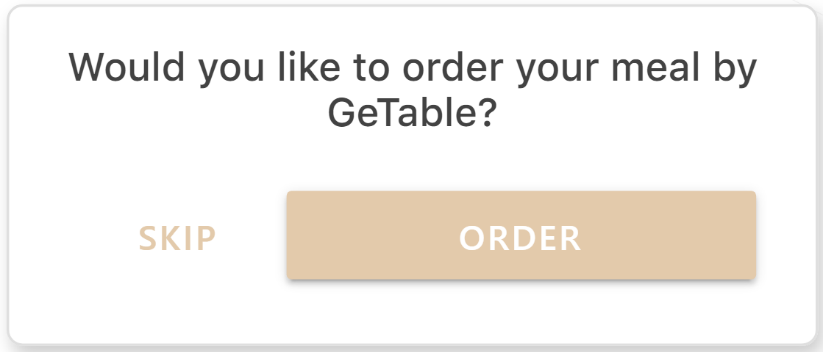


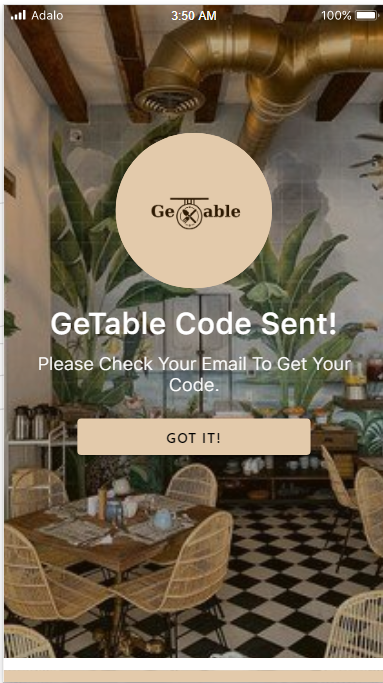
# 









Graphical user interface, application

Description automatically generated

Graphical user interface, application, chat or text message

Description automatically generated

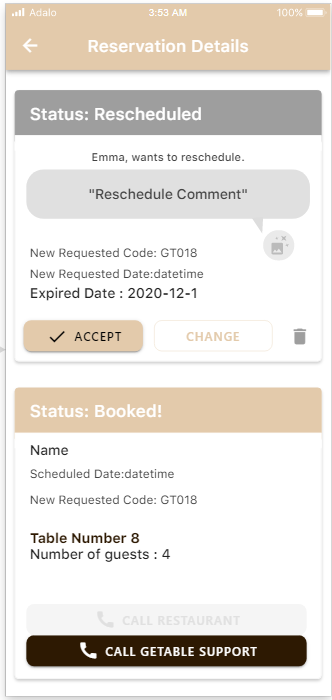
Graphical user interface, application

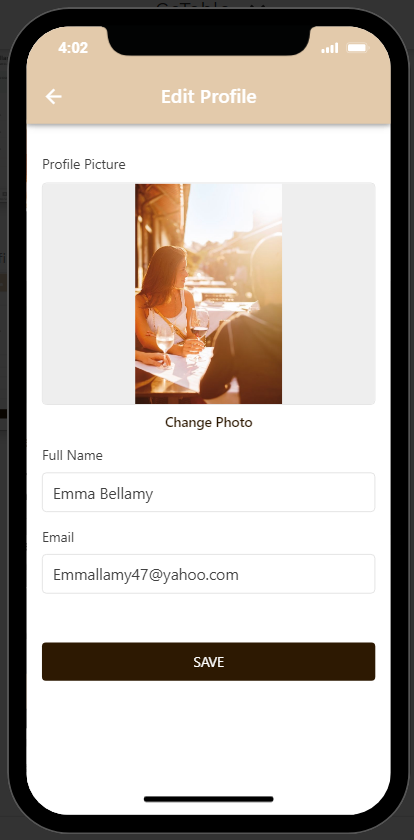
Description automatically generated

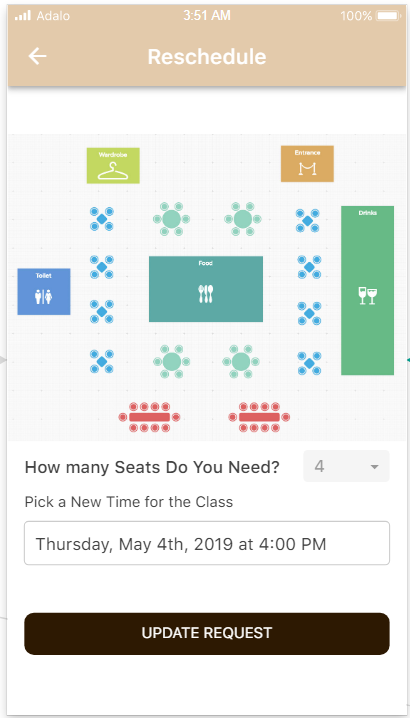
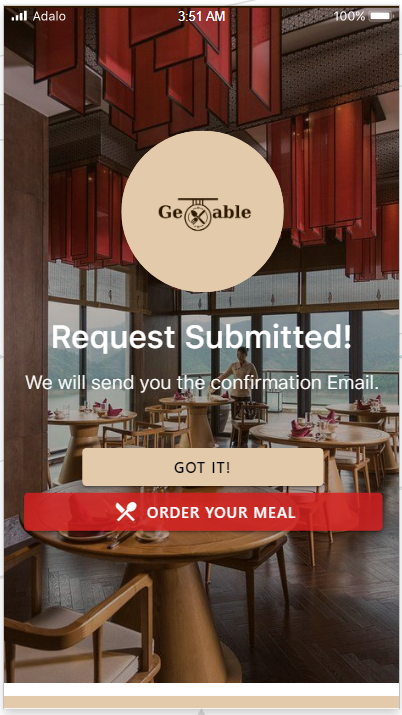
# 

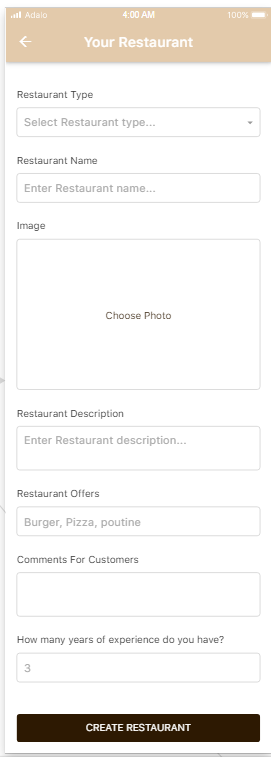
# Graphical user interface, application Description automatically generated

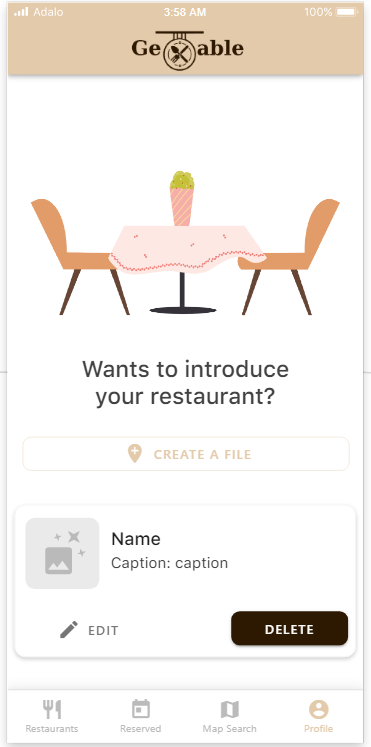
# 

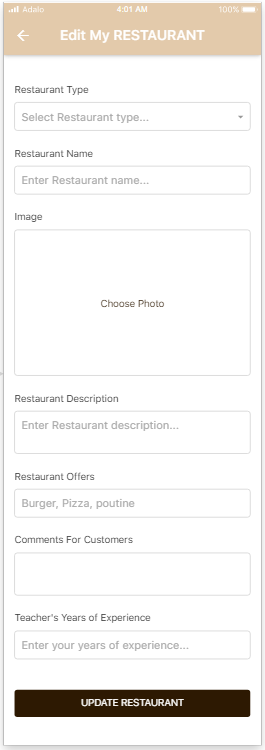












# Conclusion:

In conclusion is GeTable a good value for restaurants? In a word, yes.

We are not thinking to change our customer’s behaviour, we are thinking of coming up with a solution. In addition to helping restaurants run their operations, GeTable helps restaurants attract incremental guests.

Thanks

GeTable Team