

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
THE UNIVERSITY OF TEXAS AT ARLINGTON**

**PROJECT CHARTER
CSE 4316: SENIOR DESIGN I
SUMMER 2023**



**TEAM O
GOURMETBOOK**

**BEREKET AYALEW
BHUMIKA SHRESTHA
HYEONJUN AN
RAKSHAV PATEL
REETY GYAWALI**

REVISION HISTORY

Revision	Date	Author(s)	Description
0.0.1	07.04.2023	HA	document creation
0.0.2	07.04.2023	HA	content 1, 2, 3, 4, and 5 edits
0.0.3	07.05.2023	RG	content 13 edits
0.0.4	07.05.2023	BS	content 7 edits
0.0.5	07.05.2023	BS	Mock UI design
0.0.6	07.07.2023	HA	content 6, 8, 9, and 10 edits
0.0.7	07.07.2023	RG	content 14 edits

CONTENTS

1 Problem Statement	6
2 Methodology	6
3 Value Proposition	6
4 Development Milestones	6
5 Background	7
6 Related Work	7
7 System Overview	8
8 Roles & Responsibilities	11
9 Cost Proposal	11
9.1 Preliminary Budget	11
9.2 Current & Pending Support	11
10 Facilities & Equipment	12
11 Assumptions	12
12 Constraints	12
13 Risks	12
14 Documentation & Reporting	13
14.1 Major Documentation Deliverables	13
14.1.1 Project Charter	13
14.1.2 System Requirements Specification	13
14.1.3 Architectural Design Specification	13
14.1.4 Detailed Design Specification	13
14.2 Recurring Sprint Items	13
14.2.1 Product Backlog	13
14.2.2 Sprint Planning	13
14.2.3 Sprint Goal	13
14.2.4 Sprint Backlog	14
14.2.5 Task Breakdown	14
14.2.6 Sprint Burn Down Charts	14
14.2.7 Sprint Retrospective	14
14.2.8 Individual Status Reports	14
14.2.9 Engineering Notebooks	14
14.3 Closeout Materials	14
14.3.1 System Prototype	14
14.3.2 Project Poster	15
14.3.3 Web Page	15

14.3.4 Demo Video	15
14.3.5 Source Code	15
14.3.6 Source Code Documentation	15
14.3.7 Hardware Schematics	15
14.3.8 CAD files	15
14.3.9 Installation Scripts	15
14.3.10 User Manual	15

LIST OF FIGURES

1	High level overview	10
2	UI design	10
3	Sprint 1 burn down chart	14

1 PROBLEM STATEMENT

What COVID-19 has changed in the past few years is not limited to the change in people's perspective on public health. People are transitioning from "the closely-related and physical world" to "the remote and virtual world". In a way, people are currently living in a hybrid world of the real and remote.

Therefore, the huge success and growth of the delivery platform companies such as DoorDash and UberEats are worth noticing. Similarly, the noticeable potential is found in table reservation applications. People no longer want to wait in a long line or sit outside to get into a restaurant. They started to value their time over money and the restaurant owners always want to make the best profit out of the given hours. However, the solutions already in the market provide either too many or too less functionalities. And the consequence is on both customers and restaurant owners to get lost in their apps because they cannot find what they are looking for in either ways.

GourmetBook will provide a simple and easy reservation environment for the customers and an "at-a-glance" restaurant management helper for the owners. This application is aimed to provide compact and concise functionalities to navigate the users to more quickly find what they desire to do.

2 METHODOLOGY

GourmetBook is a Mobile Application that will be available on both Android, iOS, and possibly Web. It will provide two different user screens. One is for customer reservations, and the other is for the restaurant owners to help them to manage their restaurants. Minimum but essential functionalities will be provided to the customer screen and simple restaurant management service will be provided to the restaurant owners.

3 VALUE PROPOSITION

- "**Customers**" are no longer dealing with excessive and unnecessary features for a table reservation.
- "**Restaurant Owners**" do not have to focus on and pay for the unnecessary features to view their restaurant status.
- "**Developers**" can learn how to develop platform app and APIs for payments and data analytic.

4 DEVELOPMENT MILESTONES

- Project Charter first draft - July 7th, 2023
- System Requirements Specification - July 25th, 2023
- Architectural Design Specification - August 10th, 2023
- Detailed Design Specification - September 2023
- Demonstration of <Finalized UI/UX Implementation> - September 2023
- Demonstration of <Payment API Implementation> - October 2023
- Demonstration of <Database Design and Implementation with Server Connectivity> - October 2023
- Demonstration of <Implementation of Extra Functionalities> - November 2023
- Demonstration of <Bug Fixes and Quality Assurance Check> - November 2023
- Final Project Demonstration - December 2023

5 BACKGROUND

When people want to make their day special with someone loving or that means a lot, they often go out for a fancy dinner. People then either reserve their tables and menu through phone calls a few days before D-day or wait in front of the restaurant with their names on the waitlist; however, situations are not always optimal and sometimes might end up ruining their moods after wasting their time. The phone calls to the restaurant are either not picked up or never answered. Your reservation information gets mixed up or omitted. Even more, people have waited for longer than an hour just to get into the restaurant.

So, people who are now living a "hybrid life between reality and remotely virtual life" get online and make their reservations. Their information is saved, transmitted to the server, and safely gets to the hand of the restaurant management. Then, what is wrong with that? Most of the nationwide/world-famous table reservation apps and systems are flooded with unnecessary features and too many restaurants that are most likely not attractive options, and it sometimes makes the users feel get lost. Moreover, users can only look at the menu; they cannot include their menu selections in their reservation information.

In addition, the restaurant owners do not receive accurate and meaningful data analysis of how their restaurant is run on a daily or monthly basis. The restaurant owners want to receive not only feedback from the customers but also numerical data that they can visually absorb and utilize to help their business flourish even more. It does not have to be a hundred pages long report, but a simple page of what the customers like and dislike, when the restaurant gets busy, and such will suffice.

GourmetBook is designed with strained good features of the precedent table reservation apps and to pinpoint the specific target users. Each of the users and restaurant owners will be able to find what they are looking for through this application.

6 RELATED WORK

Referential Products in the Market

i. OpenTable

Pros:

1. It integrates with many restaurant POS systems.
2. It offers a robust set of features, including reservation management, table management

Cons:

1. It is potentially more expensive than other options.
2. Although the user interface is generally well-reviewed, some users find it less intuitive or harder to navigate than other platforms.

ii. ResyOS

Pros:

1. Offers a wide range of features, including reservation and table management, and a waitlist system.
2. Well-regarded for its user-friendly interface and efficient customer service.

Cons:

1. Some restaurants may find the cost of ResyOS to be a drawback.
2. While it's generally well-reviewed, there have been some complaints about occasional glitches and limitations in its customization options.

iii. TableIn

Pros:

1. It is simple and easy to use.
2. It is more affordable pricing compared to some of the other options.

Cons:

1. It may not integrate as seamlessly with other systems as some alternatives.
2. It lacks some of the advanced features provided by larger platforms.

iv. Eat App

Pros:

1. It offers comprehensive features for table management and reservations.
2. It includes a customer relationship management (CRM) system and various analytics tools.
3. It is a simple and intuitive interface.

Cons:

1. It lacks the depth of customization offered by other platforms.

v. Yelp Reservation

Pros:

1. It leverages the popularity of Yelp to attract customers.
2. It offers basic table management and reservation features and integrates with Yelp Waitlist.

Cons:

1. It is limited customization options and a lack of some more advanced features.

7 SYSTEM OVERVIEW

The application is designed to help users to save time, effort and get connected virtually in this rapidly growing hospitality business. Users can get the best customer service experience from a click entirely through their own devices rather than being on a call for hours. The system contends major features that are straightforward and user-friendly. The system will be implemented with the following major component

i. User Interface(UI): It is an interface where users can interact with the application. The major features can be seen on the home screen which includes, a list of restaurants, restaurant availability, events nearby, promotions, and so on. The UI will be intuitive, visually appealing, and responsive to ensure a seamless user experience.

ii. Application server/backend: This component communicates with the database, external APIs, and other services to fetch and process data. It handles user requests, manages reservations, and checks and updates the availability of restaurants and events selected.

iii. Database: The database stores all the relevant data for the system, such as restaurant and event information, reservations, waitlists, and payments.

iv. Reservation Management: This component handles the reservation process, allowing users to select their preferred restaurant or event, choose available time slots, and complete the reservation. It communicates with the backend and external systems to ensure the reservation details are properly stored and updated in the database.

v. Payment gateway: This component handles the payment API where the user puts in their information and securely transferred it to an authorized bank for successful payment. This feature applies when users do not show up during their reservation in the form of a penalty.

vi. Confirmation: This component communicates between the application server and the user whether the reservation is confirmed or not. Once a request is sent to an application server, it confirms the availability and other primary requirement and sends a confirmation notification in the form of email, sms, or notification on the application itself.

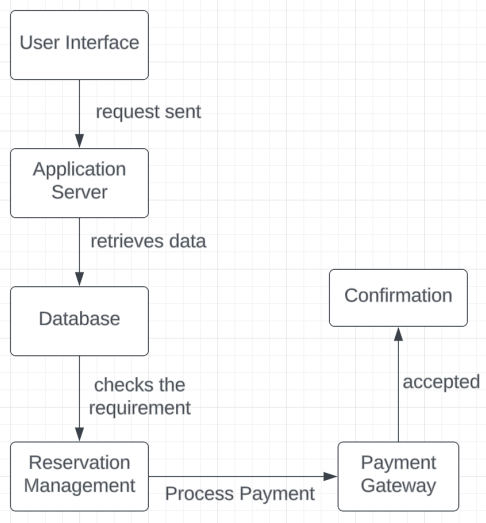


Figure 1: High level overview

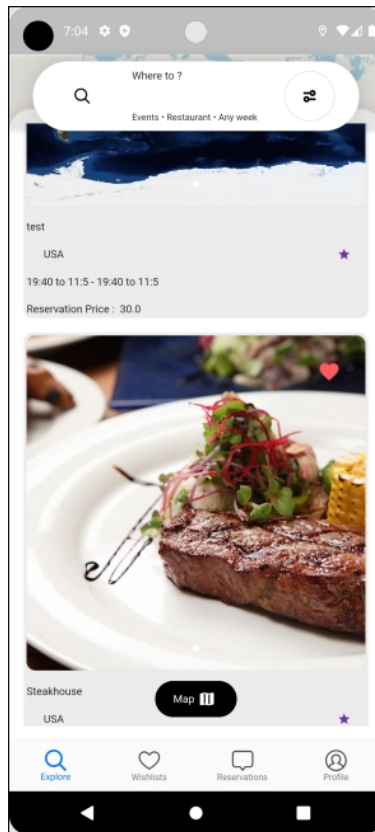


Figure 2: UI design

8 ROLES & RESPONSIBILITIES

- **Stakeholders**

- Restaurant Customers
- Restaurant Owners
- Payment Gateway Provider
- Developers (Project Participants)

- **Individual Member Responsibility**

- **BEREKET AYALEW** will be participating in both front-end and back-end development depending on the priority at the point in time.
- **BHUMIKA SHRESTHA** will be in charge of the UI designs and wireframe, which determines the main flow of the application system.
- **HYEONJUN AN** will be in charge of the main back-end development including server integration and database setup.
- **RAKSHAV PATEL** will be participating in both front-end and back-end development depending on the priority at the point in time, major research, and reference search.
- **REETY GYAWALI** will be participating in both front-end and back-end development depending on the priority at the point in time and keep track of development progress throughout.

- **Scrum Master**

Scrum master role will change periodically depending on what task the team is on at the point of the time. For example, during UI / UX finalizing phase, "Bhumika Shrestha" will be the Scrum master for the team and lead the overall work progress.

9 COST PROPOSAL

9.1 PRELIMINARY BUDGET

Title of Item	Minimum Expected Expense per month (\$)	Maximum Expected Expense per month (\$)
Server and Database Maintenance Fee	5.00	20.00
Payment Gateway (API) Fee	0.00	20.00

Table 1: Preliminary Budget

9.2 CURRENT & PENDING SUPPORT

Title of Item	Spending in July (\$)	Spending in August (\$)	Spending in September (\$)	Spending in October (\$)	Spending in November (\$)	Subtotal (\$)
Server and Database Maintenance Fee	0.00	0.00	0.00	0.00	0.00	0.00
Payment Gateway (API) Fee	0.00	0.00	0.00	0.00	0.00	0.00
Total Spending (\$)	0.00					

Table 2: Current and Pending Spendings

10 FACILITIES & EQUIPMENT

During the development of the application, the team requires to have:

1. Meeting Space

Most of the team meetings will be through the Discord channel, but if needed, physical meeting space is required in addition to anything to display the shared ideas, such as TV screen, whiteboard, etc.

2. Individual Laptops

3. Server System

Either a Web-hosted server or physical server implementation will be required at a point of development. The options are the following:

- (a) Hosting through a web hosting service
- (b) Server setup with a Raspberry Pi through port forwarding

11 ASSUMPTIONS

The following list contains critical assumptions related to the implementation and testing of the project.

- We will have a base platform or environment ready in coming days.
- We will get started with implementing are GUI design by 2nd sprint.
- The basic application with home screen would be running by 3rd sprint.
- By this time we can try adding features to this application and testing them out.
- By the end of sprint 4 or 5 we can have are list of restaurant added.
- We could begin with the back end work like setting up reservation and conformation email.

12 CONSTRAINTS

The following list contains key constraints related to the implementation and testing of the project.

- Final prototype demonstration must be completed by December 5th, 2023
- Total development costs must not exceed \$800
- Give the customer their desired product by making sure every feature and design were implemented as they wisied it to be.
- All the payments are authorized by bank. And are secured on are back end.
- Limited storage space in the database

13 RISKS

The following high-level risk census contains identified project risks with the highest exposure. Mitigation strategies will be discussed in future planning sessions.

Risk description	Probability	Loss (days)	Exposure (days)
Security breach of sensitive user and admin information	0.50	20	10
Issues with third-party API integration and functionality	0.50	10	5.0
Features not tested or implemented correctly	0.70	15	10.5
Insufficient resources to implement certain features	0.30	20	6.0
Not maintaining or updating certain features	0.60	15	9.0

Table 3: Overview of highest exposure project risks

14 DOCUMENTATION & REPORTING

14.1 MAJOR DOCUMENTATION DELIVERABLES

14.1.1 PROJECT CHARTER

The project charter has been shared with all team members on Overleaf. We will update it if there are any major changes. The initial version will be delivered July 7, 2023. Final version delivered December 2023.

14.1.2 SYSTEM REQUIREMENTS SPECIFICATION

We will create a latex document on Overleaf and share it with all team members. We will update it frequently as we come up with more functionalities for our app. Initial version delivered July 25, 2023. Final version delivered December 2023.

14.1.3 ARCHITECTURAL DESIGN SPECIFICATION

We will create a latex document on Overleaf and share it with all team members. We will update it only if there are any major changes to the structure of the app. Initial version delivered August 10, 2023. Final version delivered December 2023.

14.1.4 DETAILED DESIGN SPECIFICATION

We will create a latex document on Overleaf and share it with all team members. We will update it frequently as we change how we implement the design of the app. Initial version delivered September 2023. Final version delivered December 2023.

14.2 RECURRING SPRINT ITEMS

14.2.1 PRODUCT BACKLOG

We will likely use group vote to prioritize items from the SRS and add to the product backlog; it depends on each of our strengths and what is important to implement. We will maintain product backlog online.

14.2.2 SPRINT PLANNING

Each sprint will be planned by looking at how well we have done on are previous sprint and by what pace everyone is working on. also depends on availability of each member of are group. I believe there are 8 sprint section during combined of 2 semester.

14.2.3 SPRINT GOAL

We could plan out a specific time to meet with entire group online or in person and discuss are sprint goals. we can involve the customer by showing them what we worked on during this sprint and also ask them if they are satisfied with our work done. we can also ask the customer if they need anything

changed. Asking them if they need any changes during sprints is very important because if this question was to be asked at end of the project it would be difficult to modify it.

14.2.4 SPRINT BACKLOG

At the start of each sprint, scrum lead will assess the team's progress and push product backlog items into the sprint backlog accordingly. We will maintain the backlog on Discord.

14.2.5 TASK BREAKDOWN

This will depend on everyone's skill set. We as a group are making sure that no one works with their discomfort unless we don't have anyone to do it. So, yes each team member to voluntarily claim task. The time spent on tasks will be documented on an individual sprint report later we could check on everyone if they are truly updating their individual sprint report by adding how much time they have worked on the task.

14.2.6 SPRINT BURN DOWN CHARTS

This burn down chart was generated on Excel.



Figure 3: Sprint 1 burn down chart

14.2.7 SPRINT RETROSPECTIVE

At the end of each sprint cycle, we will share what we worked on and our progress. We will also discuss any issues that came up and any changes that need to be made for the next sprint. We will document the tasks completed and time spent (in hours).

14.2.8 INDIVIDUAL STATUS REPORTS

Throughout each sprint cycle, we will keep track of what tasks we are responsible for. At the end of each sprint cycle, we will share what we each worked on in the team Discord server. Other key information to include is the time spent on each task (in hours) in order to create the burn down chart.

14.2.9 ENGINEERING NOTEBOOKS

We do not have physical or digital engineering notebooks. Instead, we will document our progress through posts on our team's Discord server.

14.3 CLOSEOUT MATERIALS

14.3.1 SYSTEM PROTOTYPE

The final system prototype will include all front-end and back-end components of our app. We will demonstrate our app to the customer on an internet-connected device in December 2023. We will

likely conduct a Prototype Acceptance Test (PAT) with our customer for feedback.

14.3.2 PROJECT POSTER

A project poster would include visuals of the app's system overview, technologies used, and key information about the back-end design. Size will be 40" by 32" and due date is December 1, 2023.

14.3.3 WEB PAGE

Since we intend to build a cross-platform app, our project web page will just be our app. It will be accessible to the public online, delivered to the customer by December 2023, and updated throughout the project.

14.3.4 DEMO VIDEO

This will be a walkthrough video that guides a user through all of the app's main features. This could be 3-5 minutes long.

14.3.5 SOURCE CODE

We will use GitHub and git to maintain our source code. We will transfer the repository with the source code to the customer, who will decide whether to open source the project or not.

14.3.6 SOURCE CODE DOCUMENTATION

Everyone is responsible for commenting the code they implement, so it is easy to read and understand.

14.3.7 HARDWARE SCHEMATICS

None; this project does not involve hardware.

14.3.8 CAD FILES

None; this project does not involve mechanical design.

14.3.9 INSTALLATION SCRIPTS

No prior installation required. The customer can easily access the app on the internet or an app store.

14.3.10 USER MANUAL

If time permits, we will implement an in-app "Help" section with tips for new users.

REFERENCES