

Design Document

The Guest Streamliner 9000™

Prepared by: Ruben Orozco, Marshall Brister, Oren Cummings, Jason Jenkins, Emalee Keesler and Chris Adams.

The McJore's Software Engineering Firm of Fun™

September 2019 - December 2019

Table Of Contents

Table Of Contents	1
1. Introduction	2
1.1. Goals and objectives	2
1.2. Statement of scope	2
1.3. Software context	2
1.4. Major constraints	3
2. Data and Component Design.	3
2.1. Internal Software Structure	3
2.2. Component Rationale	3
2.3. Database Description	4
3. Architecture Overview	4
3.1. Architectural Boundaries	4
3.1.1. Level 0 Context Diagram	4
3.1.2. Rationale for the Level 0 Context	4
3.1.3. Level 1 Context Diagram	5
3.1.4. Rationale for the Level 1 Context	6
3.2. Use Case specification achievement	6
3.2.1. Below is a list of the use cases and how the system will handle each.	6
3.3. Outside system requirements	11
4. Restrictions, Limitations, and Constraints	11

1. Introduction

1.1. Goals and objectives

The goal of the Guest Streamliner 9000™ is to streamline the guest experience at Ivancic's Italian by allowing guests to make online or over the phone reservations and storing them in a system that will automatically manage them and allow the host/hostesses to mark them as "completed" after the guests are sat or "no show" if the guest does not show up within 15 minutes of their reserved time. The Guest Streamliner 9000™ will also be used to allow customers to order and pay from their tables, to further enhance the guest experience by giving them the choice to not wait on their server. The Guest Streamliner 9000™ will also allow workers to see what tables are available and what tables need to be cleaned to minimize the guest's wait time and improve overall guest experience.

1.2. Statement of scope

This software can be split into three different parts. The first part handles the reservations the customers will make. The second part handles the staff side of things. Lastly, the last part is what the customers will interact at the time of them being seated. The first and third parts have inputs from customers, therefore it should be handling customer interactions automatically. The second part will have its input from the wait staff, kitchen staff, and bussers.

The goals of this system is to provide the users a simple and easy to understand process to get through. If successful, the turn over time for a table will greatly improve. The reservation system being automated will make it faster to find open time slots. The customers being able to order and ask for their receipt at their own table gives waiters/waitresses more chance to serve other tables at the same time.

1.3. Software context

The software is to be placed on iPad minis on each table of the restaurant, their intended use is by the customers of the restaurant to place/pay for orders without the need for a waiter/waitress.

It will also be set on an online website that customers can go to in order to make their reservations. This system ties into the reservation list the host would be able to see at any time.

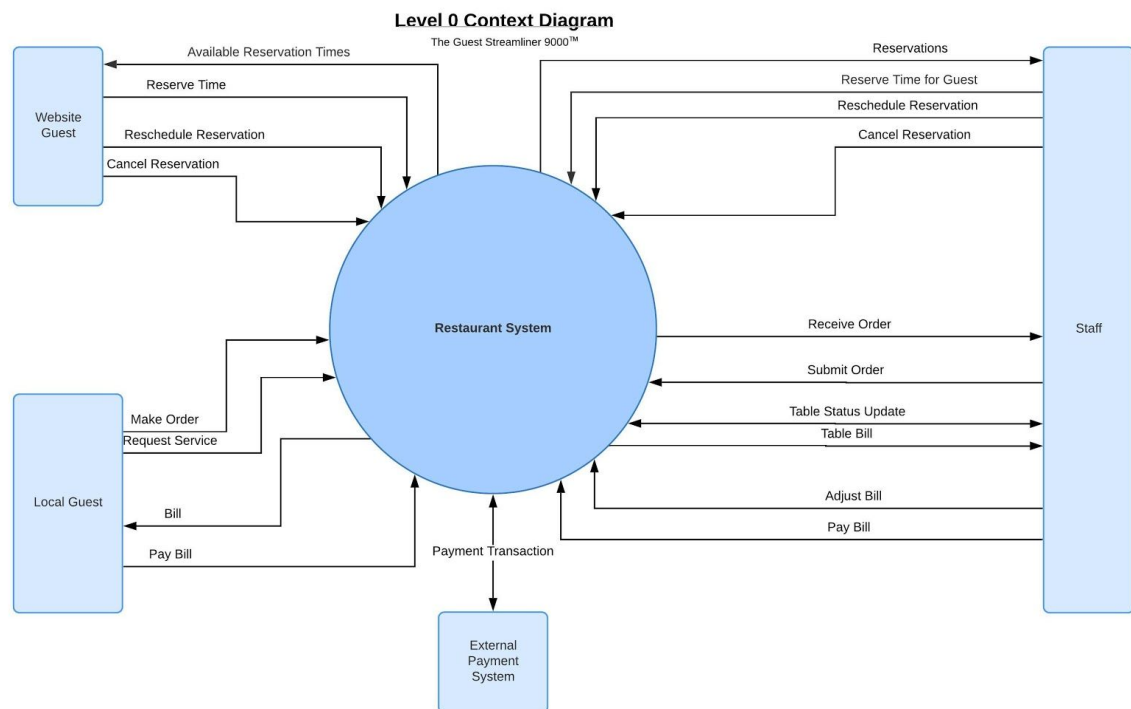
There will also be stations placed throughout the restaurant for wait staff to access. Bussers will be able to make table status changes at any of these stations

1.4. Major constraints

The software will have to integrate with an already existing website, because of the new information being collected, the database should be updated to hold the new information. Another Constraint is the external payment system which must be coupled to the software. Once released it will be Ivancic Italian's responsibility to maintain the software and to contact our cooperation if major problems arise.

2. Data and Component Design.

2.1. Internal Software Structure



Key

Circle: Process

Arrow: Flow of data

Rectangle: External entity

2.2. Component Rationale

There will be a component of the software that links the website to the database it will input user information for account creation and output reservation times and availability.

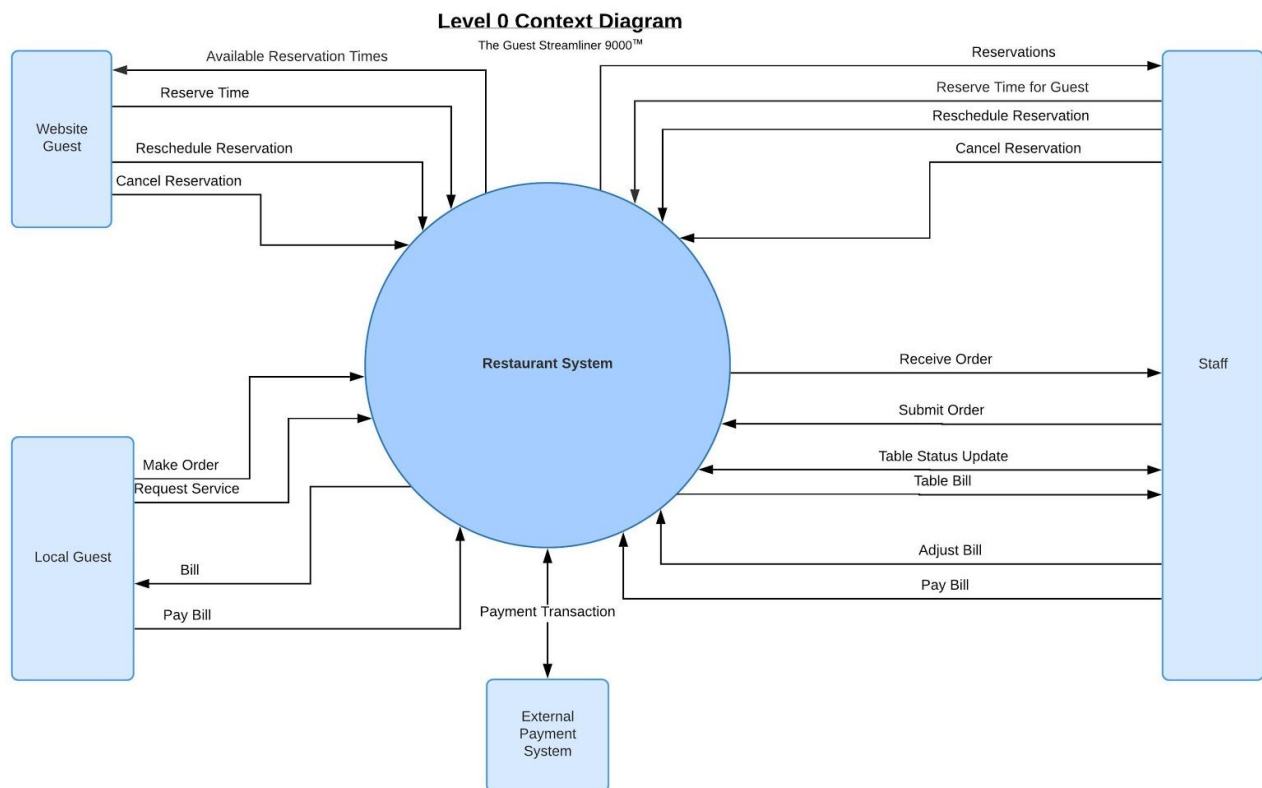
2.3. Database Description

The reservation system will have a database connected that will store the names, phone numbers, emails, times, and dates of a reservation. This will be linked to the website to insure that the correct amount of reservations are shown as to not overbook. This database will also be used by the host/hostess to check reservations as guests arrive as well as modify and add reservations through a front-end implementation for the guests who make reservations in person or over the phone.

3. Architecture Overview

3.1. Architectural Boundaries

3.1.1. Level 0 Context Diagram



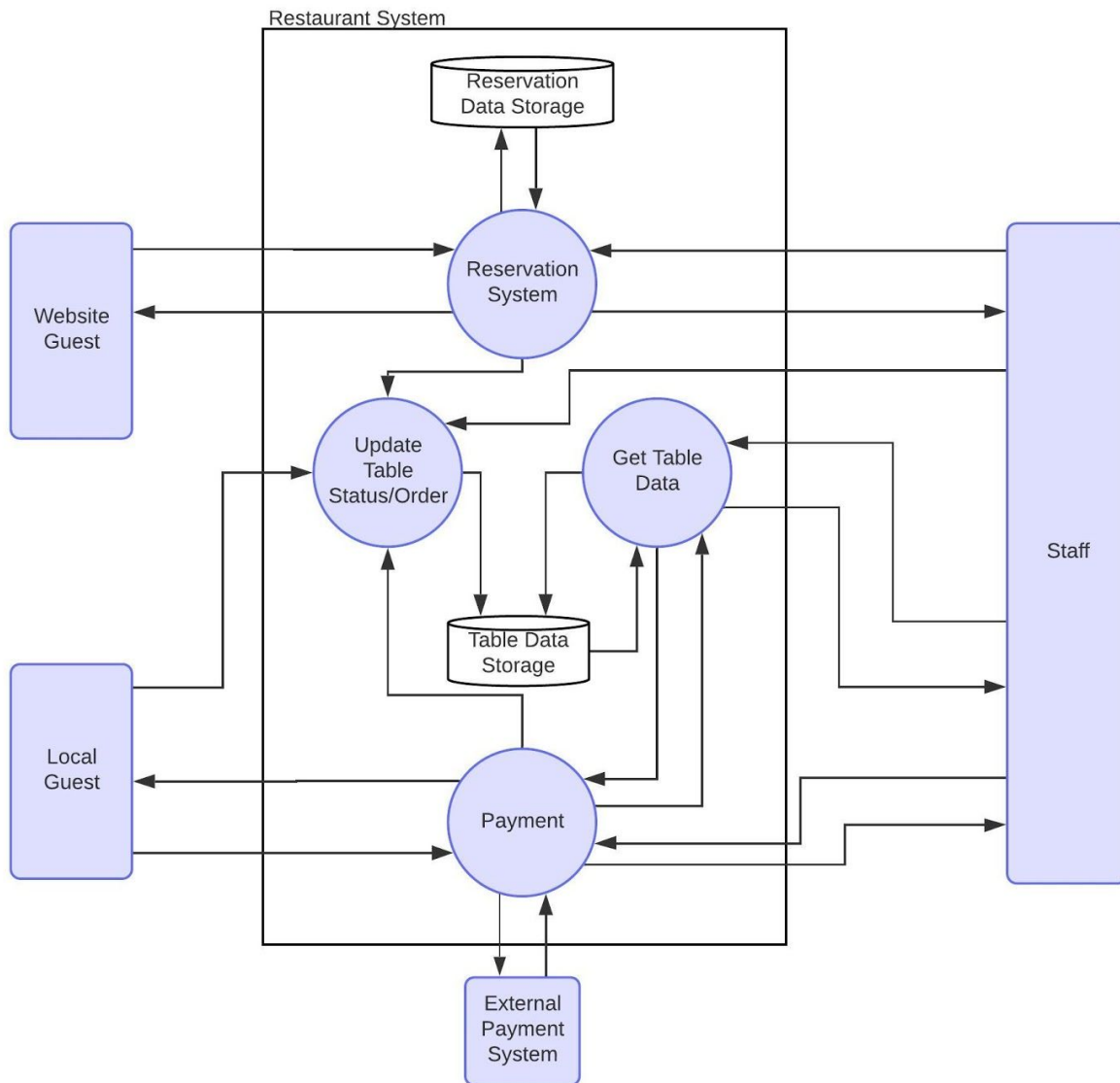
3.1.2. Rationale for the Level 0 Context

In the Level 0 Context Diagram there are four external entities. One being the Website Guest which is the website a guest will be interacting with in order to reserve a time. There is also the Local guest which is the POS systems at the tables that the guest can order, request services, or pay. The staff system which handles both reservations and the local services such as ordering, table status, and billing. Also for any non cash payments the external payment system is used.

3.1.3. Level 1 Context Diagram

Context Diagram Level 1

The Guest Streamliner 9000™

Key

Circle: Process

Arrow: Flow of data

Rectangle: External entity

Cylinder: Data Storage

3.1.4. Rationale for the Level 1 Context

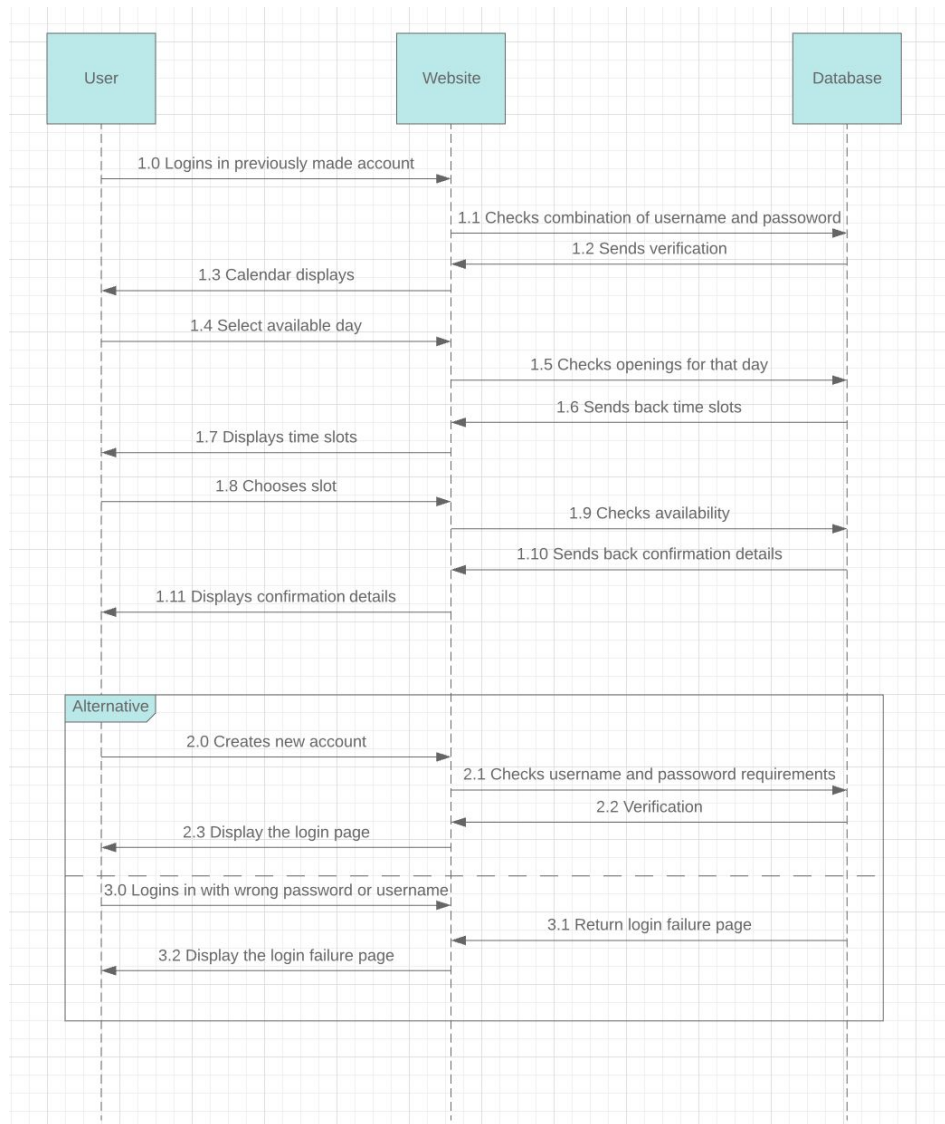
In the level 1 Context diagram the Reservation system will accept requests for retrieving reservation times and and accept changes to the existing reservation storage. The Update Table Status/Order system will accept changes to the status of tables and orders to assign to a table, it will update the tables data storage. Get Table Data will retrieve data related to the specified table and return it from Tables Data. The Payment system will get table data and display totals for tables also will accept payment. Which will be used with the External Payment System to confirm and charge the payment used.

3.2. Use Case specification achievement

3.2.1. Below is a list of the use cases and how the system will handle each.

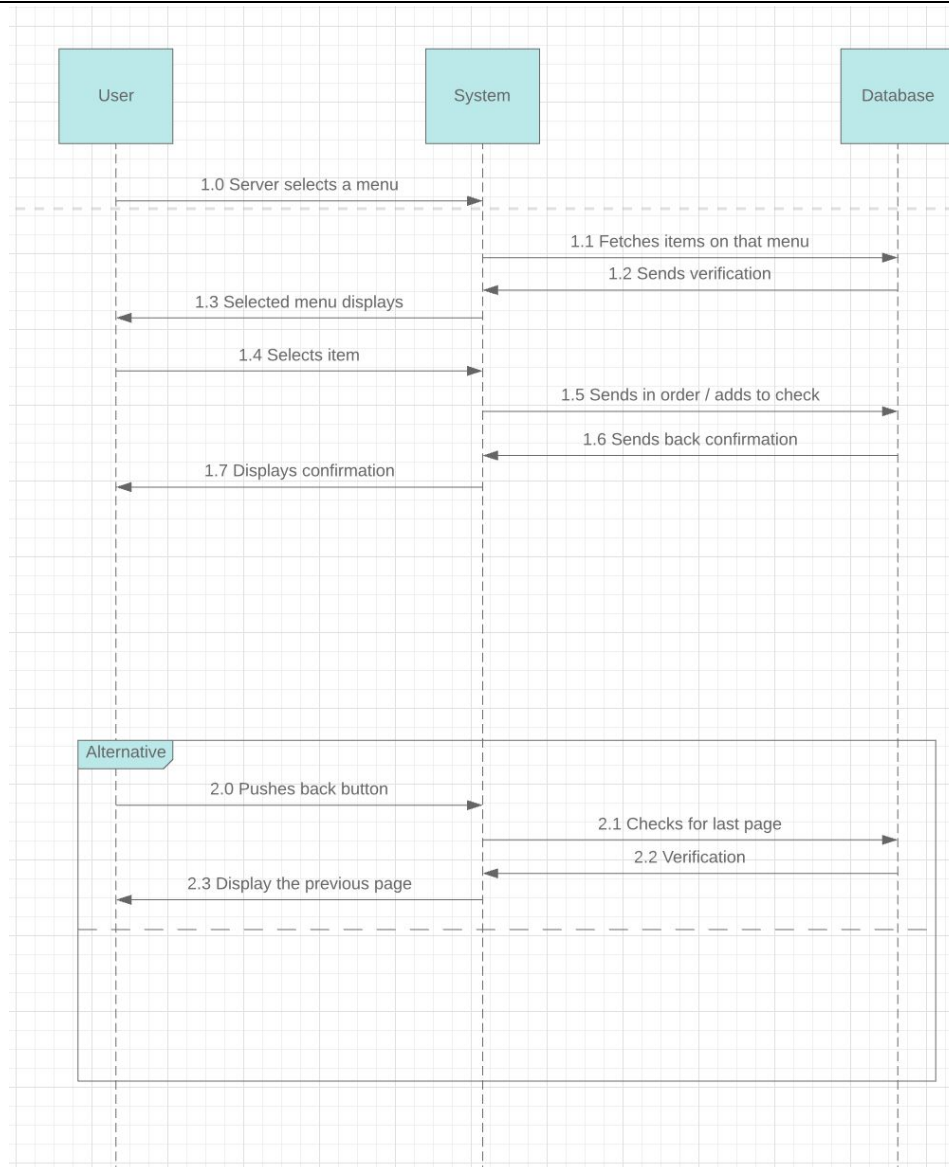
Use Case 1: Reservation Service

Name: Make a reservation
Description: Customer can visit website to make reservation
PreCondition: The user is on the website.
PostCondition: Customer has a table reserved.
Alternate Execution: There are not enough chairs available for selected time, offer times that are closest to the requested time. Restaurant closes in ≤ 30 minutes, offer them a reservation for the next day. They want a reservation sooner than an hour away, offer them to walk-in, or suggest a later time.
Assumptions: Customers and Wait staff make reservations the same way, should be through the same form.



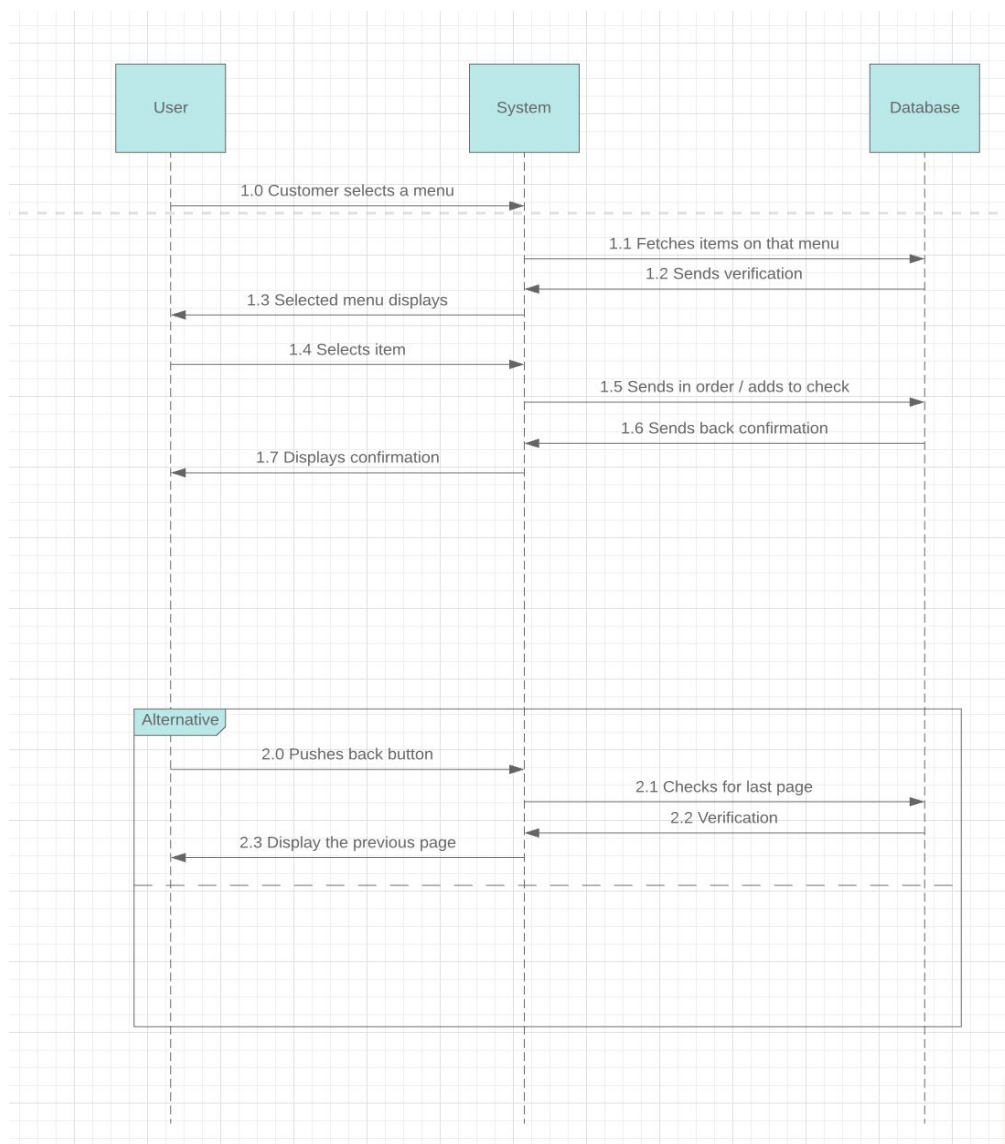
Use Case 2: Standard Ordering

Name: Standard Ordering
Description: Customer is able to place an order with the server.
PreCondition: Customer must be seated, and table status must be set to seated.
PostCondition: Food order is sent to the kitchen and table ticket updated.
Alternate Execution: TBD
Assumptions: Customer does not want to use the automated system.



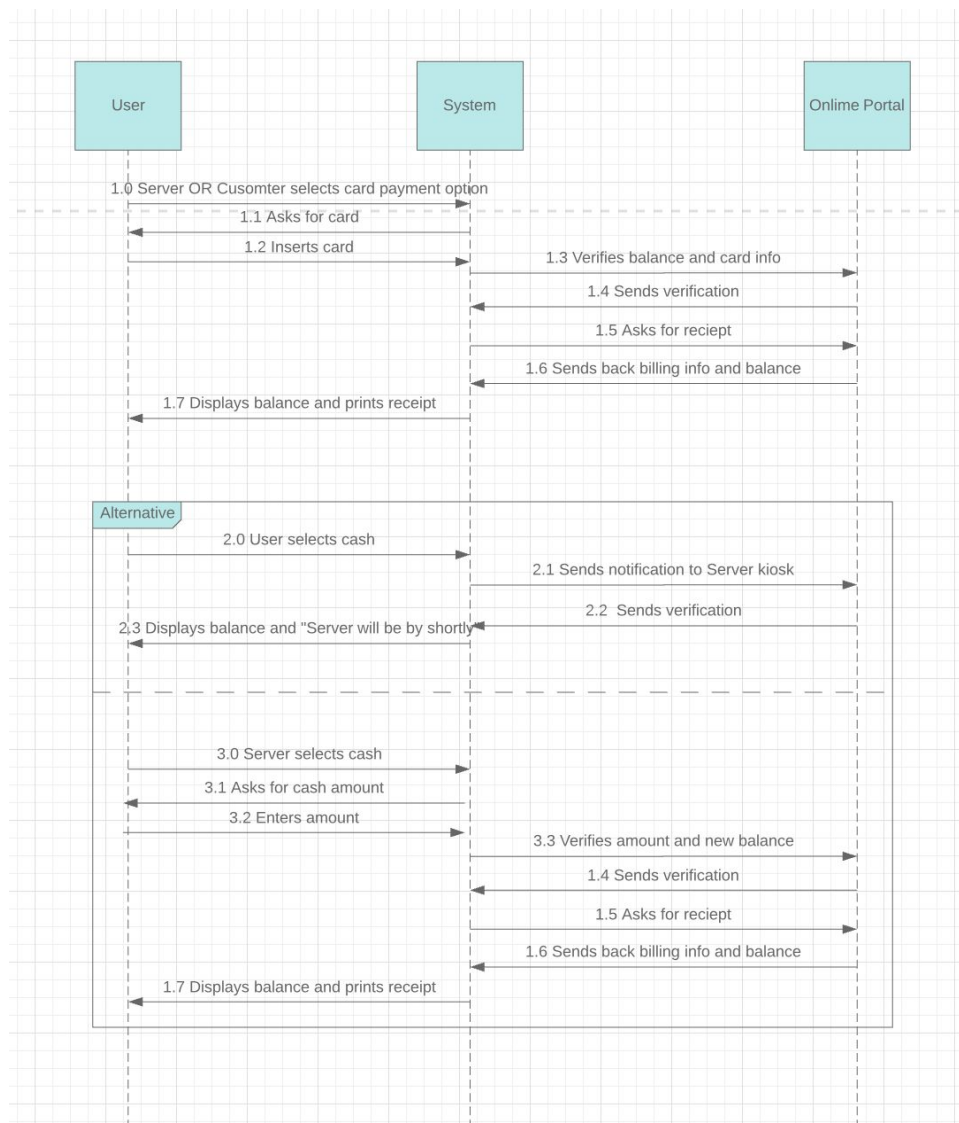
Use Case 3: Ordering from table

Name: Ordering from table
Description: Customer is able to make an order at their table.
PreCondition: Customer must be seated, and table status must be set to seated.
PostCondition: Drink order is sent to waitstaff and food order is sent to the kitchen.
Alternate Execution: Option for item is not on the list of item specific options. The Customer will have the option to page/call a server to make any special requests.
Assumptions: Customer is seated and table status is updated.



Use Case 4: Payment

Name: Payment
Description: Payment process at customer POS Terminals
PreCondition: The customer has a tab waiting to be paid
PostCondition: The customer's card has been charged and the table's status has been set to paid.
Alternate Execution: The charge fails
Assumptions: The customer's complete order is already in the system. The network is online.



3.3. Outside system requirements

The system needs to connect to a database for storing/retrieving reservations as well as the restaurant's website in order to have the ability to take online reservations and store them in the database.

4. Restrictions, Limitations, and Constraints

Software will have to be able to run on ipads meaning the design will have to be able to fit smaller screens. Software will be linked across multiple different types of devices so it must be compatible with all types.