Lennart Cockx & Guangming Luo

## GROUPT

Online Cash register

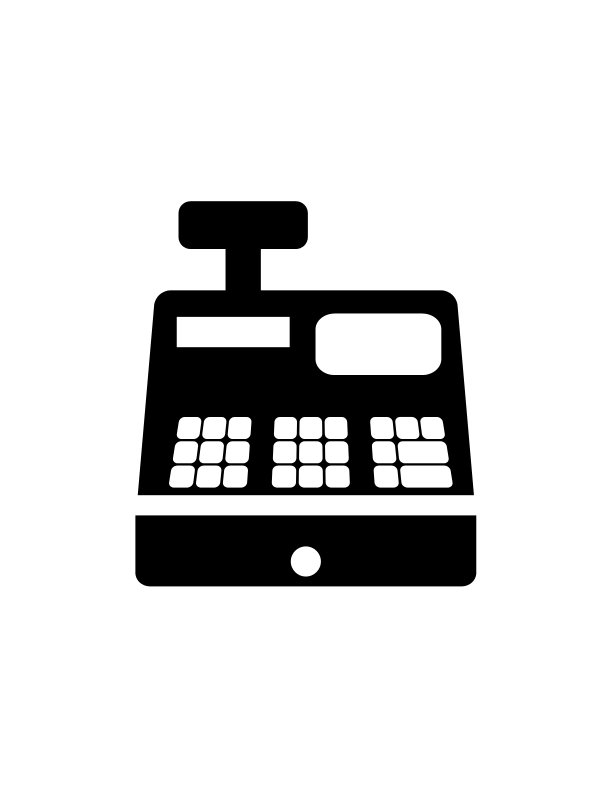


Table of Contents

[GROUPT   0](file:///C:\Users\lenna\Google%20Drive\Work%20&%20School\Master%20Thesis\Documentation\Master%20thesis%20v1.5.docx#_Toc462773820)

[Project objective 2](#_Toc462773821)

[Overview of the actors 2](#_Toc462773822)

[Project features 3](#_Toc462773823)

[Must-have 3](#_Toc462773824)

[Nice-to-have 3](#_Toc462773825)

[Arguable 3](#_Toc462773826)

[Design choices 4](#_Toc462773827)

[Application for the client 4](#_Toc462773828)

[Introduction 4](#_Toc462773829)

[Interesting remarks and challenges 5](#_Toc462773830)

[Application for the waiter and chef 6](#_Toc462773831)

[Introduction 6](#_Toc462773832)

[Multi-tenancy 7](#_Toc462773833)

[Interesting remarks and challenges 8](#_Toc462773834)

[Application design 9](#_Toc462773835)

[Application type choice 9](#_Toc462773836)

[Accessing the application 10](#_Toc462773837)

[Additional information 10](#_Toc462773838)

[Existing systems 11](#_Toc462773839)

[Application development and frameworks 11](#_Toc462773840)

[Bibliography 11](#_Toc462773841)

Brief

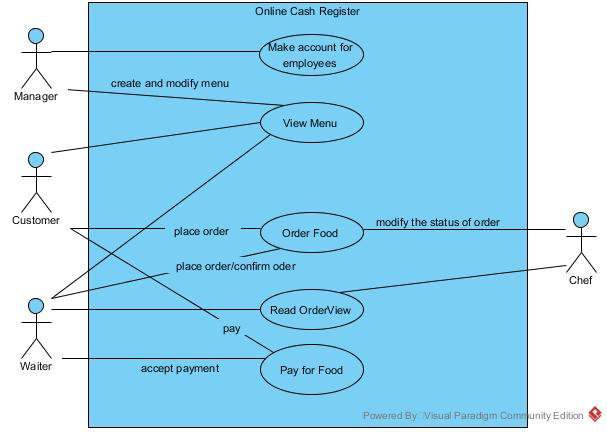
# Project objective

The goal of this project is to create a service that can be used at events and in restaurants to process orders made by customers, relay these orders to the waiters and chefs and finish off by handling the payment. This service should include an application that clients can use to view the menu and order items from it. Waiters also have their own application that can place and acknowledge orders. These orders will be send to a central system that manages all the required information.

# Overview of the actors

In this project we can distinguish several different actors that will each interact with the system in a different way:

* Customer
  + Can view the menu of the event/restaurant
  + Can place orders, either through the application or via a waiter
  + Can pay, either through the application or via a waiter
* Waiter
  + Can view the menu and order food and drinks for clients
  + Can accept orders and payment from clients
  + Can see an overview of current orders and their corresponding table number
* Chef
  + Can read order information from an overview and modify the status of an order
* Manager
  + Can create and modify the menu and pricing for the current menu of the restaurant/event and can give the necessary privileges to the waiters and chefs.



Project features

## Must-have

* The system can be used by multiple managers or event-organizers simultaneously.   
  **(Multi-tenancy)**
* A central application will handle orders and payment.   
  (**Central application)**
* Waiters can use an application to send the orders of customers to the kitchen and accept payment.   
  (**Employee interface + Payment system**)
* Customers can view the menu of the restaurant/event with their mobile device.   
  (**Customer interface**)
* An order overview is available for the waiters and chefs. It shows the completed orders and orders which have to be made.   
  (**Order overview**)
* Customers can add comments to certain items. E.g. no tomato  
  (**Special requests**)
* Customers can place their own orders with their mobile device.   
  (**Customer Mobile ordering**)
* Prevent customers from making ridiculous orders or misusing the system.  
  (**Misuse prevention**)

## Nice-to-have

* Event organizers can run the central application on their own local network.   
  (**Local network hosting**)
* After an order is finished, the completed order number is shown on a big screen.   
  (**Take-a-number/Now serving System**)
* The manager can find statistics on his restaurant and orders via a web app.   
  (**Order Statistics**)
* Customers can split the bill or combine orders   
  (**Bill splitting**)
* Provide an easy way for customers to use our system at an event/restaurant. Even if they don’t know it yet.   
  (**Easy access**)
* Customers can order additional food or drinks. This will be added to the total bill.  
  (**Multiple orders**)

# Design choices

## Application for the client

### Introduction

We want to make an application for customers of restaurants and/or events. The customer can use this application to get an overview of the available menu items and prices. If the customers have decided, they can order the desired dishes straight from the application. Finally, the customers can pay for their meal with this application.

We can differentiate a couple of different scenarios when the customer goes to a restaurant or event that supports this system. We assume that a web app is used. If a native or hybrid application is used, installing the app will be an additional step before using the system.

Customer has no device

Get a menu from a waiter

Order desired food from the waiter

Customer has a smartphone or tablet

Navigate to website and select a restaurant

Receive the menu of the chosen restaurant/event

Scan a AIDC1 code

Order desired food with the smartphone or tablet

*1Automatic identification and data capture. This can be a barcode, QR-code or something similar.*

After (or before) the meal is finished, customers should be able to pay directly from within the application or pay a waiter.

Order placed at waiter

Receive bill from waiter

Pay waiter in cash or by card

Order placed through the app

Receive an overview of order with a total price

Pay from within the app

### Interesting remarks and challenges

When using the application to place an order for **multiple people**, the customers should have the option to **order and pay separately or order together and split the cost.**For example: Three people want to place an order, two of which have access to the application.

* One person can place the order for the entire group.
* One person can order for himself and the person without the application. While the third person makes his own choice with his own device.
* The two persons with the application order for themselves, while the other person orders something from a waiter.

In all three cases the order will be combined into one order for that table. After the meal the bill can be paid for.

* One person decides to pay for the entire bill. This can be paid to a waiter or with his application.
* Each person pays for the part he ordered separately, with the price corresponding to what they ordered.
* Everyone decides to split up the bill equally, dividing the total cost over three people.

Typically, when people eat at a restaurant, they will place **multiple orders**. During the meal additional drinks might be ordered and at the end of the meal some people might want a dessert or coffee. We have to design the application so it can **accept multiple orders** and display the sum of these orders at the end of the meal.

## Application for the waiter and chef

### Introduction

To be able to interact with customers using this system, the waiters and chefs will need their own version of this application. They will need to get an overview of the current orders placed and they should be able to modify and confirm these orders.

Customer places order at waiter

Waiter adds the order to the correct table number

The order is added to the order overview

Customer orders with the app

Waiter is notified and can accept the order   
(or decline)

Customer requests bill

Waiter is notified and delivers the bill to the customer

Chef updates the overview when a dish is ready to serve

Customer immediately pays with the app

Payment is made and the orders for that table are reset

The order overview will have all the relevant information of the order: The different dishes and drinks ordered by the customer, the table from which the order was placed, the price and the status of the order. Additional information such as a time and date and an order number can be added for statistical purposes.

The order overview could look something like this:



Whenever an order is served, it will be removed from the overview and the next item will be moved up. When the bill is requested, all orders from the corresponding table will be added together. After payment the orders for that table are reset to allow the next customers to place their order.

### Multi-tenancy

Starting from our home page, our users, which are usually managers from restaurants or events, can first register here and login. After logging in, they can add one or multiple restaurants to his personal center. By entering, they can create a menu and modify it later if they want. Upon completing the menu, the system will generate the website for that restaurant or activities based on that specific menu.

Center Application Home page

Register

Login

Restaurant A

Restaurant B

Activity C

Add

Add

Add

Menu

Menu

Menu

Create Modify

Create Modify

Create Modify

Webpage A

Webpage C

Webpage B

Generate

Generate

Generate

### Interesting remarks and challenges

In some cases, an **order might be wrong** or a mistake with the number of dishes is made. The application should be flexible and allow a waiter to make **small modifications to the order** to correct a mistake.

Additionally, a customer might make a **special request** for his order. This can be due to allergies or plainly because of a certain preference. It might be useful to provide a bit of space to the order overview for extra remarks.

An order might contain a lot of different items. The full names of these items will take a lot of space in the order overview. Typically, **a shorthand notation** is used by waiters to more efficiently note down and read orders. A similar system could be used for the order overview.

Also, for showing the menu to the customer we will display a **vertical list** with the name of every menu item. This list will contain **detailed information** like the name of the dish, the ingredients, pricing and possibly even a picture. These menu items can be sorted into categories for easy navigation. On the other side, the waiter does not need all the detailed information. Instead a **compact and efficient** layout is preferred to be able to easily take an order. This could be a **grid layout** with simple icons or abbreviations for each dish.

 *Figure 1: Customer menu* *Figure 2: Waiter menu grid*

It is possible that certain people with bad intent might try to place orders for **ridiculous amounts** of food, **repeatedly place small orders** to fill the system or even place orders for **another table.** Some of these situations can be prevented with some smart programming like limiting the amount of orders one table can make each minute. Another way to prevent some of these issues is to have waiters confirm orders on their device before adding them to the kitchen. In day to day use, these scenarios will be rare, but it’s still important to think about the possible problems that might occur.

While a **user account** for the customer might be desirable to track certain statistics and prevent misuse of the system (like mentioned above), requiring an account might not be interesting, as this can make the process of using the application slow and unintuitive for the customer.

Technical

# Application design

## Application type choice

In our case the application will need to load menu and pricing information from an external source. This means that a **network connection will be required** regardless of which type of application we want to develop. Because of this, the required network access for both the web app and hybrid app will not be considered a disadvantage.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Web App | Hybrid App | Native App |
| Development costs | Low | Middle | High |
| Maintenance Updates | Simple | Simple | Complicate |
| User Experiences | Poor | Middle | Superior |
| Store or market | No | Yes | Yes |
| Installation | Not required | Require | Require |
| Cross-platform | Superior | Superior | Poor |

## Application Development frameworks

When developing a web application, we can follow several different routes during development:

* The web application can be built from scratch (E.g. with HTML and CSS)
* We can utilize a web framework (E.g. Spring MVC, JSF, Vaadin, google app engine)
* We can use a content management system (E.g Wordpress, Drupal)

From Scratch

Web Framework

Content management System

If we develop an application from scratch, we will have a lot of control over how we build the application and what functions that we put in. However, making an application this way will take considerably more time than if we make use of a web framework or CMS.

Using a content management system will significantly speed up the development, with some systems giving the option to generate a functioning website in only several clicks. These systems are typically much more limited in the possible applications that you can develop on it in comparison to working from scratch or using web frameworks.

A web framework gives us a multitude of pre-made tools and systems to work with, which makes it easier and faster to build a working system. It still retains a significant amount of freedom to develop as we like in comparison to CMS.

## Google now integration

Google now is a system available on Google now is a system available on android phones that enables the user to quickly find relevant information depending on their situation/request. Now is also provides context sensitive features that utilize other apps on your phone.

This system can provide an interesting way to interact with the online cash register we are developing. For example, a user could use a voice command to ask what the menu at this restaurant/event is. From locational data, the corresponding restaurant could be found and the menu could be displayed. Also, if a user searches for restaurants, google now can indicate whether the restaurant supports our system for ordering and payment.

Currently Google now integration is only available to select partners decided upon by Google. The system is still under development and not openly accessible yet. (Google, n.d.)

## Accessing the application

Instead of having the customer navigate to the website, choose the correct restaurant and type in their table number to start ordering food, we could provide a QR-code or something similar that the user can scan with their device. By scanning the code, the user will be directed to the subpage corresponding to the restaurant that they are in. It is also possible to launch a native or hybrid app with the correct parameters by using Intent. (Google, n.d.)

Additional information

* Make it possible to show the status of ongoing orders and perhaps show an estimate of how long before the current order is ready
* At an event there might be limited connectivity. In this case the event organizers could host a local network which customers can connect to and place their orders as usual.
* Currently there are several existing payment systems that we can utilize in the final system. A custom payment system is also possible, but requires significant investment in security and reliability.
* We can provide statistics of each dish sold. This can be used by the manager to check which dishes are the most popular and can provide an insight in the success of the restaurant/event.

Research

# Existing systems

* <https://www.gloriafood.com/>
  + This is a free online service that can help your restaurant take online orders and track these orders with fitting statistics.
* <https://www.chownow.com/>
  + Online ordering system, good payment system, notification for when the food is ready.
  + Get custom-built iPhone and Android ordering apps that are branded to your business. ChowNow mobile apps showcase your logo and food, creating the ultimate brand experience from mobile, to web, to in-store.
  + It’s not free:
    - Monthly Plan: $149/mon + $399 Setup fee
    - Annul Plan: $119/mon + $199 Setup fee
    - Two-Year Plan: $99/mon + $199 Setup fee
  + All plans include: unlimited orders; 1 WI-FI enable ChowNow tablet; branded iPhone & android ordering apps; unlimited ChowNow dashboard access; marketing services; website ordering; Facebook ordering; google ordering; menu upload; one-o-one training…

# Application development and frameworks

* Google now cards
  + Google now is a system available on android phones that enables the user to quickly find relevant information depending on their situation/request. Now is also provides context sensitive features that utilize other apps on your phone.
  + More information at <https://www.google.com/landing/now/integrations.html>
* Web app development engines
  + Multiple systems to accelerate the development of web applications exist.
  + Some available sources:
    - Google app engine <https://cloud.google.com/appengine/>
    - Google progressive/accelerated apps <https://developers.google.com/web/>
    - Other mobile app development platforms <http://www.techworld.com/picture-gallery/apps/18-best-mobile-application-development-platforms-tools-3375307/>

# Bibliography

Google. (n.d.). *Google Now Integration*. Retrieved from Google.com: https://www.google.com/landing/now/integrations.html

Google. (n.d.). *Intent - Android Developer*. Retrieved from https://developer.android.com/reference/android/content/Intent.html