Information Technology and Management Science
ISSN 2255-9094 (online)
2021, vol. 24, pp. 15–25
https://doi.org/10.7250/itms-2021-0003
https://itms-journals.rtu.lv

Automated System for Restaurant Services

Liva Deksne[1*], Arturs Kempelis[2], Toms Sniedzins[3], Armands Kozlovskis[4]
[1–4]*Riga Technical University, Riga, Latvia*

Abstract **– The study proposes a smart restaurant system and analyses its benefits to be able to determine system potential advantages in restaurants. Service time is one of the main criteria that can be improved to enhance the speed of the customer service as well as to increase the number of restaurant visitors. To develop the system, solutions found in scientific literature, software and their different architectures are analysed. It has been found out that it is possible to decrease the average restaurant service load time by 52.76 %. Two hypotheses have been proposed for further research in order to determine how a smart restaurant service system can increase chef's efficiency and how the use of different algorithms can decrease chef's workload during peak hours.**

Keywords **– Digital restaurant service system, digitization of restaurants, digitalized restaurant order processing, restaurant automation.**

I. Problem Statement

Implementation of process automation in business nowadays is increasing, as it enables specific actions to be performed more efficiently, faster or using less resources [1]–[3]. Digitization of business processes allows executing specific processes digitally using informational technologies, and creating control mechanisms, monitoring, increasing data and information transparency in order to analyse business processes more efficiently and detect weak points.

Customer service digitalization in restaurant business is a step towards a more efficient service. Currently, overall system automation that allows customers to place an order directly from the table sending information to chefs is not very common. There are several time intervals when a customer must wait while certain operations are executed in restaurants in case a waiter performs manual client service. Fig. 1 shows these several time intervals during the customer service process, when the customer has to wait. A significant part of this waiting time can be reduced by introducing process automation. An automated ordering system can not only increase the customer service time, but can also:

- facilitate waiters' job;
- allow chefs to review orders more conveniently as well as receive information about them instantly;
- give opportunity to customers to view the full menu or separate sections more easily;
- help plan the menu and purchase of the raw materials more efficiently;
- provide opportunity to create specialized offers based on most popular orders;
- allow receiving reviews from customers;
- increase productivity;
- reduce human errors.

In order to analyse benefits of the automated ordering system implemented in a specific restaurant, it is necessary to collect data and information as well as gather staff reviews in an extended period of time. However, if we look at the potential benefit – service time reduction, there is a possibility of analysing the data right after implementation of the system, because such digital automation has a direct impact on customer service time.

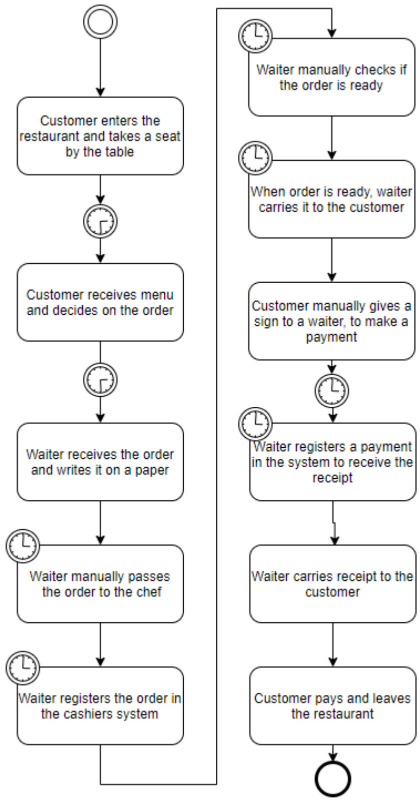

Fig. 1. Process for manual restaurant customer service system with reference to customer time consumption, waiting for a certain action to be performed.

In order to gain maximum benefits using the automated restaurant service system, it is necessary to integrate it with the payment system and to use direct information transfer to chefs.

[*] Corresponding author. E-mail: liva.deksne@rtu.lv

II. Literature Review

Analysing the scientific literature about restaurant automation possibilities and technologies that enable such process organisation, mostly process automation includes not only the option for a customer to place an order directly from a table using equipment with dedicated software, but also the creation of a general automated system for data transfer to a kitchen, a cash register system and general system management. Implementation of such an automated system allows achieving all the objectives and minimising the overall length of customer service processes (Fig. 1).

The main restaurant service benefits, using digitalized system for ordering meals [4]:

- improvement of customer service speed that results in an increase for both customer satisfaction and number of orders;
- reducing human errors;
- increase in service efficiency;
- decrease in restaurant labour costs.

Placing a touchscreen device on restaurant tables to give a customer ability to digitally browse restaurant's menu and place the order, combined with kitchen and management maintenance systems, is an effective way to automate the work of restaurants, make it more functional, more flexible and more cost-effective [5]. It is necessary to use appropriate equipment that has a high level of durability, is less sensitive to dirt and is scratch resistant. The screen has to be transparent enough to visualize the menu, as well as it needs good response time and overall stability [4].

There are several roles defined to which the digitalized system has adapted in many studies, and they differ from each other with enabled specific operations of restaurant processes. Many studies discuss the automated restaurant system and possible operations from the user's perspective.

1) Customer – uses a touchscreen device placed on the restaurant table to view full menu with images. Using this device, customer places an order. Information about the order is transferred to the user type Cashier and user type Kitchen. Customer has an option to leave a review about their ordering experience, the food and drinks [4], [5]. Additionally, there is a functionality to call a waiter with a single button click in the application [6].
2) Cashier – receives information about placed orders, reviews purchases and their prices, summarises the total price after the customer has finished the order, and receives money from the customer [4], [5].
3) Kitchen – receives information about the order and marks the order when it is completed [4]–[7].
4) Manager – owns full system control and ability to review all operations. Also manager's centralization is used – manager receives information about the ready order from the kitchen and gives a signal to the waiter. Manager can use the application to create the menu and, if necessary, make changes [4]. Storing these data in a cloud environment is a great option that allows working remotely. Remote editing of the information is a convenient mode and it also allows immediately displaying all changes on a restaurant's website [6].

Defining roles based on user types makes it possible to divide use of the system functions according to business needs and ongoing processes. However, in the approach where the role of the Manager is centralized and the user type Manager acts as a middleman between the user type Kitchen and waiters, a waiter does not have a separate user type. It can complicate the overall ordering process and create delays in the delivery of ready meals to customers, because the centralized user type Manager has to monitor the status of ready meals and manually transfer information to waiters.

In the research publications reviewed by the authors, the system covers the entire restaurant ordering process from customer and waiter orders to kitchen and cashier applications [4]–[6], [8]–[12]. In order for a customer to place an order at the table, it is most often recommended to use Android operating system applications that run on tablets [4], [6], [10]–[12]. Android operating system is the most popular Linux based operating system, which is primarily developed for mobile and tablet devices. From the developer's point of view, it is cheaper than iOS. System specifics differ in different literature sources. For example, one of the options is to use two Android operating system applications – customer application and restaurant application, where the customer application contains information about the exact location of the table to trace it when the customer is placing the order, but the restaurant application is intended for restaurant employees and contains three types of users (Customer, Cashier, Kitchen) with the already mentioned functions [4].

A different approach uses four types of applications – customer and waiter applications, restaurant web application and management application [6]. In this case, the waiter application is defined separately and contains several functions, for example, additional information about the order, grouping of the payments, as well as order confirmation before it is transferred to the kitchen. The waiter application is also used to receive information about ready orders from the kitchen, unlike the approach described above, in which this function is intended only for the manager. A similar system is also described in papers [11], [12].

The choice of system application user types depends on the specific restaurant business and its customer service. In restaurants where the cashier's and ordering duties are performed only by waiters, it would be more convenient to use a system that provides more functions in the waiter application, though in cases, where the service process is organised with the involvement of the manager, a system can be used in which the manager also acts as a middleman between the kitchen and waiters.

Based on one of the research approaches in restaurant automation, GSM layer is used for data transmission. The authors conclude that it has lower costs, improved quality performance, large transmission range (1000+ meters) and high level of reliability. If a GSM module is used, several modules can be used in the same range, unlike RF modules that do not work if there are more than three modules in one area. GSM bandwidth is 900 MHz, and the communication format does not require a large channel bandwidth. As a result, the number of channels increases significantly and, thus, the efficiency. It is possible to add additional channels using CDMA technology [4].

The LCD screen is most often used for better review of orders in the kitchen area. To connect the Android app used by the chef with the LCD screen, it is possible to use Arduino tool, which uses Bluetooth connection to mirror application information on LCD monitor. Arduino is an open-source physical computing platform, which is based on a simple microcontroller board, and is used as an environment for adding software. Likewise, there is a logical queuing system that allows chefs to see orders in specific order. Arduino uses wireless Bluetooth connection to transmit order information to the LCD monitor [4].

RFID-based technology integrated with WLAN and database technology is also used for data transfer in the automated system of restaurant processes. It is used to develop an e-restaurant system for customer-oriented services. In this case, RFID-tagged customer cards can be used that allow waiters to immediately identify customers and then provide customized services [6].

The architecture of restaurant customer service systems usually is created using a centralized database. Such a system contains a database server with restaurant menu and all of the information regarding ordering logic. Devices can be connected using secure and private WLAN network [5]–[7], [9]–[12]. It is also possible to use real-time data reading when the customer application is connected to a database server and downloads restaurant menu in real time [6]. Another approach to send customer order data to the server is to use the already described wireless GSM connection [4].

In addition to the existing functionality, it is possible to add video cameras in the kitchen in order to let customer to see how their order is being prepared in real time [9].

Customers can retrieve information directly from the database using touchscreen devices in the comprehensive restaurant service system. The menu list can be designed to call PHP functions to extract information from the database and save it to an XML file. The order information selected by the customers can then be stored as data objects in the database [5].

Zend Framework is one of the frameworks that can be used to develop a comprehensive automated restaurant service system. It provides wide component libraries for the functional web management system development. The Zend framework provides many modules and classes developed over the years by the open source community and it is easily extensible [5]. Using the framework, it is possible to develop the required controllers with individual methods – login authentication and user type classification, menu updating, display daily messages, database retrieval operations and order generation according to received requests, as well as order accounting and queuing to display in the kitchen (Fig. 2) [5].

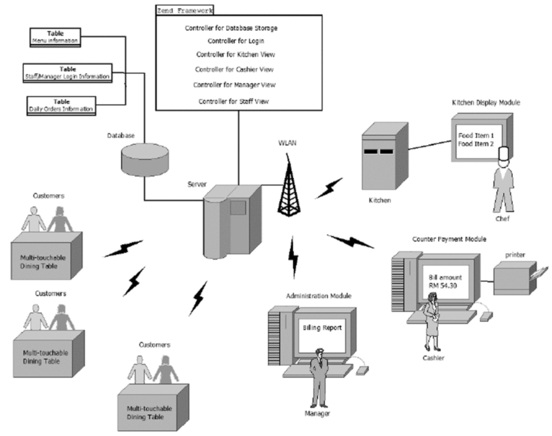

Fig. 2. Automated restaurant service process system architecture [5].

Extending the architecture of the considered automated system with Internet of Things (IoT devices) or in other words – sensors, it is possible to achieve additional system functionality in the case of long queues, where customers are waiting for the available table. Use of the sensors can significantly improve customers’ experience, promote the restaurant brand value, increase the overall level of service quality, as well as improve effectiveness and decrease the expenses. Sensors can serve as a basis for a smart queuing and reservation system. An existing WeChat application with sub-software can be used, which can also serve as a website where, for example, with the help of a QR code developed by a restaurant, a customer can occupy the queue and come to the restaurant only when the table is available. Besides, it is possible to use a sub-software of WeChat application in order to develop the restaurant menu environment, which can be used by customer through touchscreen tablets installed in the restaurant [8].

Extending automation to restaurant warehouse processes make it possible to use RFID technology in order to label goods in the restaurant warehouse in order to make the management of raw materials more efficient and ease the purchase process by digitalizing information about expiry dates of goods [8]. Robots equipped with various necessary sensors, as well as RFID readers can be used for kitchen warehouse service management, as well as for waiter functions [8]. Robotic systems of IoT technologies are used to replace the traditional workforce, to make various activities more efficient, to reduce the mistakes made by employees, which are often caused by long working hours of waiters.

Using robots to accept and deliver orders to customers is another possible approach to automate the overall restaurant service system. Specialized robots with laser scanner and ultrasound sensors can be developed in order to recognise their location in the restaurant. When the restaurant is being scanned by sensors, it is possible to give a voice command to move the robot to previously defined place in a restaurant. Likewise, it is possible to teach the robot different types of voice commands. Voice commands can also be used to move the robot or interact with another robot software. By adding a serving tray to a robot, it can also be used to deliver orders to a customer [6]. In this case, a robot can partially replace the waiter's job responsibilities while not losing the usual restaurant concept in accepting customer orders [6]. However, the use of such robots is not common worldwide and they do require full digitization of the customer service system.

Analysing a comprehensive smart restaurant system, it is possible to use [8]:

- IoT-based systems, including smart numbering system and automatic ordering system;
- intelligent kitchen using RFID automatic identification technology, automatic warehouse manipulator and automatic food delivery robot.

It is important not only to digitize the customer service process, but also to implement automatic inventory and control of raw materials in restaurants with high turnover and demand. A specialized order management system for this function is able to determine the necessary raw materials for the dishes ordered by customers, thus facilitating the work not only for the cook, but also for the manager [13].

The most widely used device for processing customer orders in restaurants is the personal digital assistant (PDA). It is used to automate part of the overall service process, as this device is usually designed for waiters who accept orders from customers. Afterwards data can be transferred to the cash register system and the kitchen [7]. Such systems allow speeding up service time, increasing service efficiency and the accuracy level of orders. PDA system typically uses a wireless network for data transmission [7]. However, it does not offer a separate customer and overall management applications, which means, all of the processes cannot be automated. The potential reduction in customer waiting times will be less than using a comprehensive digital restaurant system. The strengths of the PDA system are the reduction of the waiter's service time, the reduction of errors and others that focus directly on the waiter's duties. The weakness is its inability to support and to be integrated into other business processes [7].

Analysing different existing automation approaches of the restaurant service processes, it is possible to conclude that it is important to adapt the logic of the system to the business operations of a specific restaurant. Most approaches are based on the implementation of an integrated system that not only automates the ordering process, but also connects it to the chef's work in the kitchen and the cashier's system. A comprehensive system for automating restaurant orders makes it easy to add additional features such as feedback collection, highlighting most popular orders and more. This extra functionality provides additional benefits to the business and increases the level of customer loyalty.

The architecture of the considered system is most often created using a centralized server with a database, through which one or more applications can exchange data. Likewise, there are some existing application solutions on the market that can be used as a basis to create the application for a specific business. The technological techniques for creating a basic ordering system in the analysed scientific papers are very similar, except for those techniques in which robots are involved in the process.

Technologies and development process techniques for restaurant service process automation in research papers are summarised in Table I.

TABLE I

TECHNOLOGY AND AUTOMATION PROCESS APPROACHES DESCRIBED IN THE CONSIDERED REFERENCES

No	Technology or technique used	Reference
1.	Automation covers customer ordering process, chef's systems and payment system digitalization	[4], [6], [5], [8], [9]–[12]
2.	GSM usage	[4]
3.	WLAN usage	[5]–[7], [9]–[12]
4.	Arduino tool for data transmission to the LCD screen using Bluetooth	[4]
5.	Android based application	[4], [6], [10]–[12]
6.	Centralized database	[4], [6], [5], [10]–[12]
7.	Integration of two applications – customer and restaurant	[4]
8.	Integration of three applications – customer, manager and kitchen	[11]
9.	Use of robots in process automation	[6], [8], [9], [13]
10.	Use of WeChat application sub-applications for automated queue management	[8], [13]
11.	Use of RFID for automated raw material warehouse management	[8], [13]
12.	RFID customer card usage	[6]
13.	LCD usage in chef's space	[4]–[7], [10]
14.	Development and implementation of a system prototype in business	[6]
15.	Use of a personal digital assistant to automate the work processes of waiters	[7]
16.	Use of Zend framework	[5]
17.	Use of PHP logic	[5]
18.	Use of MySQL database	[5], [10]
19.	Use of Adobe Flash AS3	[5]
20.	XML usage	[5]
21.	Touch-sensitive device available to customers to place orders	[4], [6], [5], [8]–[13]
22.	Using Visual Basic 2008 to develop a software user interface	[9]

III. METHOD

In order to assess how an automated restaurant service system and application can reduce the service time and give additional benefits to the business, the authors develop an application to provide basic functionality, which can serve as a basis for an automated restaurant service system, as well as can be supplemented with various additional functions required for specific business needs, such as customer reviews, highlights of the most popular dishes, different data analytics, etc. Similarly to the systems described in research papers, the authors include customer and kitchen area applications, so that the customer order automatically appears in the kitchen. The authors do not integrate waiter and manager applications because the system was developed with the aim to evaluate the time efficiency of the food ordering process. The system can be upgraded by adding additional features of the waiter and manager that can increase its usability and provide other benefits.

The following business functions are distinguished in the application developed by the authors:

1) Customer registration.
2) Table reservation and order placement.
3) Automated order transfers to the chef's application.
4) Opportunity for the cook to view incoming food orders, accept them, reject them and announce when the specific order is ready to be served.
5) Customer receipt payment option.

The scheme of comprehensive business functions is shown in Fig. 3. It contains representation of two main views that include customer and kitchen areas. The system ensures data exchange between these two parts of the restaurant business.

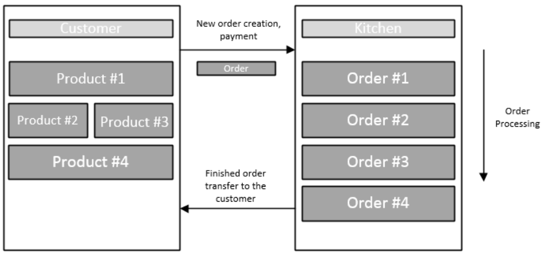

Fig. 3. The business parts covered by the created automated restaurant system.

To ensure the mentioned business functions that serve as a basis for the automation restaurant system, the following technology scheme is created (Fig. 4).

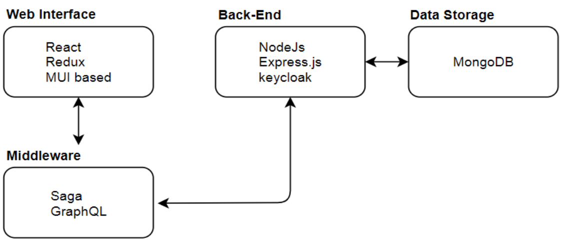

Fig. 4. Technological scheme of the created automated restaurant system.

In order to create the automated restaurant system, four subsystems have been developed – web interface, middleware, back-end and data storage (Fig. 4). This technological scheme also shows data request and exchange path in this application. The infrastructure provides a computing resource with 2 GB RAM, 50 GB HDD and Ubuntu 18.04.2 operating system.

Data storage

The database is created using the MongoDB provider's service, which is based on the containerization method to provide databases. MongoDB is a global cloud database service for modern applications. It is a database that stores data in JSON-type documents, which in this particular application makes it easier to operate with database requests, while allowing the development of flexible and dynamic schemas. MongoDB offers a wide range of query options that allow filtering database objects by any field regardless of the complexity of the overall structure, as well as creating queries in JSON format that facilitates their creation. The database stores restaurant menus, user and order information. The back-end system will provide data entry and deletion of customer orders.

Back-end

A JSON type API is used to transfer the received data to the database, as well as to retrieve the necessary information from the database. It is created using the Express.js web service framework for the Node.js cross-platform that is a real-time JavaScript environment designed to execute JavaScript code on the server side. Using Node.js allows creating dynamic web page content by allowing the server and client part of a web application to be written in the same programming language.

Data exchange between the back-end (server side) and the client interface side uses the JSON data exchange format and the HTTP protocol. They are provided by these endpoints:

1. Request (retrieve menu data):

```
GET /users/meals?type=dessert,meat
```

Response:

```
[
      {
    "_id": 1,
    "name":  "Veal  chop  in  breadcrumbs  served  with
    butter fried potatoes and tomato salad",
    "price": 19.95,
    "type": "meat"
  },
  {
    "_id": 18,
    "name": "Tiramisu",
    "price": 6,
    "type": "dessert"
  },
  ...
]
```

2. Request (add new user data):

```
POST /users
```

Request body:

```
{
    "name": "Students Artūrs",
    "email": arturs@students.lv,
    "password": "12345"
}
```

Response:

```
{
    "name": "Students Artūrs",
    "email": "arturs@students.lv",
    "createdAt": " Wed Apr 08 2020 14:03:17 GMT+0300
    (Eastern European Summer Time)",
    "id": "5e8daf7589cf2d78741d45d1"
}
```

3. Request (create new order):

```
POST /users/5e8daf7589cf2d78741d45d1/order
```

Request body:

```
{
  "meals": [
      {
    "_id": 1,
```

```
    "name": "Veal chop in breadcrumbs served with
    butter fried potatoes and tomato salad",
    "price": 19.95,
    "type": "meat"
  },
  "tableId":"5"
]
```

Response:

```
{
  "userId": "5e8daf7589cf2d78741d45d1",
  "tableId": "5",
  "status": "NEW",
  "createdAt": "Wed Apr 08 2020 14:06:28 GMT+0300
  (Eastern European Summer Time)",
  "meals": [
      {
    "_id": 1,
    "name": "Veal chop in breadcrumbs served with
    butter fried potatoes and tomato salad",
    "price": 19.95,
    "type": "meat"
  },
  "id": "5e8db03589cf2d78741d45d2",
  "action": "CREATED"
}
```

4. Request (View all orders):

```
GET /kitchen/orders
```

Response:

```
[
      {
    "id": "5e8db03589cf2d78741d45d2",
    "userId": "5e8daf7589cf2d78741d45d1",
    "tableId": "5",
    "status": "INPROGRESS",
    "createdAt": "Wed Apr 08 2020 14:06:28 GMT+0300
    (Eastern European Summer Time)",
    "meals": [
            {
      "_id": 1,
      "name": "Veal chop in breadcrumbs served with
      butter fried potatoes and tomato salad",
      "price": 19.95,
              "type": "meat"
    }
    ]
  },
  ...
]
```

5. Request (Delete order):

```
DELETE /kitchen/order/5e8db03589cf2d78741d45d2
```

Response:

```
{
  "deletedOrder": "5e8db03589cf2d78741d45d2"
}
```

For all of the endpoints, authorization with username and password has been created, which is based on a standard *Authorization Basics*.

Middleware

Middleware is designed to improve the performance of data exchange between the web interface and the back-end system. It helps reduce the number of data calls to the back-end system, which also allows for faster processing of requests, as well as reduces server load. Within the limited resources of the project, the planned middleware was not created. It is possible to add it, as well as to evaluate its impact on the processing of system requests, comparing the processing capacity and time with the system created by the authors.

Redux-saga is a library that is used to facilitate application management, run it more efficiently and reduce errors. Saga is a separate element in the middleware that is mainly used for asynchronous operations, like data retrieval, access to browser cache. Redux-saga is a Redux middleware that allows easily managing this Saga element using the main application.

The middleware solution includes GraphQL, which is an API query language that provides a complete and understandable API description. GraphQL allows running applications faster and in a more stable manner because it controls the data obtained rather than the server. Furthermore, GraphQL API in a single request retrieves all of the data that are needed for the application. This functionality gives advantage for customers because it increases the speed of application.

GraphQL queries access not only the properties of a single resource, but also smoothly follow the references between them. Although in a typical REST API data are downloaded from multiple URLs, GraphQL API does it in a single request, retrieving all necessary data for the application. Applications that use GraphQL can be fast even with a slow mobile network connection. Likewise, GraphQL allows only one endpoint to be used to access the data.

Web Interface

The React JavaScript library is used to develop the web interface. It allows creating an interactive user interface. React uses encapsulated components that are easy to use for complex user interfaces. Component logic is written in JavaScript, which allows data to be exchanged within the application while maintaining the non-DOM state.

The developed web interface is MUI (Material User Interface) based interface, which means React components are faster and easier to use to develop a web user interface. It offers a wide range of options in modern design and formatting.

The user interface of the created web application is shown in the following images (Figs. 5–8).

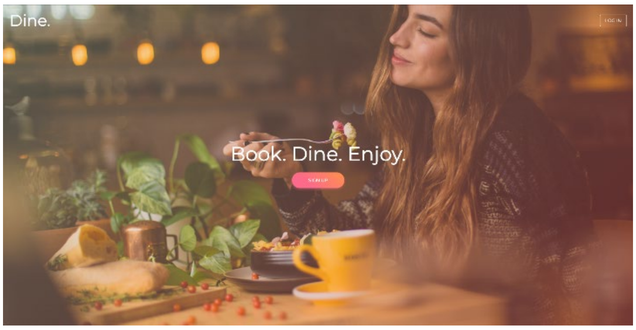

Fig. 5. Sign-up view of the developed automated restaurant system.

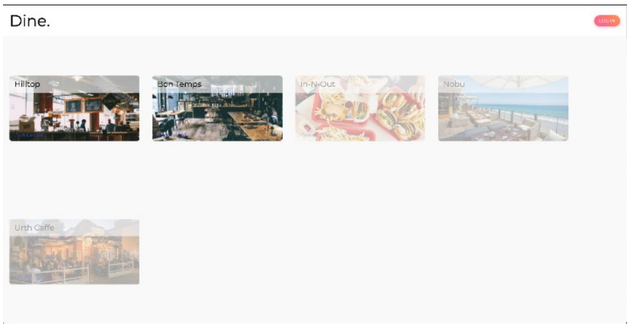

Fig. 6. The main menu view of the developed automated restaurant system.

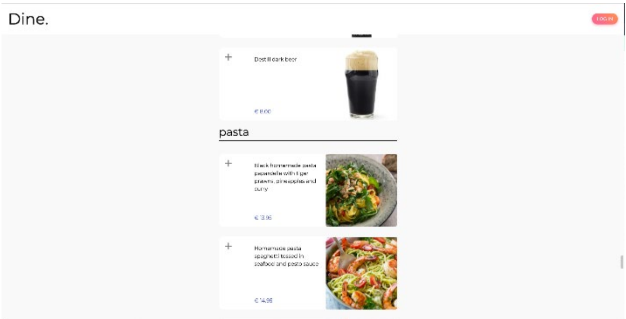

Fig. 7. Menu view of the developed automated restaurant system.

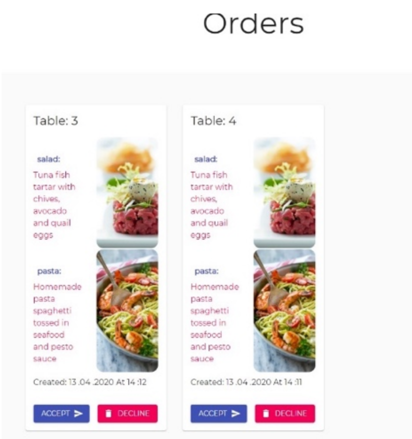

Fig. 8. Kitchen application of the developed automated restaurant system.

IV. Evaluation of Effectiveness

In order to evaluate effectiveness of the developed application, it is necessary to perform time counting experiments in a restaurant where process automation is not implemented. A restaurant that has the potential to speed up service time by implementing the automated restaurant system, must be with medium and high customer load per day, because the automated system is able to provide the benefit of service time savings, if a particular restaurant has a number of simultaneous customers, where it is not possible for waiters to serve at the same time.

For the purpose of the experiment, time was recorded in a restaurant with a medium (Table II) and low load (Table III). For more accurate time measurements, the experiments were repeated five times. Time that is spent for one order is divided into five stages according to the potential time savings using the automated restaurant service system. The time depends on the workload of the waiters and chefs, so it may change as their workload increases or decreases, as well as other factors such as the total number of waiters, chefs available at a given time, etc. can affect the total time.

TABLE II

Results of the Experiment for Manual Service in a Restaurant with a Medium Load

Time intervals	Experiments (min)					
	No. 1	No. 2	No. 3	No. 4	No. 5	AVG
Customer waits for the menu	3	5	2	6	3	3.8
Customer waits for the waiter to place the order	10	7	9	11	7	8.8
Customer waits for the order	18	20	24	17	25	20.8
Customer waits for the waiter to notify about the payment	11	9	14	8	12	10.8
Customer waits to get change	6	5	7	6	9	6.6
Total:	48	46	56	48	56	50.8

The longest waiting time in case of medium load is observed in the phase affected by the cooking of the ordered food, but the longest time affected by the waiter's activity is observed in the phase where the customer waits to inform the waiter that they want to pay (Table II).

Repeated experiments were performed to assess how each specific time interval changed in case of a low load (Table III).

TABLE III

Results of the Experiment for Manual Service in a Restaurant with a Low Load

Time intervals	Experiments (min)					
	No. 1	No. 2	No. 3	No. 4	No. 5	AVG
Customer waits for the menu	2	1	2	1	1	1.4
Customer waits for the waiter to place the order	4	4	3	4	5	4.0
Customer waits for the order	14	13	12	11	10	12.00
Customer waits for the waiter to notify about the payment	4	5	3	5	4	4.2
Customer waits to get change	4	3	3	4	5	3.8
Total:	28	26	23	25	25	25.4

In both low-load and medium-load restaurant, on average, the longest time is required during the cooking phase (Table III). The average total time period for all phases together in a low-load restaurant is 50 % shorter than in a medium-load restaurant. The most significant differences, comparing both of the experiments, occur in the phase "Customer waits for the menu", where the time consumption is 63.16 % less in a low-load restaurant than in a medium one. The smallest change (42.31 %) is observed in the phase "Customer waits for the order", comparing both low- and medium-load restaurants. Such results can be explained by the fact that bringing the menu is performed almost immediately after the customer has chosen a table in a low load restaurant, because the waiter is not busy with other customers that can significantly prolong this time. Furthermore, a reduction in cooking time, comparing medium and high-load restaurants, is only possible by the time interval in which the customer waits in the line while the chef processes the order.

By implementing an automated system in the restaurant, it is possible to significantly reduce the time of handing over the customer's order information to the chef, as well as other time intervals related to the work of the waiter (Table IV).

TABLE IV

AUTOMATED SERVICE SYSTEM EXPERIMENT RESULTS IN MEDIUM- AND LOW-LOAD RESTAURANTS

Time intervals	Experiments (min)					
	No. 1	No. 2	No. 3	No. 4	No. 5	AVG
Customer signs in the system	2	1	1	2	1	1.4
Customer finalizes the order	1	1	0.5	1	0.5	0.8
Customer waits for the order (medium/low load)	18/14	20/13	24/12	17/11	25/10	20.8/12.00
Customer pays in the payment system	1	1	1	1	1	1.0
Customer waits to get change	0	0	0	0	0	0.0
Total:	22	23	26,5	21	27.5	24/15.2

By using an automated system in a medium-load restaurant and assuming that the order cooking time is equal to this time in the case of a manual restaurant, it is possible to reduce the total waiting time by 52.76 %, but in the case of low load – by 40.16 %. Customer does not have to wait for the menu, although a minimal time was spent while waiting for authorization, when using the automated system. The authorization time can increase for the people who do not usually use digital systems.

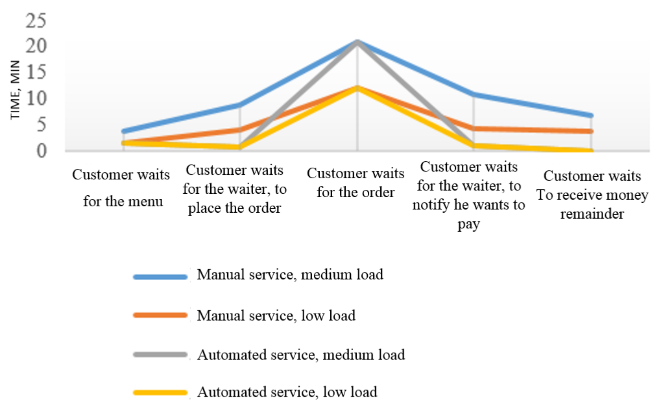

Fig. 9. Service time of manual and automated restaurant.

In the case of automated service, the cooking phase is the critical phase during the overall waiting period (Fig. 9), as the total service time depends on it the most. In the first comparison of the system effectiveness, it was assumed that this time did not change by using the automated system, but, since the data transfer of order information to the chef's system in the automated service was faster, then it was expected that at a certain load the number of orders in the queue in the chef's system would increase significantly. Their processing speed depends on the number of chefs in a particular restaurant. The authors, in consultation with restaurant experts having many years of experience in organising and maintaining a restaurant business, have formulated two hypotheses:

1) In the case of an automated restaurant system, the cooking time will be reduced to the limit of the maximum productivity of the chefs.
2) The total time for cooking will increase in case the chefs have already reached the maximum productivity due to the increasing number of simultaneous orders.

In order to confirm or reject these hypotheses, additional research, experiments or simulations are required, which are not performed in this project due to its limiting factors.

The first hypothesis is based on the collected information from experts in the restaurant field, who claim that integration of automated system can allow chefs to collect and review incoming orders more quickly, so that they can combine raw material preparation and thus increase their efficiency. Nonetheless, it should be taken into account that efficiency can be increased to a certain limit, which is equal to the maximum productivity of chefs. Exceeding this threshold is expected to increase the order execution time, which leads to the second hypothesis.

In order for a restaurant to achieve maximum benefits from using the automated restaurant service system, it is necessary to minimise the critical time associated with cooking in case of medium and heavy restaurant occupancy. This can be done by adding an additional algorithm to the system, which, at a certain number of orders, changes the digital menu offers to those that require minimal cooking time. For example, it is possible to create a special discount policy when the chefs have almost reached their maximum productivity, giving special prices to the dishes that take less time to cook. The algorithm may be used based on how many orders have been created in the system, but such a calculation may be inaccurate enough, because there may be several customers at one table who will place their total order in the system as one. In such cases, counting of separate dishes may be used. Other approaches and systems can also be used to measure the load, such as using WiFi data and Wireshark's public domain packet capture and protocol analysis software [14].

The use of a digital menu algorithm in case of heavy workload allows the restaurant to increase the productivity of the existing staff and decrease the need for additional labour during peak hours, which can lead to an increase in costs.

In the case of a low-load restaurant, the automated service system mainly benefits customers, as it is possible to reduce the total service time. Additional benefit from the restaurant point of view is the acceleration of chefs' activities, if the first hypothesis is fulfilled. In addition, if a restaurant speeds up customer service in this way, it can maintain a low occupancy rate for a longer period of time by serving more customers.

An automated restaurant service system can significantly decrease waiters' work, and as a result company can evaluate the possibilities of reducing the workforce of waiters and thus also reduce the costs. From this point of view, it is also possible to estimate how much an automated restaurant service system would cost to make it profitable to purchase and maintain it for a restaurant. The direct benefit depends on the number of waiters working at the restaurant, which means that in the case of a larger restaurant, the benefit will also be higher. It is not the only benefit to compare, but it gives a rough idea. To determine the exact benefits, it would be necessary to perform longer experiments or simulations using real data.

The automated restaurant service system can also provide other additional benefits, for example, it is possible to quickly create different user experiences while placing orders by offering customers special offers or analyse past orders for a particular customer to re-offer them, thus helping the customer make a faster decision. In addition, any changes to the menu can be implemented very quickly, and various additional algorithms can be used, such as placing the most popular dishes at the beginning of the digital menu, which automatically change according to their popularity, or potentially increase the number of ordered foods and drinks by offering them right before the customer is finalizing the order.

V. Evaluation of Request Processing Capacity and Speed

In order to evaluate the capacity and speed of the developed system during request processing, the authors generate requests using Gatling tool. Gatling is an open source tool for load testing of web applications. It allows generating a large number of requests and offers metrics to estimate the critical size of the system. The authors use Gatling to simultaneously generate 500, 1000, 10 000 requests to:

1) retrieve menu data;
2) add users;
3) retrieve data about a user;
4) place new orders;
5) retrieve data of active kitchen orders.

Analysing the processing time for 500 and 1000 simultaneous requests, the shortest processing time can be achieved for approximately 500 simultaneous requests when retrieving menu data and creating a new order. This is the maximum capacity for this time, which is less than 800 milliseconds. In addition, when adding users and retrieving their data, with the increase of simultaneous requests from 500 to 1000, processing time also increases significantly. In the first case, almost all requests are processed in 800 ms, but in the second case – processing time for almost all requests has increased and is over 1200 ms. Retrieving data of active kitchen orders, 3 % of 1000 simultaneous requests are not processed at all (Fig. 10).

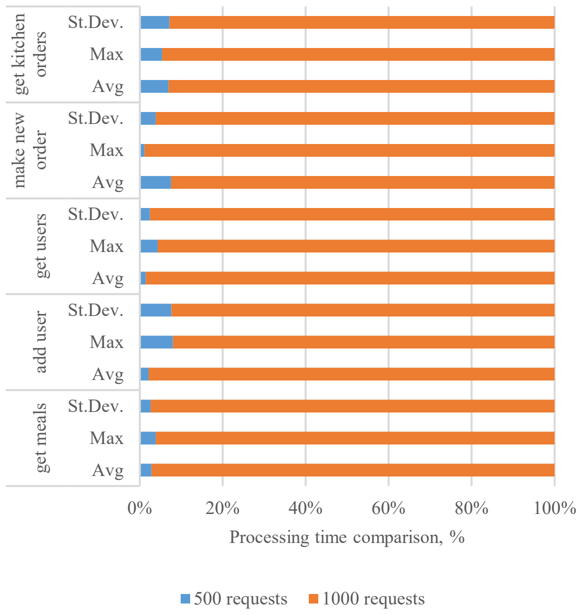

Fig. 10. Comparison of processing time for generated requests, %.

The created system is not able to process 10 000 simultaneous requests and they are rejected. With this number of simultaneous requests, failure to process requests resulting in rejection happens at the infrastructure and database layer.

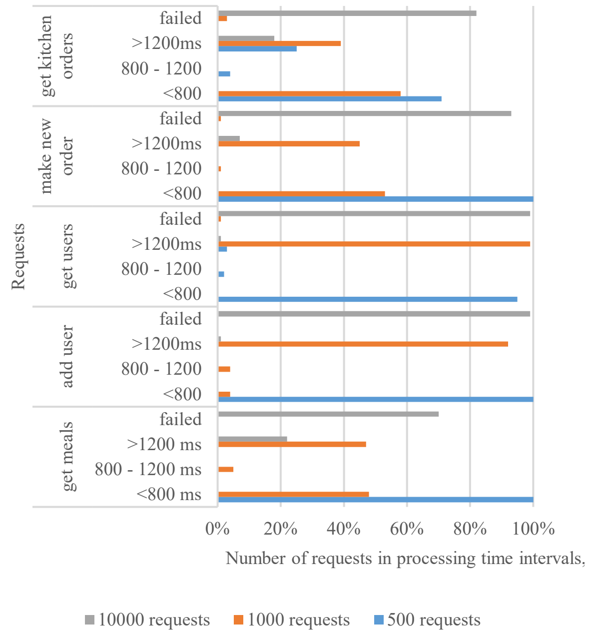

Fig. 11. Comparison of processing time for generated requests, %.

As the number of generated requests increases from 500 to 1000, their average processing time increases significantly.

For 500 simultaneous requests, the average processing time for all request types is less than 1 second, which in practice is often defined as acceptable processing time for applications without interfering with customer's use of the application and its features. For 1000 generated requests, this average time increases and is higher than 1 s. In addition, retrieving user data, the average processing time increases to 18.79 s, which is a very long processing time and will essentially make it significantly more difficult for customers to perform this operation, and potentially will lead to customer dissatisfaction (Fig. 11).

VI. Conclusion

There are various solutions available on the market for an automated restaurant service system based on the digital ordering system and the automatic data transfer to the kitchen application, thus creating a fast solution for restaurants to accept orders. To evaluate the benefits of such a system, the authors developed and experimentally tested its basic functions in order to compare customer service time with and without an automated restaurant service system.

Having performed experiments and the analysis of the time consumption, it can be concluded that it is possible to significantly reduce the time of transferring customer order information to the cook, as well as other time intervals related to the waiter's work. By using an automated system for serving orders in a medium-load restaurant, it is possible to reduce the total waiting time by 52.76 %, but in case of low occupancy – by 40.16 %. It should be taken into account that the time of placing an order in the digital system can be very fluctuating and depends on a specific customer, so a particular restaurant would need to take into account the profile of the average customer and the level of their digital skills.

In the case of automated service, the cooking phase is a critical phase in the overall waiting time, because as the number of parallel orders increases, the workload of cooks will increase as well, slowing down the order execution speed. That is why it is necessary to monitor and analyse the workload of cooks. The automated system allows for various manipulations with the functionality of the overall system, for example, to decrease the workload of chefs at certain times when the number of orders has increased. This can be done, for example, by integrating an order threshold into the system, at which the system would automatically offer customers food that takes less time to prepare at a special offer price. Doing so, it is possible to maximise the productivity of chefs.

An automated restaurant service system in the case of medium- and high-load restaurants can significantly reduce the workload of waiters, as a result of which the company can evaluate the possibilities of reducing the workforce of waiters and also reduce costs.

The authors have generated requests for the created system in order to determine the maximum possible number of simultaneous requests. The automated restaurant system with the specified infrastructure and database is able to process 500 simultaneous requests quickly – in less than 800 ms. The average request processing time increases significantly for 1000 simultaneous requests, but they are rejected for 10 000 requests. The speed of system request processing depends directly on the infrastructure used, as well as the database.

The authors, in consultation with restaurant business experts, have formulated two hypotheses for further research:

1) In the case of an automated restaurant system, the cooking time will be reduced to the limit of the maximum productivity of the chefs.
2) The total time for cooking will increase in case the chefs have already reached the maximum productivity due to the increasing number of simultaneous orders.

References

[1] T. O'Grady, H.-Y. Chong, and G. M. Morrison, "A systematic review and meta-analysis of building automation systems", *Building and Environment*, vol. 195, p. 107770, May 2021. https://doi.org/10.1016/j.buildenv.2021.107770

[2] T. Samad, P. McLaughlin, and J. Lu, "System architecture for process automation: Review and trends", *Journal of Process Control*, vol. 17, no. 3, pp. 191–201, Mar. 2007. https://doi.org/10.1016/j.jprocont.2006.10.010

[3] L. Ivančic, D. S. Vugec, and V. B. Vukšic, "Robotic process automation: Systematic literature review", in *Lecture Notes in Business Information Processing*, pp. 280–295, 2019. https://doi.org/10.1007/978-3-030-30429-4_19

[4] K. Mishra, B. S. Choudhary, and T. Bakshi, "Touch based digital ordering system on Android using GSM and Bluetooth for restaurants", in *2015 Annual IEEE India Conference (INDICON)*, New Delhi, 2015, pp. 1–5. https://doi.org/10.1109/INDICON.2015.7443374

[5] S. N. Cheong, W. W. Chiew, and W. J. Yap, "Design and development of multi-touchable E-restaurant management system", in *2010 International Conference on Science and Social Research (CSSR 2010),* Kuala Lumpur, Malaysia, 2010, pp. 680–685. https://doi.org/10.1109/cssr.2010.5773867

[6] S. Pieskä, M. Liuska, J. Jauhiainen, A. Auno, and D. Oy, "Intelligent restaurant system Smartmenu", in *2013 IEEE 4th International Conference on Cognitive Infocommunications (CogInfoCom),* Budapest, 2013, pp. 625–630. https://doi.org/10.1109/CogInfoCom.2013.6719177

[7] M. Prasad, E. Scornavacca, and H. Lehmann, "Using wireless personal digital assistants in a restaurant: impact and perceived benefits", in *International Conference on Mobile Business (ICMB'05)*, Sydney, NSW, 2005, pp. 69–74. https://doi.org/10.1109/icmb.2005.112

[8] B. Deng, S. Li, B. Zhang, F. Wang, D. Li, and H. Lin, "Iot intelligent restaurant system design*"*" in *ACM International Conference Proceeding Series*, 2019, pp. 1–7. https://doi.org/10.1145/3331453.3361284

[9] YongChai Tan, KienLoong Lee, ZhiChao Khor, KaeVin Goh, KhimLeng Tan and BentFei Lew, "Automated food ordering system with interactive user interface approach*"*, in *2010 IEEE Conference on Robotics, Automation and Mechatronics*, Singapore, 2010, pp. 482–485. https://doi.org/10.1109/RAMECH.2010.5513147

[10] A. B. Amir, Mohd Syazwan Nazmi Bin Abul Kassim, and Mohamad Ikhsan Bin Johari, "Automated food ordering system", School of Computer Sciences, University Sains Malaysia,11800, Penang, Malaysia. [Online]. Available: https://www.academia.edu/31094268/Automated_Food_Ordering_System

[11] A. Baranwal, A. Srivastava, B. Rani, "An innovative approach for online food order management system", *International Journals of Advanced Research in Computer Science and Software Engineering*, vol. 8, no. 3, Mar. 2018, pp. 19–23.

[12] M. M. Chouhan, A. Tiwari, N. Agarwal, P. Patkar, N. Kumbhar, and P. S. Kulkarni, "Automated table ordering system*"*, *International Journal of Advance Research and Development, Computer Science*, vol.2, no. 4, pp. 72–76, 2017. [Online]. Available: https://www.ijarnd.com/manuscripts/v2i4/V2I4-1154.pdf

[13] B. Deng, F. Wang, B. Zhang, D. Li, S. & Lin, H., "Internet of things smart restaurant design scheme", in *ACM International Conference Proceeding Series*, no. 38, Oct. 2019, pp. 1–4. https://doi.org/10.1145/3331453.3361283

[14] J. A. Polack, M. Clark, and J. E. May, "Simulating restaurant traffic using Wi-Fi data (WIP)", in *SCSC '16: Proceedings of the Summer Computer Simulation Conference*, no. 9, July 2016, pp. 1–6. https://doi.org/10.22360/summersim.2016.scsc.014

Liva Deksne has earned her Bachelor and Master degrees in Social Science in Management (2014) and Master degree in Information Technology (2021) from Riga Technical University, Latvia.
Her current position is IT Project Manager at Dynamic Yield and Research Assistant at Riga Technical University. She has gained more than 5 years of experience working in IT field and specializing in web-based projects. Her previous studies focus on data ecosystem based intellectual transportation systems in winter road maintenance.
E-mail: liva.deksne@rtu.lv

Arturs Kempelis has earned his engineering Bachelor (2019) and Master (2021) degrees in Computer Science and Information Technology from Riga Technical University, Latvia.
His current positions are a Lecturer, IT Consultant and Software Engineer in which he has over 5 years of experience, working on various IT projects in Europe. His research interests centre around IoT networks and machine learning technology use cases in precision agriculture.
E-mail: arturs.kempelis_1@edu.rtu.lv

Toms Sniedzins has earned his engineering Bachelor (2019) and Master (2021) degrees in Computer Science and Information Technology from Riga Technical University, Latvia.
His current position is a Software Engineer at TietoEvry. He has gained more than 3 years of experience in the IT field working on various projects. His previous research interests focus on computer vision and current research interests focus on various web technologies, including Web 3.0.
E-mail: tomssniedzins@gmail.com

Armands Kozlovskis is a Software Engineer with more than 5 years of experience in IT field. He has obtained the Bachelor and Master degrees in Information Technology and Computer Science (2021) from Riga Technical University, Latvia.
His best skills are targeted for web development, including JavaScript, React and Node.js. He has participated in many prototype developments for the State Revenue Service, Machine Learning and AI projects and worked with well-established companies and clients. His goals are to expand web-programming knowledge even further and learn team leader capabilities, which would allow reaching higher levels and developing better overall skillset in the IT industry.
E-mail: kozlovskisarmands@inbox.lv