Project Documentation

ISTE-330 Database Connectivity and Access

Restaurants

Document (filename): Restaurants Application

Author(s): Luka Toncic, Dominik Maric, Zoran Mustac, Laura Jurisic

Date (last change): March 25, 2020.

Version (last version): 1.0

|  |  |  |  |
| --- | --- | --- | --- |
| *Version* | *Description of Change* | *Author(s)* | *Date* |
| 1.0 | Draft | Dominik Maric, Luka Toncic, Laura Jurisic, Zoran Mustac | 2020-3-28 |
| 2.0 |  |  |  |

Table of contents

[1 Introduction 3](#_30j0zll)

[1.1 Overview 3](#_1fob9te)

[1.2 Purpose and Scope 3](#_3znysh7)

[1.3 Background 3](#_2et92p0)

[1.4 References 3](#_tyjcwt)

[1.5 Document Overview 3](#_3dy6vkm)

[2 Problem Description and Solution Architecture 4](#_1t3h5sf)

[2.1 Problem Description 4](#_4d34og8)

[2.2 Technologies and Architectural Design 4](#_2s8eyo1)

[2.3 Database Layer and Database Connectivity Layer 4](#_17dp8vu)

[2.4 Business Layer 4](#_3rdcrjn)

[2.5 Presentation Layer 5](#_26in1rg)

[2.6 Areas of particular concern 5](#_lnxbz9)

[3 Requirements 6](#_35nkun2)

[3.1 Context 6](#_1ksv4uv)

[3.2 Functional Requirements 6](#_44sinio)

[3.3 Other (Non-Functional) Requirements 6](#_2jxsxqh)

[4 User Documentation 7](#_z337ya)

[4.1 Graphical User Interface Design 7](#_3j2qqm3)

[4.2 User Manual 7](#_1y810tw)

[5 Installation, Configuration and Acceptance Testing 8](#_4i7ojhp)

[5.1 Installation 8](#_2xcytpi)

[5.2 Configuration 8](#_1ci93xb)

[5.3 Acceptance Testing 8](#_3whwml4)

[6 Final Remarks and Conclusion 9](#_2bn6wsx)

# Introduction

## Overview

Our goal will be to create a database of various restaurants in the City of Zagreb, and allow the user to search for a restaurant based on the type of food they serve. There are many “restaurant finder” apps and websites, but not many desktop applications. We intend to have the same functionality as any website or app and provide fairly detailed information about any desired restaurant, including menu, location, delivery options, etc. The difference our app will have from other typical restaurant finders is the option to search by food type so that users can pick a restaurant exactly by the type of food they feel like eating.

For example, a user can select from a list a certain food they would like to eat, click search, and a textbox will be updated with a list of restaurants that serve it!

## Purpose and Scope

*[Provide an overall description of this document and the project. Explain the purpose of this project and the intended clients. Additionally, discuss the scope of this document (what it describes in the context of the project and also what it will not describe) as well as the intended audience of this document (managers, developers, clients, etc.). Several sentences are expected (****deliverable and milestone #0 and #1****).]*

The main purpose and scope of this document is to clarify the main goal of our group project. Our goal is to provide as much information as possible and to cover all information needed for our project.

This document is intended for developers to see what we have been doing for our program and also to keep track of how it went what were our ideas.

## Background

This document is created by four group members and the main idea of this desktop app is to enable users to search easily, without unclear instructions or possible confusion, even for those users who are not handy with new technology. Our main purpose is for us to learn as much as possible about storing data and desktop application programming. One huge plus is that we will be able to learn how to cooperate nicely with each other, in order to prepare us for the future.

## References

Meetings:

Hopefully, this team will function with consensus, proper and equal contribution, with criteria that have been negotiated already. We will try our best to help each other in any aspect of the project, as some of us may have more knowledge than others, which is perfectly normal and expected, and we will try to balance that till the end of this project.

Deliverables:

Our goal is to hand in a fully working desktop application with a simple but effective user experience which will be easy to use.

Documentation:

Tools, documentation and programming languages that we intended to use:

-Java

-Mysql

Standards and policies:

Four member project.

Deadlines have to be respected.

Equality in contribution to this project.

The final version must be delivered after proper testing.

## Document Overview

The second chapter of this document will be dealing with the programming architecture of this application, including all layers and separation of concerns, as well as a description of the database.

The third chapter will be describing the requirements on the user side of our app, compatibility with different OS, and system requirements.

The fourth chapter will be the user manual and overall help for our users on how to use our application, as well as a description of the GUI and its components, and the functionality of each of its pieces.

In the fifth chapter, we will conclude the document with some final thoughts and possible concerns.

# 2. Problem Description and Solution Architecture

## Problem Description

Most, if not all, restaurant finder apps allow their users to search for a place to eat by sorting them by location, rating, and so on. If you wanted a restaurant that serves a specific food, you’d have to search for it yourself.

Our app intends to allow the user to search for a restaurant exactly by the food they want to eat. Select a food from a predetermined list, click search, obtain a list of restaurants and their locations that serve the food you’d like to enjoy. In the short market research, we’ve conducted, we have found no similar apps.

## Technologies and Architectural Design

We are going to be using Java and MySQL as our programming languages for the project. We are using Java because it is the most widely used programming language today as it comes with its strengths and weaknesses, MySQL as well as a good database programming language. We will try to stick to the DRY principle as best as we can to improve code readability and also apply the Separation Of Concerns.

## Database Layer and Database Connectivity Layer

*[****IMPORTANT:*** *Provide database structure with ERD and database schema as well as data dictionary explanations of entities (tables) and their characteristics/attributes/properties (columns). At least several sentences are expected to describe the model. There should be a physical database model figure for the chosen database with 10-15 tables in a clear relationship, with a description of each database table. The data dictionary is more than welcome – you can do it in the usual table format in Word, where each row represents one attribute from a database table, and you should have it for all attributes and all database tables. Each row should hold information about that attribute, including table name, column name, the data type in the database, short description, constraints (e.g., primary or foreign key) and restrictions (format of the data), and (if unclear) sample data. You should start doing it for the* ***deliverable and milestone #1*** *and finish it no later than* ***deliverable and milestone #2****).]*

There will be 10-15 tables in our database that will store our restaurants. Each of the tables will have a finite amount of info that will be callable. It will store the name of the restaurant and the location of the restaurant from which we will be able to locate on our type of “google maps” page. From them, you can also extract some information about the restaurants respectfully.

## Business Layer

*[Provide Business Layer description, design and explanation, as well as describe connections to Database Connectivity Layer and Presentation Layer. At least several sentences are expected to describe how the Business Layer will be constructed, what its purpose is, and how it communicates with Database Connectivity Layer (below) and Presentation Layer (above). You should start doing it for* ***deliverables and milestones #1 or #2*** *and finish it no later than* ***deliverable and milestone #3****). ]*

The Business Layer will be connected to the Database Connectivity Layer in a way that whatever is being inputted by the user on the Business Layer it will then take the information from the database and show it to the user. It will be connected to both the Database Layer and the Presentation Layer so that it can output the correct information the user wants to see.

## Presentation Layer

*[Provide Presentation Layer description, graphical user interface (GUI) design, including structure, layout and explanations, as well as a description of used technologies. At least several sentences are expected to describe how the Presentation Layer will be constructed, what its purpose is, and how it communicates with the Business Layer (below) and the users (clients). You can also include all possible actions, menus, and options. You should start by doing some prototypes or wireframes for* ***deliverables and milestones #1 to #3*** *and finish it no later than* ***deliverable and milestone #4****. In the final* ***deliverable #5*** *you can even provide some screenshots.*

The page will consist of a search bar at the top, which when you’ve typed a certain food you want into it then you will get the restaurants that serve it. Beneath the search bar there will be a list of restaurants that are available on our site where you can choose and read more about a restaurant that you are interested in. The connection it has to the Business Layer is what it is communicating to the Database Layer to show on the users screen.

## Areas of particular concern

*[In this chapter, you should provide identification of areas of particular note or concerns. It could be about prerequisites (which must be respected) and assumptions, as well as possible risks for your project. Those could be related to an organization, planning, resources, technologies, and availability, as well as team members. You can describe a plan on how to mitigate those risks. You should start doing it for* ***deliverables and milestones #1 to #3*** *and finish it no later than* ***deliverable and milestone #4****.]*

Some of the problems we might run into is not being very clear what we want to do given the current situation that is occurring in the world, it will be challenging to do everything online. We might run into the problem of not getting all the necessary information that we need for the restaurants that we want to use. Quite possibly even the google maps part of the application could be quite troubling. Another issue we will run into is getting info from the database to show on the GUI depending on the restaurant you choose. But, we will try to work around those issues to the best of our ability in order to provide our final product.

# 3. Requirements

## Context

*[Provide a description of the application in the broader context, how it will work within the environment of other systems (e.g., payment systems if there is some kind u purchasing involved), with explanations as applicable. The context of a system refers to the connections and relationships between the application and its environment. At least several sentences are expected and it relates to* ***deliverables and milestones from #1 to #5****.]*

The application is, as already stated, a desktop interface which incorporates a database of restaurants in Zagreb. People will search for a specific kind of food (e.g. pizza, burgers, pasta), and based on that input - the app will filter all restaurants and display the ones that offer the specified food.

Additionally, the application will not include any kind of payment systems, nor delivery options. It just serves as a filter when a person wants to eat specific food. Also, the app will include broad information about all restaurants in the database, photographs and location on the map using Google Maps API.

If some great idea comes up during development, we might incorporate it at a later date.

## Functional Requirements

*[****IMPORTANT:*** *List, name and explain all key functionalities – there should be approx. 15 functional requirements listed. You should provide a table where each requirement is named (short code or name, could combine letters with numbers) and a detailed description, as well as who is responsible or performing the action associated with this functional requirement. You could also use use-cases (diagrams or descriptions), including use-case names, actors, events flow, exceptions, and special requirements. Include user requirements if necessary (users, roles, privileges) and associate with specific functional requirements. This should be started for the* ***deliverable and milestone #0****, and the proposal must be agreed upon and finished with* ***deliverable and milestone #1****. Later, with the approval of the client, it could be revised in* ***deliverables and milestones #2 - #4****).]*

1. Search option

Search option is the most important tool that separates our Restaurants application from others. The app allows to directly specify which food someone is interested in and get back filtered restaurants ordered by the nearest to one’s location.

1. “Segway control”

Segway control is located in the home page of the app, in the top of the screen and it is used to quickly filter restaurants based on the type of restaurants. For example, it could be used if a person doesn’t know what to eat specifically, yet would like to eat Asian cuisine - segway control allows it to filter restaurants based on the chosen option - Asian.

The app will include 6 options in the segway control: Pizza, Burgers, Asian, Meat, Fish and Mediterranean.

1. Home screen

Home screen will incorporate the option to specifically search and to search based on the segway control. It will also include the list of restaurants randomly displayed, so that if a person would like to look at restaurants without the search options - it is possible.

1. Information about restaurants

When on the home screen - basic info about the restaurant, such as the name, photo and basic info will be displayed. However, when a restaurant is clicked - another user interface opens. That UI includes all possible information about the restaurant from our Database - Full description, menu, images, and so on....

1. List of available food

When a desired restaurant is clicked on - the list of available food, based on a restaurant, is displayed. The available food may always be added through the database, so that when restaurants update their menus, it will never be an issue.

1. Photographs of restaurants

Professional looking images of restaurants, the interior, as well as exterior will be available for the users to look at. The photos will also be incorporated into the database.

1. Map

There will be a map user interface that will feature the list of all restaurants that we offer. Each of them will be visible as a red clickable button, like in other Google Maps apps. When the button is clicked, it shows additional features like “Get Directions”, “Show Info” and “Go to Website”. Each of those is self-explanatory, so I won’t go too deep with explaining them.

1. Directions

When a person clicks on the directions button, whether it is in the Map UI or in Restaurants Info UI - the same thing happens - the user relocated to the Maps UI and the direction to the restaurant is displayed using Google Maps API and based on the user’s current location.

1. Information in Map UI

When a restaurant is clicked on in the Map UI, an option “Information about the restaurant” appears to the user. When clicked - the user is taken to the Restaurants view where he or she may see everything about the restaurants.

1. Current location

When current location is clicked - it shows the user's current location on the map.

1. About US page

All information about creators of the application.

USE CASES

Mary is hungry and would like to order some food off the Internet. Like other teen girls, she is not yet informed about all the available restaurants, which are growing in number daily. What Mary knows is that she wants to eat some Pizza. Not just every Pizza, but a Hawaii pizza with pineapple. Insead of using our smart segway control that gives her easy access to 6 most common filter options (Pizza, Burgers, Meat, Asian, Mediterranean and Fish), she uses our smart search option.

When clicked on the Detailed Search button - she is mesmerized by the option to specify the type of pizza she wants - Hawaii. Instead of filtering and getting her all the restaurants with any type of pizza - she is welcomed by the list of restaurants that offer specifically the Hawaii pizza. One of them is Duksa. Mary has heard about Duksa before, but she is eager to find out more about the restaurant. She clicks on it and is able to see photographs of Duksa’s interior.

Marry now understands that she won’t call the delivery (although contact information is at the tip of her hands in our app), but would like to go there in person. Mary still doesn’t drive, and calls her friend Jane to pick her up. Jane comes to Mary, who lives in the center and picks her up. Jane never heard about Duksa before, and she forgot her phone home. What’s even worse, Mary doesn’t have the Internet on her phone.

Thanks to our group, when Mary was searching for her Pizza - she managed to look up directions to Duksa on our application and now Mary knows how to get there. Not only that, Mary used the contact information on our Map to call the restaurant and reserve seats for her and her friend. In the end, they had a wonderful time in Duksa and when they came, Mary decided to send the message to creators of the application. Fortunately, little did Mary know - the contact info of the group is incorporated into the app, and Mary didn’t have to lose time to search us on the Internet.

Johnny was home and extremely hungry from yesterday’s drinking party. All he knows is that he wants to eat some burgers tonight to help his hangover. He comes across our application and decides to try it out. Immediately, he is greeted by the beautiful design. Johnny is hungover, and he isn’t in a position to perform detailed searches, because long staring at the display gives him headaches. Fortunately for him, we have a Segway Control option. Amazed by our piece of software, he decides to click on the burger icon in the segway control. Immediately, all restaurants that have burgers are displayed.

Johnny is a regular burger eater, and he knows almost all of them. However, he hasn’t heard of one of them - Submarine. He decides that he wants to know more and so he clicks on the restaurant. Additional info is displayed. He reads the information, sees the pictures and sees the list of burgers that they offer. What’s more, Johnny sees the contact information right there. He calls them and orders some burgers. “What a wonderful application” Johnny thought...

## Other (Non-Functional) Requirements

*[Describe the non-behavioral and non-functional requirements, including hardware and software requirements (e.g., platforms needed to support this application), programming interfaces, and any operational requirements (how the system will run and communicate with the environment). You could also provide information about application availability (time of day or week), general performance (how fast it should be in client responses), capacity (how many concurrent users or connections it will support), error handling (how is it handled), conventions used, security and similar if necessary. This should be started for the* ***deliverable and milestone #1****, and it could be revised in* ***deliverables and milestones #2 - #5****).]*

The application will include some non-functional requirements such as good design, ease of use and wide platform support. Software requirements are: any modern desktop or laptop computer that runs newer versions of Java and MySQL. The application should work on both Windows and MacOS operating systems. It will also use an Application Programming Interface - Google Maps, which is normally supported by a wide variety of computers.

Additionally, the app will function all the time and is not dependent on anything aside from the database availability. The speed of the program will be dependent on the clients’ hardware and Internet access. However, it shouldn’t be an issue for modern computers of any age and configuration, as long as they are supported by the latest Java and MySQL updates.

Also, errors are handled by exceptions, and should not be an issue in our case. We don’t deal with transitions and some other vulnerable things.

# 3. User Documentation

## Graphical User Interface Design

*[You should provide user design and user experience description, as well as a description of used technologies. This should be started for the* ***deliverable and milestone #3****, and should be finished with* ***deliverable and milestone #4****.]*

Our goal for the user design is to make it as simple as possible, however we are not expecting to get it right from the first try although we will do our best. We are going to try out a few designs and choose the final one which has to be chosen by consensus between group members. Technologies that we are going to use will be IntelliJ IDEA more likely than Netbeans for the sake of simplicity. Our expectations for the design are to be simple, clear, manageable, functional and nice to the eye.

## User Manual

*[This should provide expected usage of the available functionalities, could be divided per user roles, and should include screenshots with detailed descriptions. This should be started for the* ***deliverable and milestone #3****, and should be finished with* ***deliverable and milestone #4****.]*

We will provide a GUI with buttons and you will be asked the question, what kind of food you will like to eat? Then you will be able to press a button and you can see the available restaurants. The functionality will mainly be just to press a button and you will get information about the type of food you want.

# 4. Installation, Configuration and Acceptance Testing

## Installation

*[Provide a technical manual – prerequisites and installation process description details. Should be finished with* ***deliverable and milestone #5****.]*

You will need an IDE to open this program of ours and once you run it you will have a question asked and you will be able to choose what kind of food you want to eat.

## Configuration

*[Technical manual should also hold configuration detail and default values for this project to work. Should be finished with* ***deliverable and milestone #5****.]*

You need the newest version of java and all you have to do is run the application and everything necessary will be displayed.

## Acceptance Testing

*[Some acceptance testing should be performed to determine if the requirements are met – you should describe typical usage and tests to be executed for the application. Should be finished with* ***deliverable and milestone #5****.]*

You can run the application and then just press the buttons and the given information will be presented. You first get the option to choose between type of foods worldwide, after choosing which one you like the most you will be prompted with the options. After you decide what you feel like eating just press the option to add it on your list.

# 5. Final Remarks and Conclusion

*[Usually, this chapter should be started later, and at least partially filled with* ***deliverable and milestone #3 or #4****, and should be finished with* ***deliverable and milestone #5****. You should summarize the experiences, both in terms of the produced results and work on the project. List all project deliverables, as well as positive (and negative) experiences and concerns. Comment on missing functionalities and possibilities for improvement and extensions. Estimate project effort (person-hours) and how it was distributed in time and per team roles. This chapter can also include a work log summary for all team members (for each day who did what).]*

The result overall is not what we have expected in the beginning. We are missing some features like the google maps for instance that we didn’t incorporate. It was a learning experience as to what not to do in the future and what to improve on. I believe that in some sense with the finished product, some have done more than the others.