TARTU UNIVERSITY
FACULTY OF SOCIAL SCIENCES

NARVA COLLEGE
STUDY PROGRAM “INFORMATION SYSTEMS DEVELOPMENT“

Eduard Vassiljev

ANDROID APPLICATION FOR SEARCHING COOKING RECIPES BY INGREDIENTS LIST

Diploma Thesis

Supervisor: Pavel Kodotšigov

NARVA 2021

Olen koostanud töö iseseisvalt. Kõik töö koostamisel kasutatud teiste autorite tööd, põhimõttelised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

....

Töö autori allkiri ja kuupäev

Non-exclusive licence to reproduce thesis

I, Eduard Vassiljev (date of birth: 22.06.1999),

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright.

“Android application for searching cooking recipes by ingredients list” supervised by assistant Pavel Kodotšigov

2. I am aware of the fact that the author retains the right referred to in point 1.
3. This is to certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Act.

Narva, 17.05.2021

CONTENTS

LIST OF TERMS AND ABBREVIATIONS

IDE - integrated development environment.

JAVA - object-oriented programming language.

HTML – hypertext markup language.

MYSQL - relational database management system.

SQL - structured query language.

AWS - Amazon web services.

AWS S3 - Amazon simple storage service.

AWS RDS - Amazon relational database service.

GUI – graphical user interface.

DB – database.

APP – mobile application.

DBMS – database management system.

ID – identificator.

API – application programming interface.

JSON – JavaScript object notation.

PHP – scripting language.

1. INTRODUCTION

1.1. PROBLEM

The problem is that many people, such as students, young people, single people and children, do not have the experience of cooking with the ingredients currently available. People want cooking recipes on their smartphone to quickly prepare food from the refrigerator. A similar application's is available only in English and is not adapted to products sold in Estonia and cooking recipes common in Estonia.

1.2. THESIS GOAL

The goal of this thesis is to create an application for smartphones that is primarily oriented to Estonian products and cooking recipes common in Estonia, add Estonian and Russian language support. Create a database in which recipes will be stored and ensure maximum availability for the database. Create a file server that will store images and other recipe-related files. It is also very important to develop a system with which recipes can be placed in the database.

1.3. THESIS TASKS

To achieve this goal, we need to complete the following tasks:

- Conduct research and find out what similar mobile applications already exist on the market.
- Create a list of disadvantages and take into account the advantages of existing solutions.
- Choose a list of tools for developing an application.
- Develop a mobile application that meets the requirements.
- Develop a database for the application.
- Develop a program for adding new recipes to the database.
- Develop API for dataflow between database and mobile application

1.4. OUTLINE

The first chapter is an introduction. In the first chapter, the author describes the existing problem and proposes a solution.

The second chapter is an overview of existing solutions, identification of their disadvantages and comparison with the author's idea.

In the third chapter, the author lists the technologies and tools used in the work.

In the fourth chapter, the author describes the system development process. It includes:

- list of functional and non-functional requirements.
- Use cases and dataflow diagrams.
- database structure and its development.
- mobile application structure and its development.
- system structure for adding recipes to the database and its development.
- file server and relational database server deployment and structure.
- application deployment.
- API description

In the fifth chapter, the author will list the functionality implemented in the system and the functionality that the author was unable to implement.

In the sixth chapter, the author describes the future development of the application and lists the functionality that can be added in the future.

2. EXISTING SOLUTIONS

In this chapter author overview existing solutions, making identification of their disadvantages and comparison with the author's idea.

There are many similar applications, but the author did not consider many of them in detail, since they have significant drawbacks.

To check the functionality of applications, the author will indicate the same products in each application, the following product names will be used:

- Milk
- Egg
- Bacon

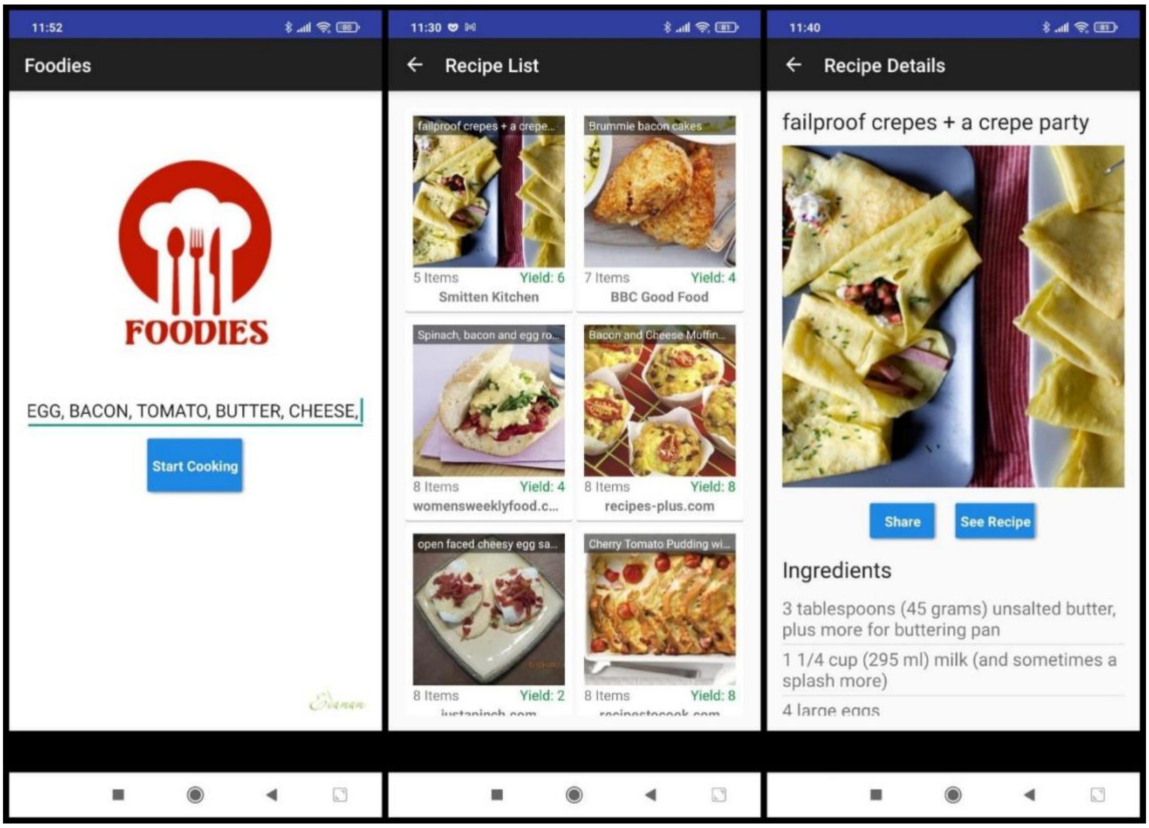
- Tomato
- Butter
- Cheese

2.1. Foodies

The app was last updated on December 8, 2016, the app does not support Estonian and Russian language.

In this mobile application, the user must specify the ingredients in the search bar separated by commas, in the author's opinion, this is an inconvenient way to enter products. After specifying the products and clicking the start button, the application tells you that it works best when four or fewer products are selected. The author ignored this message and continued with the current list of products. Dishes are arranged in 2 rows on the dishes page, each dish has a name, the number of necessary products for cooking, an image of the dish and a link to the source. The recipe details indicate the amount of ingredients for the dish, to view the recipe, click the "View recipe" button After clicking on the button, the application opens the website inside the application.

The application displays different sites depending on the recipe, which in turn negatively affects the design. The displayed pages are inconvenient to view because there is too much non-recipe information on them. It also negatively affects the design of the application, as each website that the application links to has a different design.



EGG, BACON, TOMATO, BUTTER, CHEESE,

failproof crepes + a crepe party

3 tablespoons (45 grams) unsalted butter, plus more for buttering pan

1 1/4 cup (295 ml) milk (and sometimes a splash more)

4 large eggs

Figure 1. *Foodies application main menu, dishes list and recipe page (Source: author)*

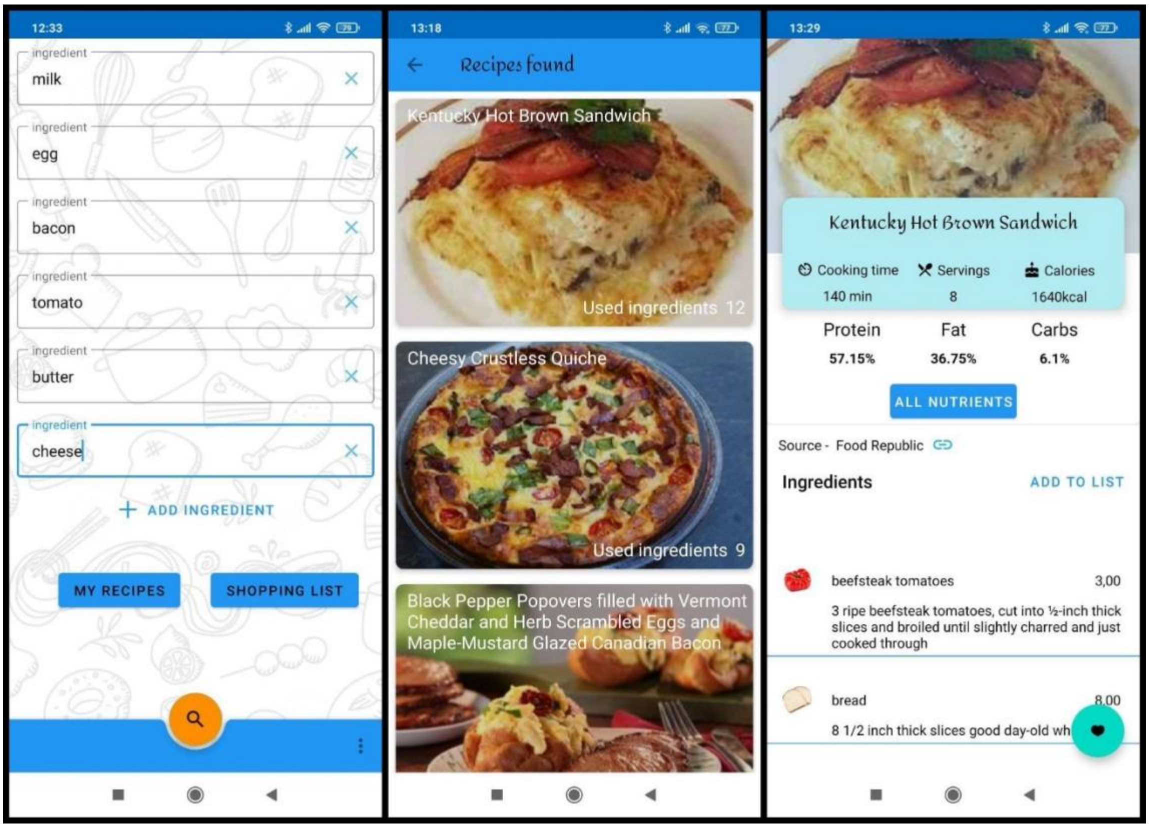
Application Disadvantages:

- There is no support for Estonian and Russian.
- Inconvenient input of the desired products.
- The app reports that it does not work well with more than four products.
- Poor design.
- There is no accounting for food allergies.
- The application offers recipes with those products that the author did not indicate.

2.2. What 2 Eat

The app was last updated on March 11, 2021, the app does not support Estonian and Russian language.

In this application, the user specifies the products line by line, after specifying the products and pressing the "magnifying glass" button, the user is taken to the recipes page. Recipes are arranged in a row, there is the name of the dish, the number of ingredients used and the image of the dish. By clicking on the desired recipe, the author goes to the recipe page, here the number of servings, cooking time, number of calories and a list of ingredients are indicated, to view the cooking instructions, you need to press a special button, in this case the instructions will open. The app also has a shopping list and saved recipes, user can add a recipe to favorites list.



ingredient: milk
ingredient: egg
ingredient: bacon
ingredient: tomato
ingredient: butter
ingredient: cheese

Kentucky Hot Brown Sandwich — Used ingredients 12
Cheesy Crustless Quiche — Used ingredients 9
Black Pepper Popovers filled with Vermont Cheddar and Herb Scrambled Eggs and Maple-Mustard Glazed Canadian Bacon

Kentucky Hot Brown Sandwich

Cooking time	Servings	Calories
140 min	8	1640kcal

Protein	Fat	Carbs
57.15%	36.75%	6.1%

Source - Food Republic

Ingredients

beefsteak tomatoes 3,00
3 ripe beefsteak tomatoes, cut into ½-inch thick slices and broiled until slightly charred and just cooked through

bread 8,00
8 1/2 inch thick slices good day-old wh

Figure 2. *What2Eat application main menu, dishes list and recipe page (Source: author)*

Application Disadvantages:

- There is no support for Estonian and Russian.
- The application offers recipes with those products that the author did not indicate.
- There is no accounting for food allergies.

2.3. Supercook

The app was last updated on January 11, 2021, the app does not support Estonian and Russian language, it has more than 500 000 downloads on the Google play market.

This application has a good design and intuitive interface. In the main menu, the author indicates the desired products, this can be done using a search query, voice commands and by pressing buttons. The products are divided into categories, with a very large number of products to choose from.

After choosing the products, the author clicks the "see recipes" button, and gets to the menu with recipes. There is also a division into categories, for example "3 ingredients or less" or "low calorie". The application shows how many dishes can be prepared from the available products.

After clicking on the recipe, the author is taken to the recipe page. This page contains tags, cooking times, food image, ingredients, calories and user ratings. It is also possible to add a recipe to your favorite dishes, but for this you need to be a registered user. To view cooking instructions, the application redirects the user to a web browser and opens different websites depending on the recipe.

Application Disadvantages:

- There is no support for Estonian and Russian.
- The application redirects the user to a web browser and opens different websites depending on the recipe.
- Registration required to add a dish to your favorites list

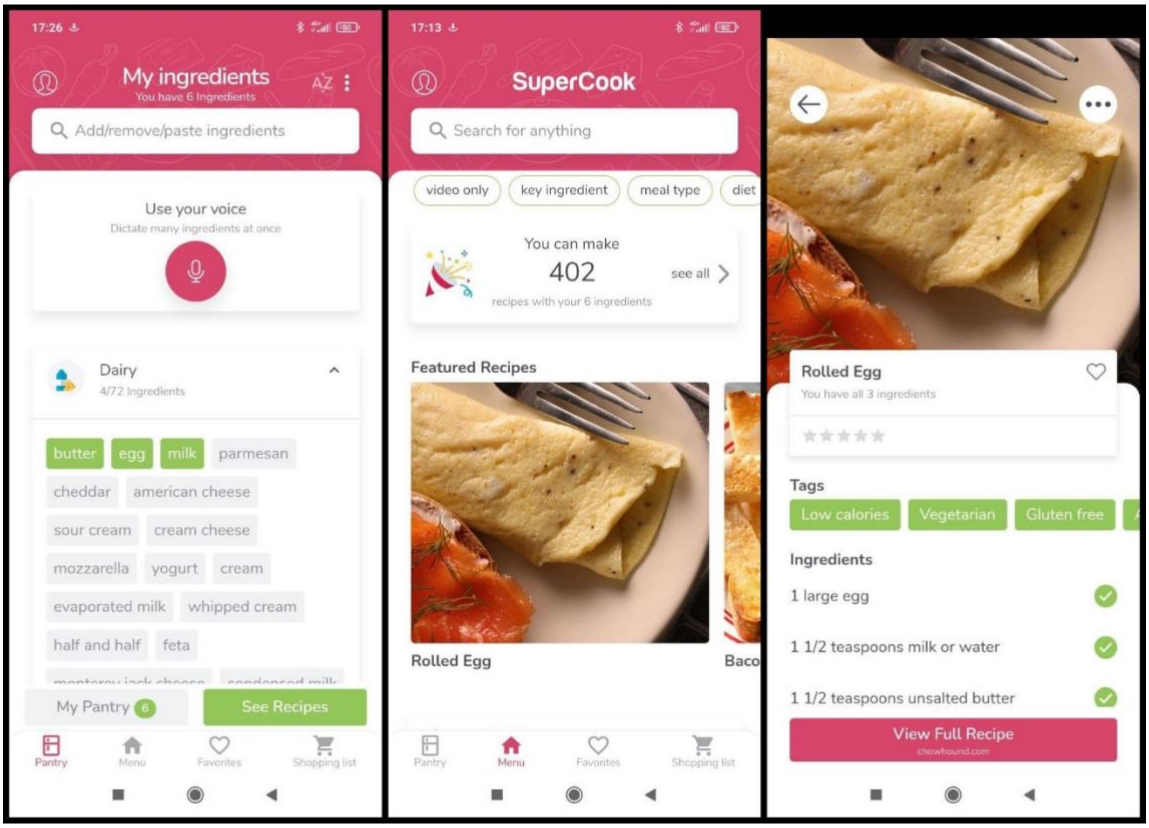


Figure 3. *Supercook application main menu, dishes list and recipe page (Source: author)*

3. USED TECHNOLOGIES AND TOOLS

In this chapter, the author of the thesis lists the technologies and tools used to solve the problem, and also gives a brief description of them.

3.1. MySQL

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. MySQL Server works in client/server or embedded systems. (MySQL Reference Manual: What is MySQL, 2021).

3.2. MySQL Workbench

MySQL Workbench is a graphical tool for working with MySQL servers and databases. MySQL Workbench fully supports MySQL server versions 5.6 and higher. It is also compatible with older MySQL server 5.x versions, except in certain situations (like displaying the process list) due to changed system tables. It does not support MySQL server versions 4.x (MySQL Workbench Manual: What Is New in MySQL Workbench, 2021).

MySQL Workbench functionality covers five main topics:

- SQL Development
- Data Modeling (Design)
- Server Administration
- Data Migration
- MySQL Enterprise Support

3.3. SQL

Structured Query Language (SQL) is a programming language that is typically used in relational database or data stream management systems.

It was developed by IBM in the early 1970s and is now an official standard recognized by the American National Standards Institute (ANSI) and the International Organization for Standardization (Structured Query Language (SQL), 2021).

3.4. Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA. On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps (Meet Android Studio, 2021).

3.5. Java

Java is a programming language and computing platform first released by Sun Microsystems in 1995. There are lots of applications and websites that will not work unless you have Java installed, and more are created every day. Java is fast, secure, and reliable. From laptops to datacenters, game consoles to scientific supercomputers, cell phones to the Internet (What is Java technology and why do I need it, 2021).

3.6. Amazon Web Service – S3

Amazon Simple Storage Service (Amazon S3) is storage for the internet. You can use Amazon S3 to store and retrieve any amount of data at any time, from anywhere on the web.

You can accomplish these tasks using the AWS Management Console, which is a simple and intuitive web interface (Getting started with Amazon S3, 2021).

3.7. Amazon Web Service – RDS

Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the AWS Cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks (What is Amazon Relational Database Service, 2021).

3.8. HTML

HTML is a standardized markup language for documents on the World Wide Web. Most web pages contain HTML markup. It is interpreted by browsers; the resulting formatted text is displayed on the screen of a computer or mobile device (HTML, 2021).

3.9 XML

XML stands for extensible markup language. A markup language is a set of codes, or tags, that describes the text in a digital document. The most famous markup language is hypertext markup language (HTML), which is used to format Web pages. XML, a more flexible cousin of HTML, makes it possible to conduct complex business over the Internet (Explaining XML, 2000).

3.10 PHP

PHP is a server side scripting language. that is used to develop Static websites or Dynamic websites or Web applications. PHP stands for Hypertext Pre-processor, that earlier stood for Personal Home Pages (What is PHP, 2021).

3.11 PYTHON

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together (What is Python? Executive Summary, 2021).

3.12 TKINTER

Tkinter is Python's de-facto standard GUI (Graphical User Interface) package. It is a thin object-oriented layer on top of Tcl/Tk (TkInter, 2021).

4. DEVELOPMENT

In this chapter, the thesis author describes the application development process, main features of the developed application also will be described.

4.1 STRUCTURE OF THE SYSTEM

4.1.1 Main Components

Knowing the disadvantages of similar solutions, the author of the thesis can identify a list of what needs to be added to his own solution.

In order to achieve the desired result, the author must decide how his solution will work. It should be possible to store, add, delete recipes. The user must be able to view the recipes using the phone on the android system. Recipe files (images, recipes) must be stored on the network and available 99.9% of the time of the year. A person without IT skills should be able to add a recipe to the system. recipes should be correctly displayed on any screen resolution on a smartphone. the application should not be able to communicate directly with the database.

4.1.2 Use Cases

The image (**Figure 4**) shows what rights people who interact with the system have. User can view the recipes. Moderator able to view the recipes, add new recipes and add data related to the recipe (images, html files). Administrator, in turn, can do everything, including deleting recipes and related data.

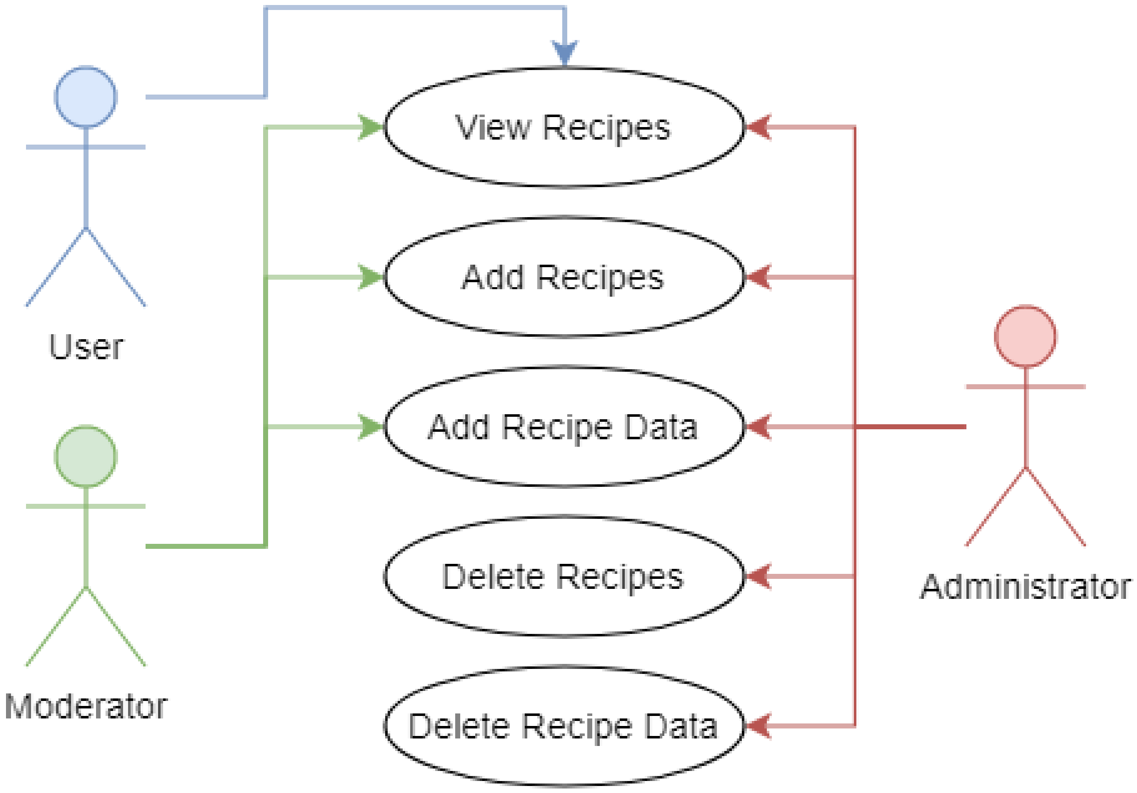


Figure 4. *System Use Case (Source: author)*

4.1.4 Functional Requirements

Functional requirements answer the question "What does the software have to do?" (for example, the system must allow the goods to be ordered). In the above list, functionality is mainly determined by functional suitability indicators.

F1. The application should be able to let the user choose:

- Is he a diabetic?
- Does he have lactose intolerance?
- Is he a vegetarian?
- Is he allergic to nuts?

F2. The application should be able to select products by clicking on buttons

F3. Application must be able to communicate with API

F4. The application must be able to access the Internet

F5. The program for adding recipes must be able to send images and data of the recipe to the Cloud storage by pressing a button

F6. The program for adding recipes should be able to save recipe data by pressing a button

F7. The program for adding recipes should be able to send recipes and all related information to the database

F8. The API must be able to receive data from the Android application and send a request to the database, and also it must be able receive data from the database and send it to the Android application

F9. The application must have recipes in Russian, Estonian and English

F10. The application must be able to save user settings (vegetarian, lactose-intolerance, diabetes, Nuts allergy)

4.1.5 Non-Functional Requirements

Non-functional requirements answer the question "How should the software perform the required functions?". For example, the system response time must be within predetermined limits (efficiency), the system must operate smoothly for certain periods of time (reliability).

N1. The Android application should have a clear and simple design

N2. The program for adding recipes should have a simple design so that a person who does not understand IT can understand how to add a recipe

N3. The database should be available 99.9% of the time of the year

N4. Recipes should be displayed correctly on any smartphone on the Android system

N5. The program for adding recipes should be able to generate html files of recipes in English, Russian and Estonian languages

4.1.6 Dataflow

The diagram (**Figure 5**) shows how data flows between the user, the application, the API, and the database.

In the diagram (**Figure 5**), the user sends the products that he has chosen, after that the application sends a request to the API, the API, in turn, sends a request to the database. The database sends a response in the form of recipes that contain at least one of the products specified by the user in the API. API, in turn, filters out those recipes that do not have all the products specified by the user, after that it sends a response to the application and the user sees only those recipes that have products specified by the user.

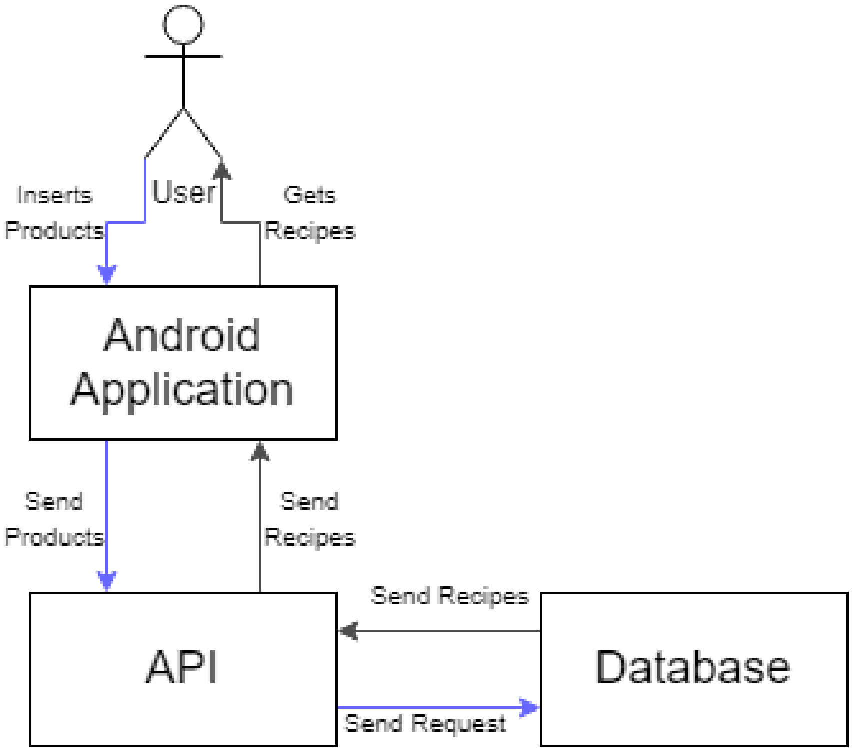


Figure 5. *Dataflow diagram, user side (Source: author)*

The diagram (**Figure 6**) shows the interaction between the administrator / moderator, program, Amazon S3 and database.

In the diagram (**Figure 6**), the moderator / administrator sends images of the recipe, basic information about the recipe and a description in Russian, English and Estonian to the program. The program sends images and generated html files to Amazon S3, when recipes and images are uploaded to Amazon S3, the application sends all information to the database, and sends the program an answer whether it was able to add a recipe or an error occurred.

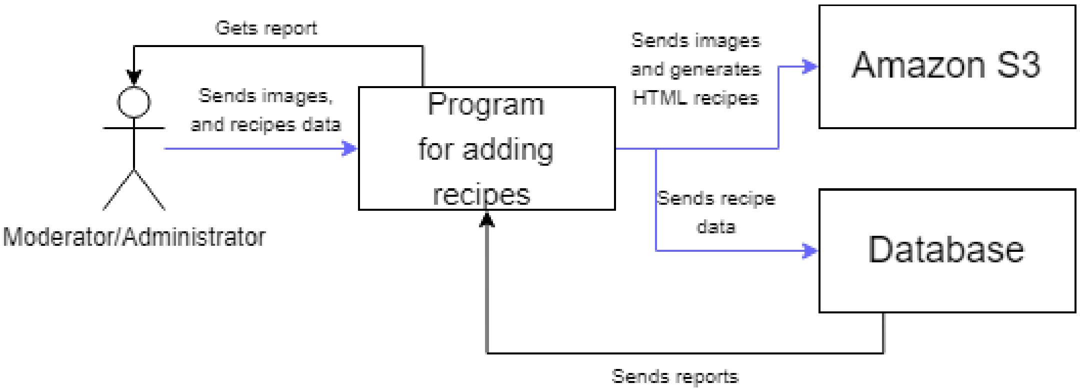

Gets report
Sends images, and recipes data
Program for adding recipes
Sends images and generates HTML recipes
Amazon S3
Sends recipe data
Database
Sends reports
Moderator/Administrator

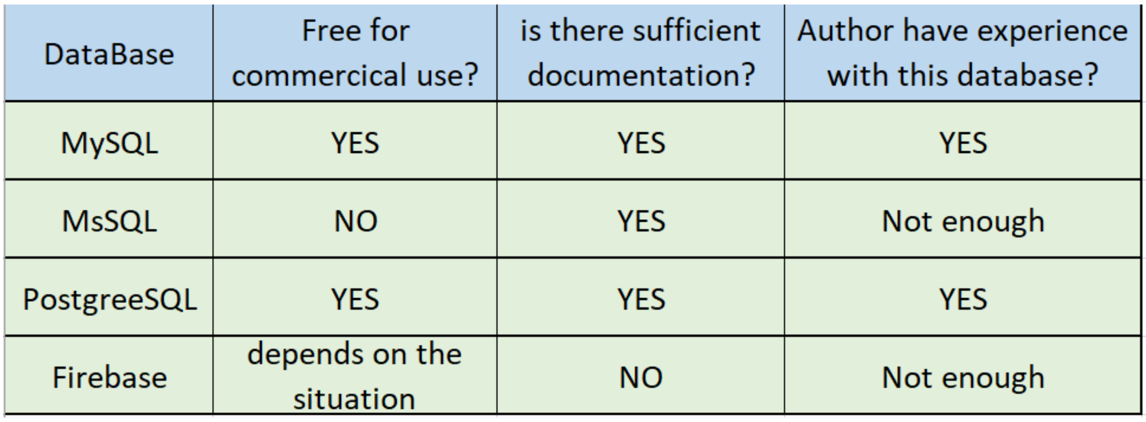
Figure 6. *Dataflow diagram, moderator/administrator side (Source: author)*

4.2 Database

The author considered several databases for this task, namely:

- MySQL
- Firebase
- MariaDB
- MsSQL

The author did a short research and chose the most suitable option for him. In the table (**Figure 7**), the author compared these databases according to criteria important to the author. MsSQL is not suitable for the author as it is paid for commercial use and the author has insufficient experience with this database. Firebase is not suitable for the author, since the author does not have enough experience with this database and although free use for commercial purposes is available, in some situations it may be necessary to change the tariff plan to a paid one. PostgreeSQL and MySQL is ideal for the author, they are free for commercial use, they have enough documentation, in case the author has any questions - these databases have a large community, and the author has sufficient experience with both databases. The final choice fell on MySQL since the author has more experience with it than with PostgreeSQL.



DataBase	Free for commercical use?	is there sufficient documentation?	Author have experience with this database?
MySQL	YES	YES	YES
MsSQL	NO	YES	Not enough
PostgreeSQL	YES	YES	YES
Firebase	depends on the situation	NO	Not enough

Figure 7. *Database comparison (Source: Author)*

The diagram (**Figure 8**) shows how author's database and tables in it looks like.

The recipe table (**Figure 8**) has a large number of columns. The author will briefly describe each column.

1. **Recipe_ID** – *Stores the recipe identifier, cannot be empty, automatically filled when adding a recipe.*
2. **Recipe_name_eng** – *Stores the name of the recipe in English, cannot be empty.*
3. **Recipe_text_ url_eng** – *Stores a link to the generated recipe in Estonian, cannot be empty.*
4. **Recipe_name_rus** – *Stores the name of the recipe in Russian, cannot be empty.*
5. **Recipe_text_ url_rus** – *Stores a link to the generated recipe in Russian, cannot be empty.*
6. **Recipe_name_est** – *Stores the name of the recipe in Estonian, cannot be empty.*
7. **Recipe_text_url_est** – *Stores a link to the generated recipe in Estonian, cannot be empty.*
8. **Recipe_image** – *It stores a link to the main image that the user can see in the recipe selection panel, cannot be empty.*
9. **Cook_time** – *It stores a string, namely, the time it takes to prepare a dish from a recipe, cannot be empty.*
10. **Calories_per_100g** – *It stores a string, namely the number of calories per 100 grams of a recipe dish, cannot be empty.*
11. **Vegeterian** – *It stores a number, if* **0**, *then the recipe is not suitable for vegetarians, if* **1**, *then the dish is suitable for vegetarians, it cannot be empty.*

12. **Lactose_Free** – *It stores a number, if* **0***, then the recipe is not suitable for people with lactose intolerance, if* **1***, then the dish is suitable for vegetarians, it cannot be empty.*
13. **Diabete** – *It stores a number, if* **0***, then the recipe is not suitable for people with diabetes, if* **1***, then the dish is suitable for people with diabetes, it cannot be empty.*
14. **Nuts_allergy** – *It stores a number, if* **0***, then the recipe is not suitable for people with nut allergies, if* **1***, then the dish is suitable for people with nut allergies, it cannot be empty.*
15. **Sum_of_products** – *Stores the number of products needed to prepare a dish from a recipe, it cannot be empty.*

The Product List table (Figure 8) has 3 columns. The Android application and the program for adding recipes know which product is equal to the product identifier (Product_ID), so there is no need to store the names of the products in the database. The author will briefly describe each column.

1. **Product_list_ID – Stores the product identifier in the recipe, a unique value, cannot be empty.**
2. **Recipe_ID – Stores the recipe identifier from the Recipe table, cannot be empty.**
3. **Product_ID – Stores the identifier of the product that is present in the recipe, cannot be empty.**

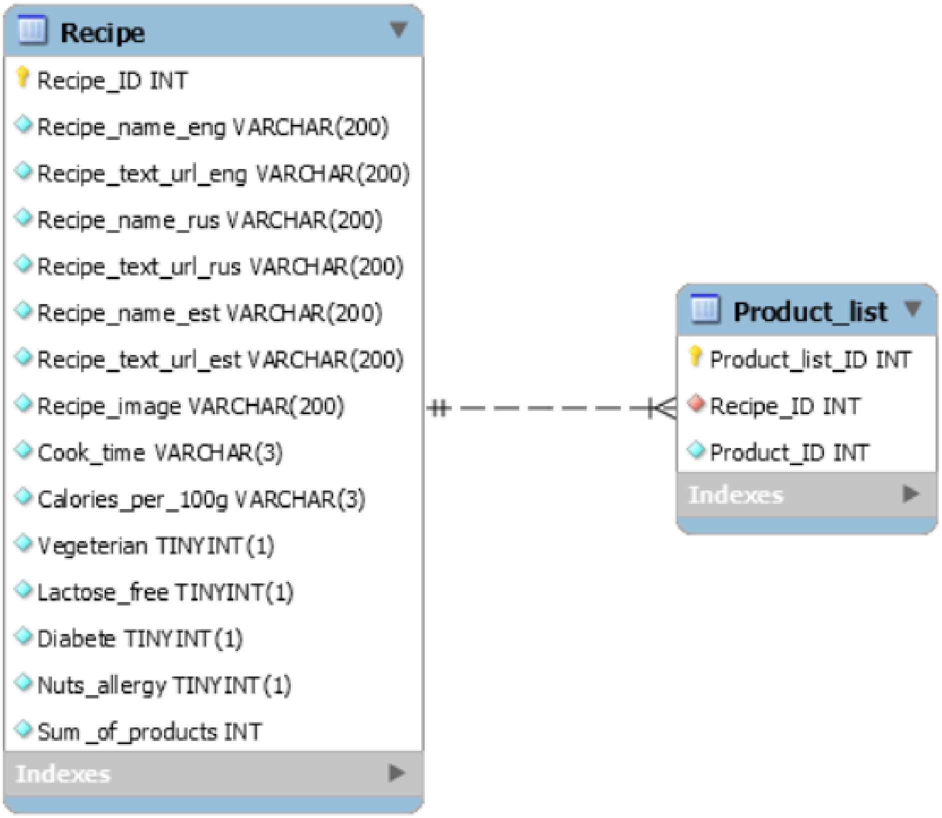

Recipe
Recipe_ID INT
Recipe_name_eng VARCHAR(200)
Recipe_text_url_eng VARCHAR(200)
Recipe_name_rus VARCHAR(200)
Recipe_text_url_rus VARCHAR(200)
Recipe_name_est VARCHAR(200)
Recipe_text_url_est VARCHAR(200)
Recipe_image VARCHAR(200)
Cook_time VARCHAR(3)
Calories_per_100g VARCHAR(3)
Vegeterian TINYINT(1)
Lactose_free TINYINT(1)
Diabete TINYINT(1)
Nuts_allergy TINYINT(1)
Sum _of_products INT
Indexes
Product_list
Product_list_ID INT
Recipe_ID INT
Product_ID INT
Indexes

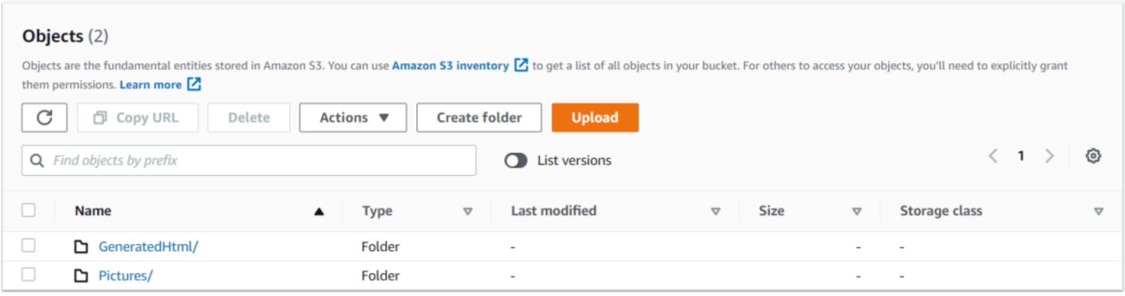
Figure 8. *Database ER Diagram (Source: author)*

4.3 Aws

Amazon Web Service offers 12 months of free use of the database and file hosting, which fully meets the author requirements.

4.3.1 S3

On the file hosting, author stores recipe images and generated files (**Figure 9**). For the system to work properly, each recipe in the image folder has its own folder (**Figure 10**), which, in turn, stores the images of this recipe (**Figure 10**). The folder with the generated HTML files also contains a separate folder for each recipe (**Figure 11**), in which the HTML files are stored (**Figure 11**).

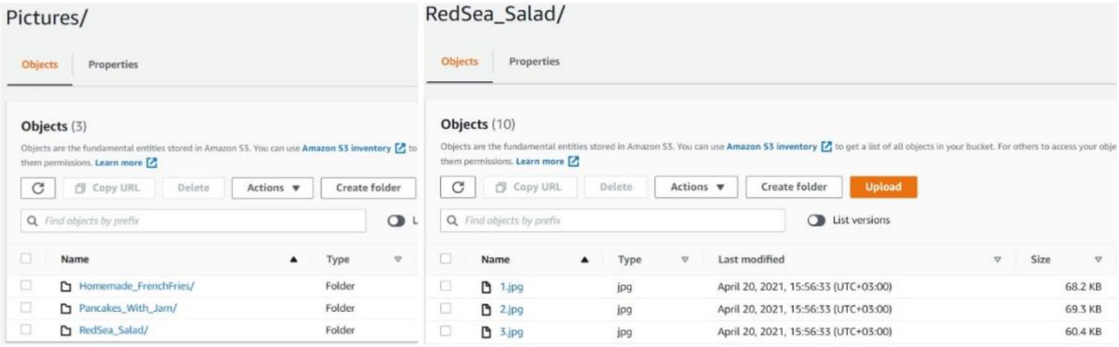


Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name ▲	Type	Last modified	Size	Storage class
□	GeneratedHtml/	Folder	-	-	-
□	Pictures/	Folder	-	-	-

Figure 9. Main folders *(Source: author)*



Pictures/

Objects (3)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to them permissions. Learn more

□	Name ▲	Type
□	Homemade_FrenchFries/	Folder
□	Pancakes_With_Jam/	Folder
□	RedSea_Salad/	Folder

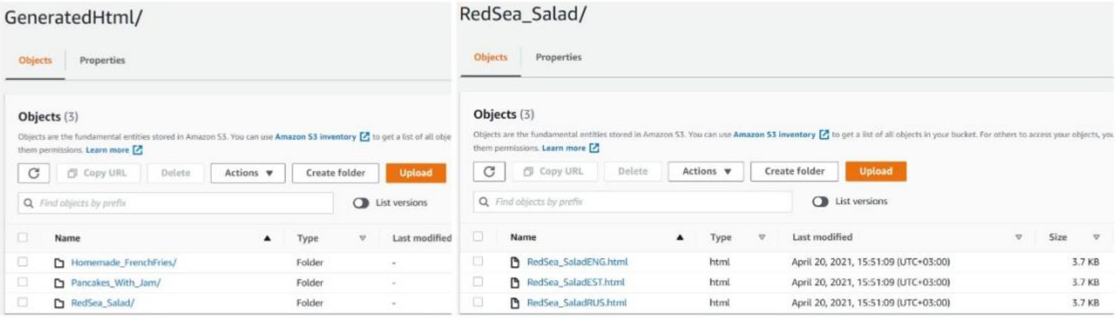
RedSea_Salad/

Objects (10)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your obje them permissions. Learn more

□	Name ▲	Type	Last modified	Size
□	1.jpg	jpg	April 20, 2021, 15:56:33 (UTC+03:00)	68.2 KB
□	2.jpg	jpg	April 20, 2021, 15:56:33 (UTC+03:00)	69.3 KB
□	3.jpg	jpg	April 20, 2021, 15:56:33 (UTC+03:00)	60.4 KB

Figure 10. *folder relationships in Pictures (Source: author)*



GeneratedHtml/

Objects (3)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all obje them permissions. Learn more

□	Name ▲	Type	Last modified
□	Homemade_FrenchFries/	Folder	-
□	Pancakes_With_Jam/	Folder	-
□	RedSea_Salad/	Folder	-

RedSea_Salad/

Objects (3)

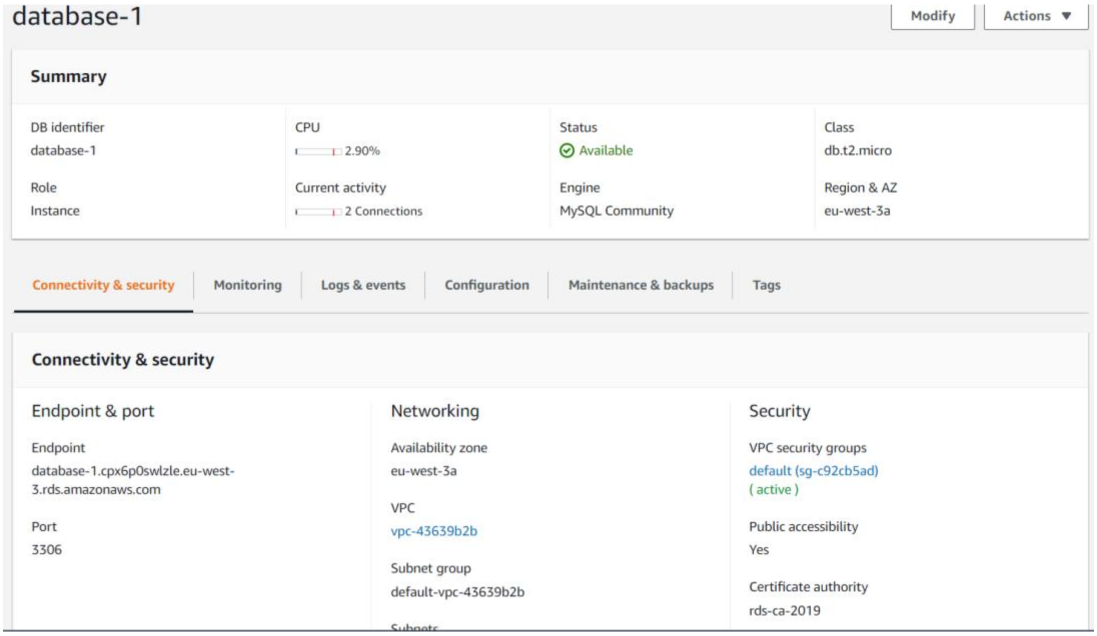
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you them permissions. Learn more

□	Name ▲	Type	Last modified	Size
□	RedSea_SaladENG.html	html	April 20, 2021, 15:51:09 (UTC+03:00)	3.7 KB
□	RedSea_SaladEST.html	html	April 20, 2021, 15:51:09 (UTC+03:00)	3.7 KB
□	RedSea_SaladRUS.html	html	April 20, 2021, 15:51:09 (UTC+03:00)	3.7 KB

Figure 11. *folder relationships in GeneratedHtml (Source: author)*

4.3.2 RDS

The author keeps his MySQL database in the Amazon RDS (**Figure 12**), the database is in the network and always available.



database-1

Summary

DB identifier
database-1

CPU
2.90%

Status
Available

Class
db.t2.micro

Role
Instance

Current activity
2 Connections

Engine
MySQL Community

Region & AZ
eu-west-3a

Connectivity & security

Endpoint & port

Endpoint
database-1.cpx6p0swlzle.eu-west-3.rds.amazonaws.com

Port
3306

Networking

Availability zone
eu-west-3a

VPC
vpc-43639b2b

Subnet group
default-vpc-43639b2b

Security

VPC security groups
default (sg-c92cb5ad)
(active)

Public accessibility
Yes

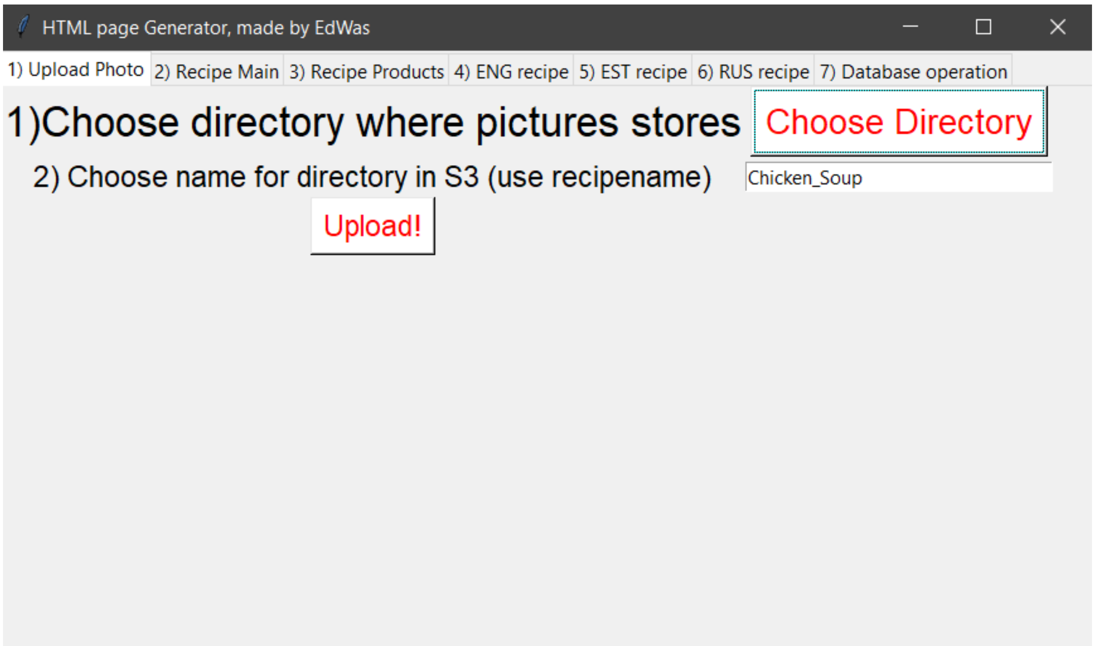
Certificate authority
rds-ca-2019

Figure 12. *Database control page (Source: author)*

4.4 Program for adding recipes

The author has created a program for adding recipes using the Python programming language and the Tkinter graphic package. The author used python because he has experience with it, also python is well suited for creating desktop programs. The author did not tried to make a beautiful or user-friendly design, since this program is not for users, it is using by a moderator or administrator. The main task was made the program functional and simple.

In the image (**Figure 13**) you can see how the upload page of the program looks. First of all, the moderator or administrator clicks the "Choose Directory" button, a dialog box appears where you need to select a folder on the local device in which the images of the created recipe are stored, after that you need to specify the name for the directory in which the recipe images will be stored on the file storage. The last step is to click the "Upload!" button, after clicking the button, the application will create a directory on the file server and send there all the images that were stored in the folder selected on the local device.

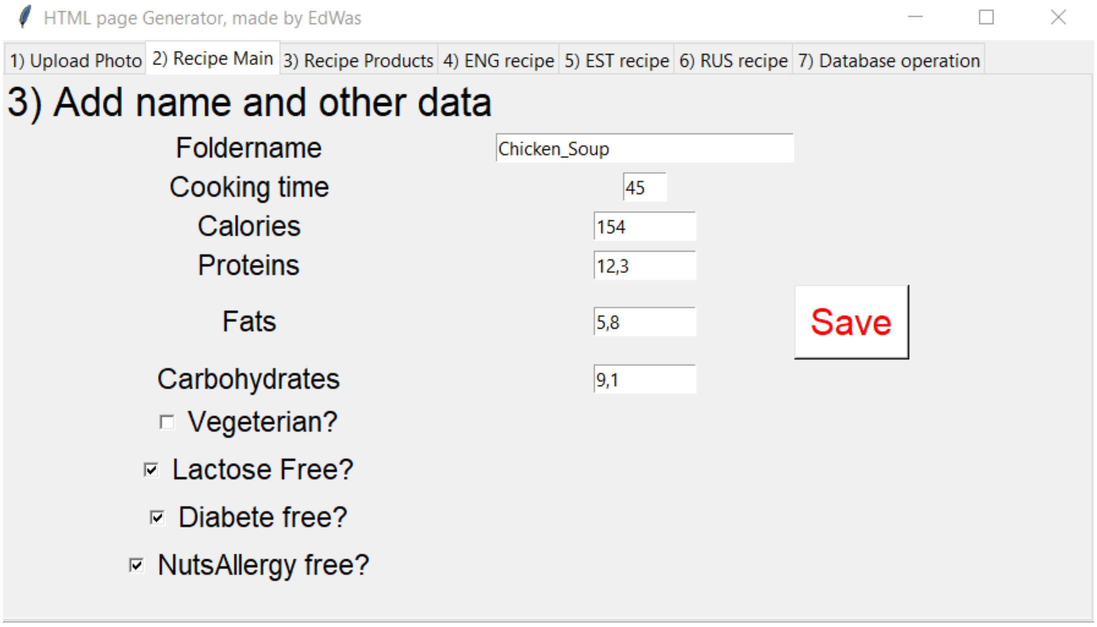


1)Choose directory where pictures stores

2) Choose name for directory in S3 (use recipename) Chicken_Soup

Figure 13. *Upload Photo page (Source: author)*

The image (**Figure 14**) shows a page where the moderator or administrator adds basic information about the recipe, "Foldername" is the name of the directory that will be created in the file server to store the generated html files, in order to avoid confusion in the future, it is advisable to use the same name for the image directory and the html file directory. "Calories", "Fats", "Proteins", "Carbohydrates" are needed to display this data in the generated html file. "Calories" and the checkboxes will be sent to the database.



3) Add name and other data

Foldername	Chicken_Soup
Cooking time	45
Calories	154
Proteins	12,3
Fats	5,8
Carbohydrates	9,1

□ Vegeterian?

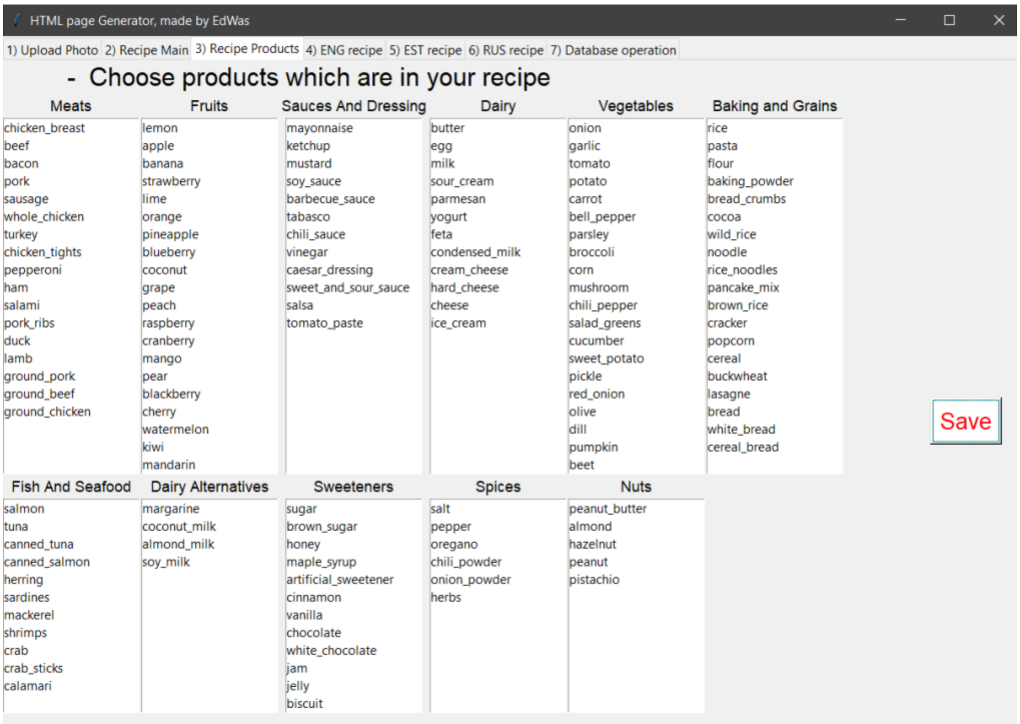
☑ Lactose Free?

☑ Diabete free?

☑ NutsAllergy free?

Figure 14. *Upload Photo page (Source: author)*

The image (**Figure 15**) shows a page where the moderator or administrator selects products that include a recipe, for convenience, the products are divided into categories, to save the selected products, you need to press the "Save" button. The selected products are then sent to the database, to the table "Product_List".



- Choose products which are in your recipe

Meats

chicken_breast
beef
bacon
pork
sausage
whole_chicken
turkey
chicken_tights
pepperoni
ham
salami
pork_ribs
duck
lamb
ground_pork
ground_beef
ground_chicken

Fruits

lemon
apple
banana
strawberry
lime
orange
pineapple
blueberry
coconut
grape
peach
raspberry
cranberry
mango
pear
blackberry
cherry
watermelon
kiwi
mandarin

Sauces And Dressing

mayonnaise
ketchup
mustard
soy_sauce
barbecue_sauce
tabasco
chili_sauce
vinegar
caesar_dressing
sweet_and_sour_sauce
salsa
tomato_paste

Dairy

butter
egg
milk
sour_cream
parmesan
yogurt
feta
condensed_milk
cream_cheese
hard_cheese
cheese
ice_cream

Vegetables

onion
garlic
tomato
potato
carrot
bell_pepper
parsley
broccoli
corn
mushroom
chili_pepper
salad_greens
cucumber
sweet_potato
pickle
red_onion
olive
dill
pumpkin
beet

Baking and Grains

rice
pasta
flour
baking_powder
bread_crumbs
cocoa
wild_rice
noodle
rice_noodles
pancake_mix
brown_rice
cracker
popcorn
cereal
buckwheat
lasagne
bread
white_bread
cereal_bread

Fish And Seafood

salmon
tuna
canned_tuna
canned_salmon
herring
sardines
mackerel
shrimps
crab
crab_sticks
calamari

Dairy Alternatives

margarine
coconut_milk
almond_milk
soy_milk

Sweeteners

sugar
brown_sugar
honey
maple_syrup
artificial_sweetener
cinnamon
vanilla
chocolate
white_chocolate
jam
jelly
biscuit

Spices

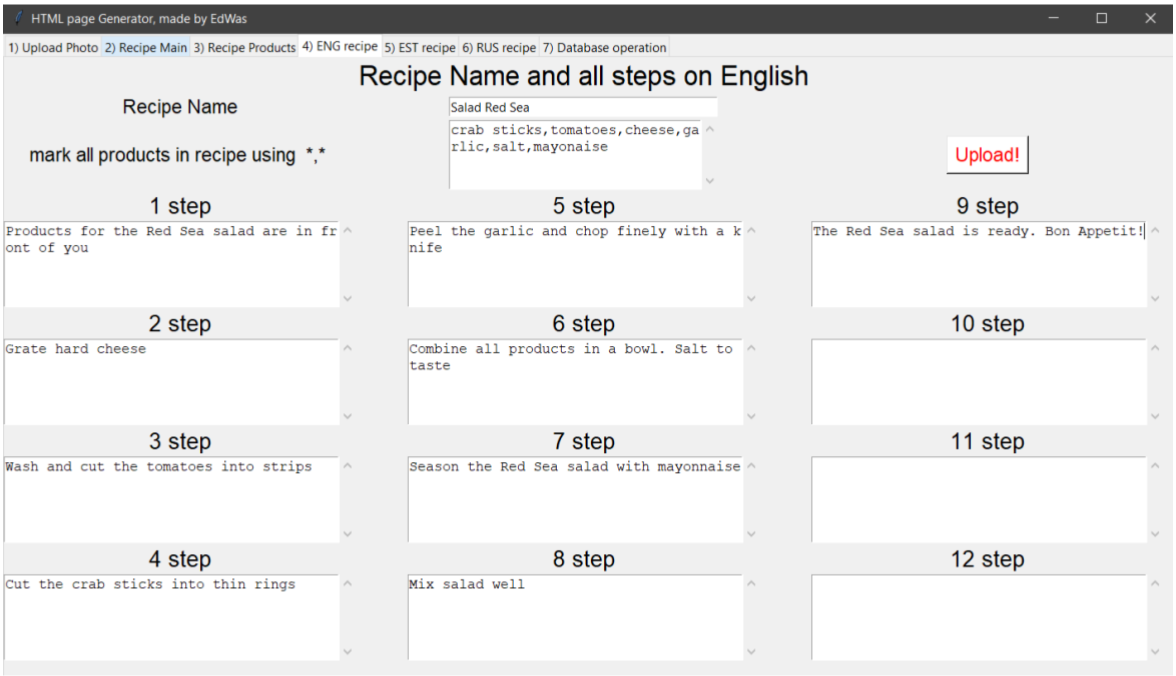
salt
pepper
oregano
chili_powder
onion_powder
herbs

Nuts

peanut_butter
almond
hazelnut
peanut
pistachio

Figure 15. *Products in recipe page (Source: author)*

The image (**Figure 16**) shows a page where the moderator or administrator must add information about the recipe in English. All information specified on this page will be part of the generated html file. You need to specify the name of the recipe, products using a comma and describe each step of the recipe. Once the data is filled in, you need to press the "Upload" button and then the data will be saved to the local device. The minimum number of steps is 1, the maximum is 12. If, for example, there are 8 steps in the recipe, then the rest of the fields should be left blank. The maximum number of steps is 12, because the author's vision is simple recipes, and simple recipes should not require many steps. It is also important to take into account the moment that for each step there should be its own image (images that were uploaded at the beginning), this was done because the recipes should be standardized, and did not have a different design style.



Recipe Name and all steps on English

Recipe Name: Salad Red Sea

mark all products in recipe using *,*: crab sticks,tomatoes,cheese,garlic,salt,mayonaise

1 step: Products for the Red Sea salad are in front of you

2 step: Grate hard cheese

3 step: Wash and cut the tomatoes into strips

4 step: Cut the crab sticks into thin rings

5 step: Peel the garlic and chop finely with a knife

6 step: Combine all products in a bowl. Salt to taste

7 step: Season the Red Sea salad with mayonnaise

8 step: Mix salad well

9 step: The Red Sea salad is ready. Bon Appetit!

10 step:

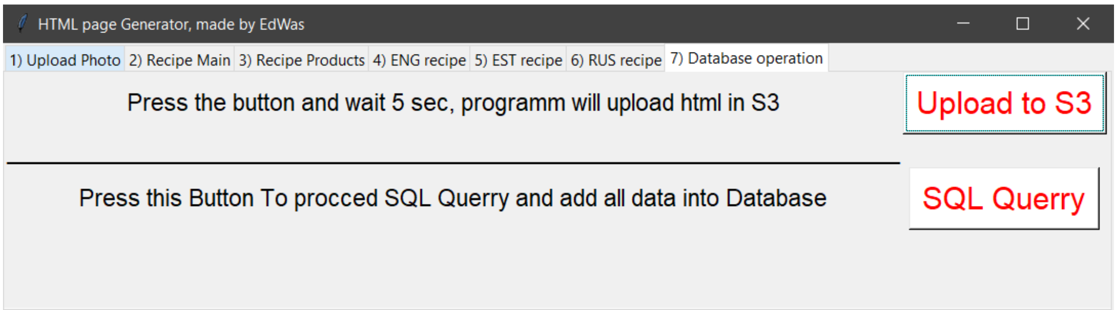
11 step:

12 step:

Figure 16. *Recipe description in english page (Source: author)*

Pages "EST recipe" and "RUS recipe" look exactly the same as "ENG recipe" (**Figure 16**), with only one difference in "EST recipe" the information must be filled in in Estonian, and in "RUS recipe", accordingly, in Russian. Therefore, the author will skip the description of these pages.

The image (**Figure 17**) shows a page where the moderator or administrator must press the "Upload to S3" button and wait about 5 seconds, this is necessary because after pressing this button, the generated html files will be uploaded to the file server (if you ignore this and press the next button immediately (SQL Querry), then will be error in the program). When 5 seconds have passed, you can press the " SQL Querry " button and then the program will add all the data to the database. After that, the recipe is ready and stored in the database, now users can see it in the application if they select the products included in the recipe.



Press the button and wait 5 sec, programm will upload html in S3

Press this Button To procced SQL Querry and add all data into Database

Figure 17. *Upload and Database operation page (Source: author)*

4.5 Android Application

The application was created in the Android studio development environment. The author had a choice between the Java programming language and the Kotlin programming language. Many sources say that Kotlin is simpler than Java (Kotlin vs Java: Which is Better for Android App Development), but the author still decided to use Java, because he had some experience before and also because Java has a larger community than Kotlin.

The application is focused on the latest version of android, at the time of writing the diploma it is Android11 (api level 30), the minimum supported version of Android is Lollipop (api level 21). For the application to function correctly, the smartphone must have access to the Internet, sometimes it may be necessary to give permission to use the Internet in the smartphone settings.

4.5.1 Choose Products Screen

When the application starts, the user sees a product screen (**Figure 18**) where the products are divided into categories, the user can vertically scroll through this part of the screen to access the rest of the products. By clicking on the button with the name of the product, it will change color to green, this indicates that this product is now selected, to cancel the selection, you need to press the button again, then it will return to its previous state and the product will no longer be selected. At the top, the user can see how many products he has chosen. Also on the screen (**Figure 18**) at the bottom there is a navigation menu between screens, with the help of it the user can switch to a screen with recipes or to a screen with settings. The author has implemented about 140 buttons, each button is a product. For the demo version, the author decided not to use all the buttons. When user selects products that he needs, he presses the button at the top (save products) and can go to the next screen (Explore) using the navigation menu.

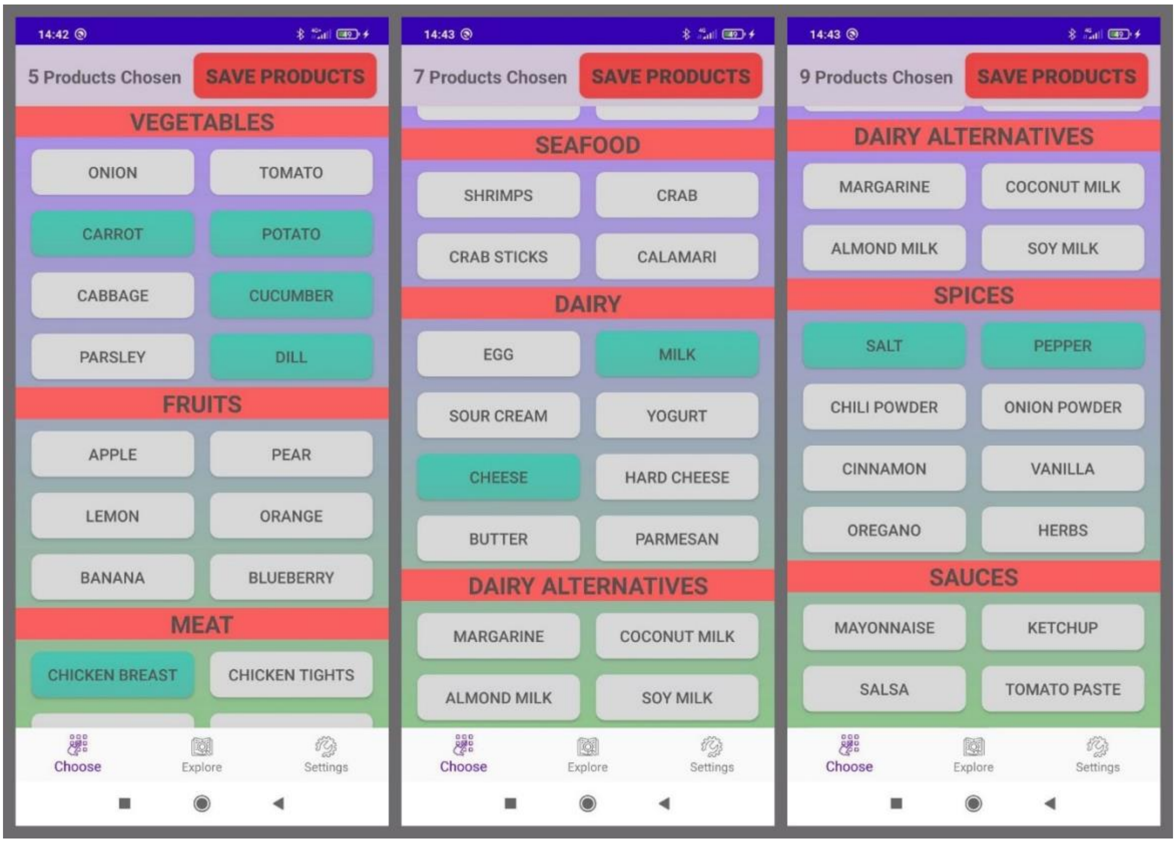
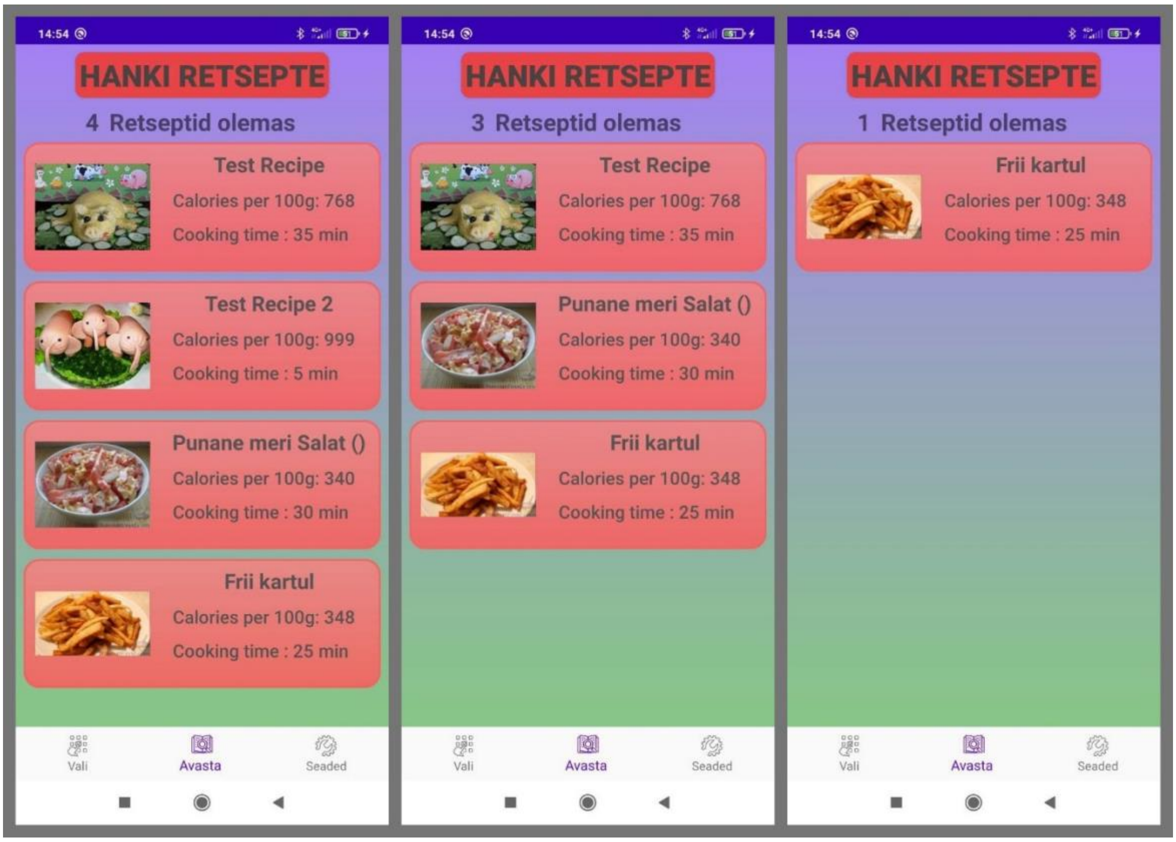


Figure 18. *Choose product screen (Source: author)*

4.5.2 View Available Recipes Screen

If the user has selected and saved products, he can click on the button at the top (**Figure 19**), after which recipes will appear that he can cook from the products he has selected. The user sees the name of the recipe, its image, the number of calories per 100 grams and the cooking time. The language of the recipe and the names depend on the language in the application, for example, if the user has the application in Estonian, then the recipe will be in Estonian, if in Russian, then the recipe will be in Russian. The user also sees how many recipes he can cook. When the user has received a list of recipes, he can select the recipe he liked, click on it, and then he will be taken to the next screen (Recipe Screen). The author did not add many recipes as it takes a very long time.



HANKI RETSEPTE

4 Retseptid olemas

Test Recipe
Calories per 100g: 768
Cooking time : 35 min

Test Recipe 2
Calories per 100g: 999
Cooking time : 5 min

Punane meri Salat ()
Calories per 100g: 340
Cooking time : 30 min

Frii kartul
Calories per 100g: 348
Cooking time : 25 min

HANKI RETSEPTE

3 Retseptid olemas

Test Recipe
Calories per 100g: 768
Cooking time : 35 min

Punane meri Salat ()
Calories per 100g: 340
Cooking time : 30 min

Frii kartul
Calories per 100g: 348
Cooking time : 25 min

HANKI RETSEPTE

1 Retseptid olemas

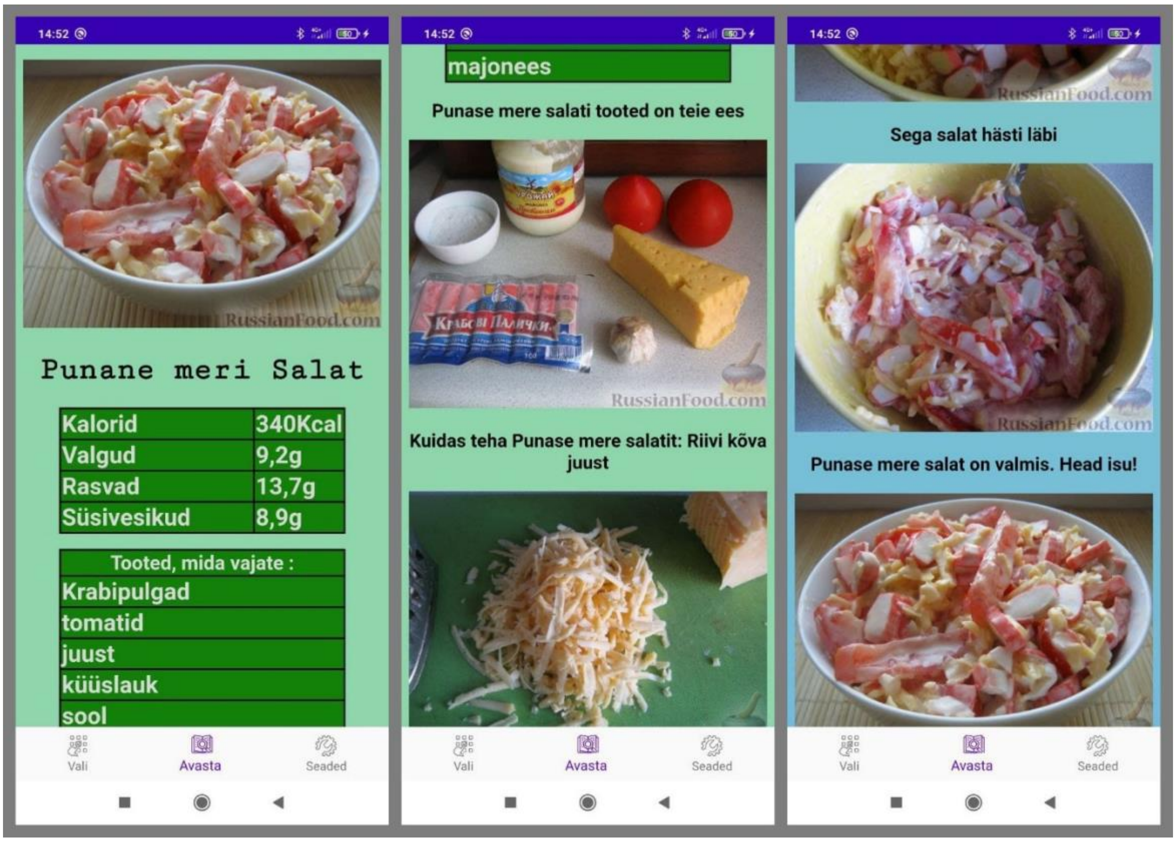
Frii kartul
Calories per 100g: 348
Cooking time : 25 min

Figure 19. *Available recipes (Source: author)*

4.5.3 Recipe Screen

After the user has clicked on the recipe that suits him, a new screen is loaded (**Figure 20**) in which the recipe, images, a list of products, and other elements are displayed. The recipe language depends on which language the user uses in the application. User sees the generated html file that is stored on the Amazon S3. In order to exit the recipe, user can use the navigation menu of the application, or use the back button on his smartphone.

To test the functionality of the system, the author used images and other information taken from the site (RussianFood).



14:52

Punane meri Salat

Kalorid	340Kcal
Valgud	9,2g
Rasvad	13,7g
Süsivesikud	8,9g

Tooted, mida vajate :
Krabipulgad
tomatid
juust
küüslauk
sool

majonees

Punase mere salati tooted on teie ees

Kuidas teha Punase mere salatit: Riivi kõva juust

Sega salat hästi läbi

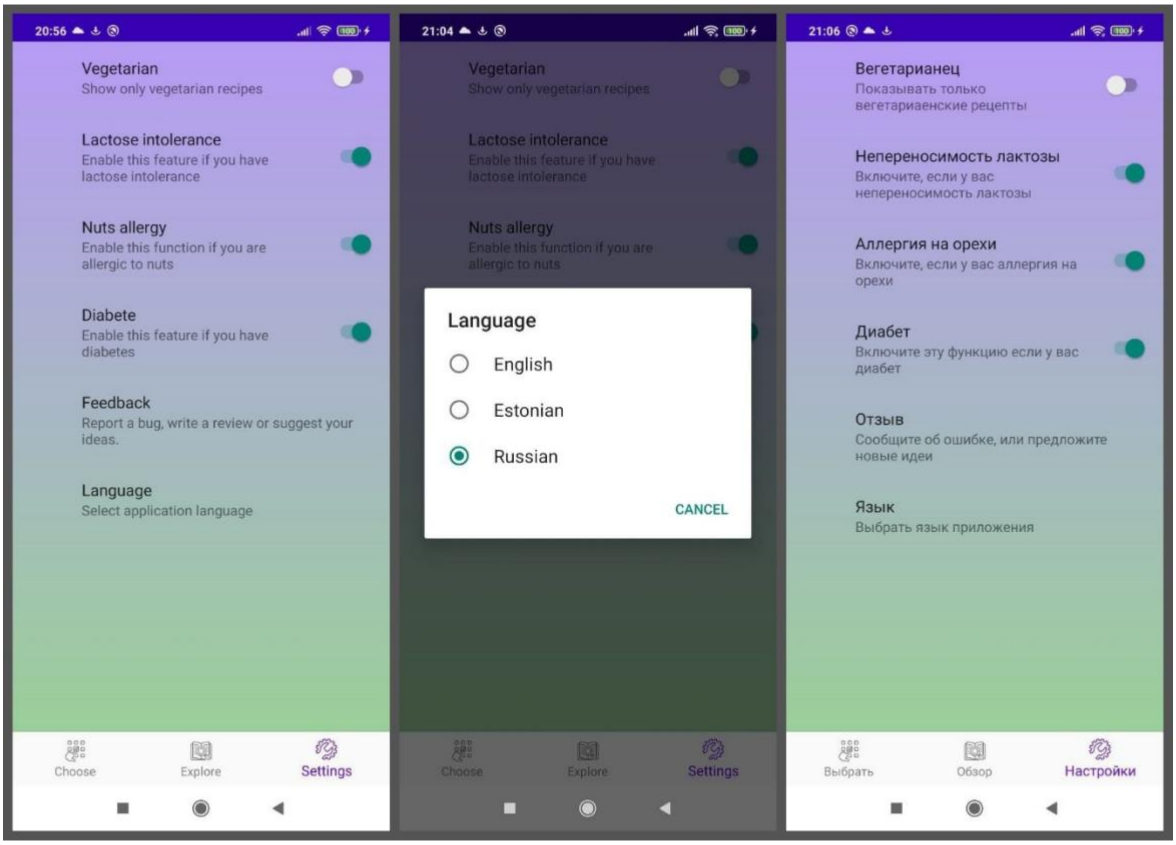
Punase mere salat on valmis. Head isu!

Figure 20. *Recipe screen (Source: author)*

4.5.4 Settings Screen

The user enters this screen (**Figure 21**) by clicking on the settings button in the navigation menu. User can choose his preferences or indicate allergies, write to the author to report a bug. Also on this screen the user selects the preferred language (Estonian, Russian, English). All settings specified by the user are saved even if the application is restarted.

By the time of the defense of the thesis, the application does not in any way use the user's allergies / taste preferences. The author did not find the correct implementation of this function, so the author transfers this part to future development.



20:56

Vegetarian
Show only vegetarian recipes

Lactose intolerance
Enable this feature if you have lactose intolerance

Nuts allergy
Enable this function if you are allergic to nuts

Diabete
Enable this feature if you have diabetes

Feedback
Report a bug, write a review or suggest your ideas.

Language
Select application language

21:04

Language

○ English

○ Estonian

● Russian

CANCEL

21:06

Вегетарианец
Показывать только вегетариаенские рецепты

Непереносимость лактозы
Включите, если у вас непереносимость лактозы

Аллергия на орехи
Включите, если у вас аллергия на орехи

Диабет
Включите эту функцию если у вас диабет

Отзыв
Сообщите об ошибке, или предложите новые идеи

Язык
Выбрать язык приложения

Figure 21. *Settings screen and language switching process (Source: author)*

4.6 API

To connect the Android application and the database, you need to write an API. It is possible to connect the application directly to the database, but this is very insecure, since an attacker can get the data from the database from the code. Therefore, the author wrote an API using the PHP programming language for this.

API takes products ID’s from the application, each product button in the application (product selection screen) has a unique identifier that matches the ID’s in the database. For example, a potato has an ID of 16, an Android application and application for adding recipes know that the ID of a potato is 16.

After the API has received data from the Android application, it connects to the database and sends a request. In response, the API receives all recipes that contain at least one of the selected products.

After that, the API, using a loop and comparison of values, saves to the list only those recipes that contain all the products selected by the user. At the end, the API converts this list into a JSON file.

All PHP files are stored on free hosting, thanks to which anyone can use the functionality of the application. This system has 4 files required for proper functioning. The author will briefly describe each of them.

- **db_config.php –** stores data for connecting to the database
- **inputproductsENG.php –** Performs the above actions (api description), returns data in English.
- **inputproductsEST.php –** Performs the above actions (api description), returns data in Estonian.

  inputproductsRUS.php – Performs the above actions (api description), returns data in Russian.

5. FINAL RESULT

In this chapter, the author will list the functionality implemented in the system and the functionality that the author was unable to implement.

5.1 Implemented functionality

For the full functioning of the system, the author had to complete 5 development steps, namely:

- Creation and construction of a database for storing recipes and other information
- Implementing the storage of images and other recipe-related information on cloud storage
- Implementation of the interface for communication with the database (API)
- Creation of a program for generating recipes, sending them to the cloud storage and writing all the information to the database
- Development of an Android application for viewing recipes

The database is fully operational, it is built using MySql and is stored on Amazon RDS. Thanks to this, the database is always available from anywhere in the world (if the Internet is available).

Images and HTML recipe files are stored on the Amazon C3 file storage, thanks to which recipes and images are always available to users if the Internet is available.

The interface (API) works as expected, it always connects to the database stably, transfers data to it and receives a response from it, discards recipes that are inappropriate in parameters and converts the response to JSON format.

The programm for recipe generating works as expected. It sends images to the file storage, generates recipes according to the specified parameters, sends the generated recipes to the file storage and writes all data to the database.

Most of the functions declared by the author work in the current version of the Android application, however, some functionality is limited or not implemented, the author will mention this in the chapter (**5.2 Unimplemented functionality**).

At the time of the defense of the diploma, the Android application allows user to:

- Move between Screens using the navigation menu
- Select products by clicking on the buttons
- See how many products the user has selected
- Save selected products by clicking on the button
- Get a list of recipes by clicking on a button
- Open available recipes for detailed view
- Change application language to Russian, English or Estonian

5.2 Unimplemented functionality

During the development of the Android application, the author faced some difficulties, namely, the author could not correctly implement the accounting of allergies / taste preferences (**Figure 21**). Therefore, at the time of the defense of the thesis, this function does not work, but in the future the author plans to fix this problem and use all the functionality of the application.

In total, the system supports about 140 products, but the author did not implement all the products in the form of buttons in the Android application, since the author plans to change the concept of choosing products in the Android application. At the moment, there are about 50 buttons in the application, which, according to the author, is enough to get started.

6. FUTURE DEVELOPMENT

In this chapter, the author will describe features and changes that may be added in the future. The author plans to continue developing and supporting the application after defending his thesis, so he considers this chapter to be very important.

6.1 Android Application Development Environment

During the development of an Android application, the author faced many problems and limitations, such as:

- **Lack of standard solutions** – Some functionality, for example, changing the language within the entire application can be implemented in several ways. Sometimes choosing “wrong” methods, in the future development, you can face many problems.
- **Many methods/functions are deprecated** – it means that Android is being supported and improved, which is good. But sometimes the author was faced with the fact that he could not find an answer to how to use the new functionality.
- **Java language complexity for the author** – Thanks to the course at Narva College, the author had a basic understanding of the language. However, Java in conjunction with Android studio was difficult for the author to understand.
- **Lack of documentation** – In fact, there is a lot of documentation, android has an official site where there is a large amount of documentation, but sometimes it was difficult for the author to understand the documentation and after reading there were questions. Searching for answers on other resources often did not help, often there were deprecated solutions.

For this reason, author is considering two options. The first option is to continue studying and working in Android Studio with Java. The second option is to try to switch to Web-application, for example React.Js and Javascript.

6.2 Create a Separate Screen For Each Product Category

Creating a separate screen for each category will make the main screen cleaner and easier for the user to navigate between products **(Figure 22)**.

The author tried to implement this, but ran into a problem, namely, the inability to save the selected products after moving to the screen with a different category of products. For example - the user opens the category "vegetables", selects potatoes and cucumbers in it, after that the user returns to the screen with the categories of products, the products selected by the user are not saved. The author knows the cause of the problem, but does not have enough experience to solve this problem.

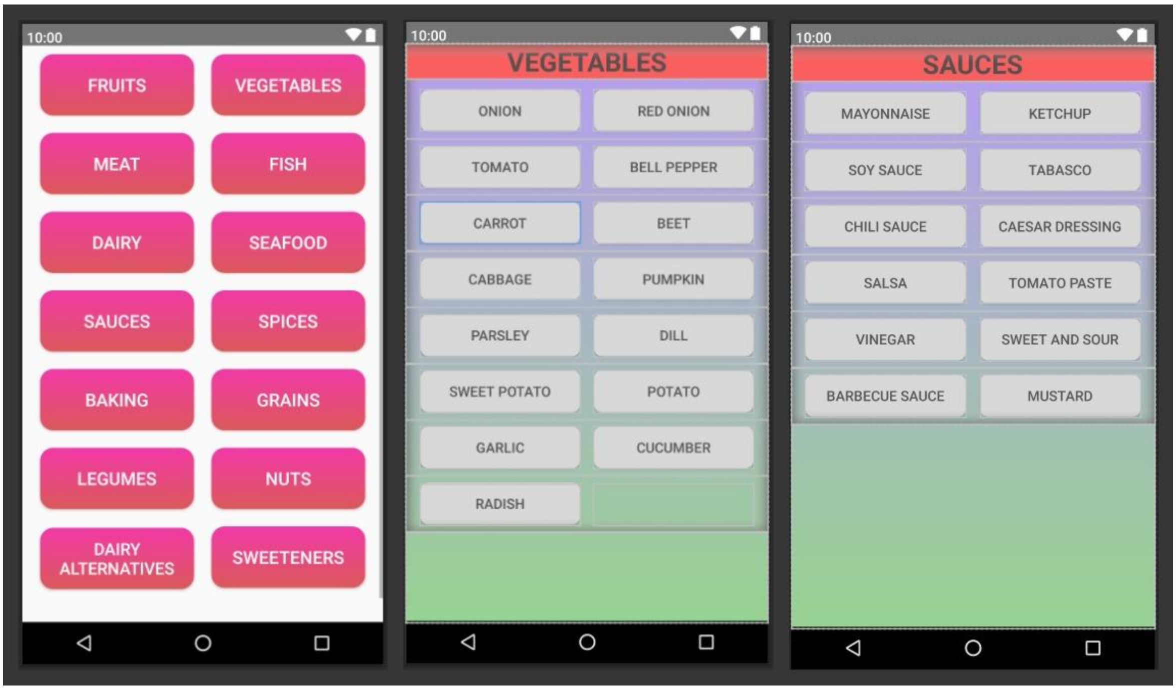


Figure 22. *Realization example (Source: author)*

6.3 Expanding the functionality of the application

A lot of functions can be added to the application in the future, the author will list some of them.

1. **Rating System For Recipes -** According to the author's opinion, such a system is very important in this application, since with its help the moderator/administrator of the system can understand which recipes users dislike and which recipes users like. In turn, this implementation improves the quality of the application and the system as a whole.
2. **Save Recipe To Favorites -** According to the author's opinion, many users of the application would like to be able to save recipes that they like to their favorites list.
3. **Make Application More Socially Oriented -** By this point, the author means the development of a function so that the user can share recipes with other users or his friends, relatives, acquaintances.

4. **Give Users Ability To Suggest Their Own Recipes -** The author thinks this feature is very useful and important. First of all, some users may want to share their recipes, for the author this in turn gives more recipes in the application.

CONCLUSION

The author has achieved the result, most of the declared functions work. In order to release this application to the play market, the author must make some changes and fixes, for example:

- change the design of the application
- complete the declared functionality
- fix small bugs
- change the database
- implement all product buttons.

However, the application and the rest of the system are usable, and the author believes that he has achieved a good result. The author plans to continue developing this system and the Android application in particular.

The author believes that this work gave him a large amount of experience in developing applications for Android, building databases, communicating with the database using API and programming in python.

CONCLUSION IN ESTONIAN

"ANDROID RAKENDUS TOIDUVALMISTAMISE RETSEPTIDE OTSIMISEKS KOOSTISOSADE LOENDI JÄRGI"

Autor on saavutanud tulemusi, kus enamus rakenduse välja kuulutatud funktsioonidest korralikult töötavad. Enne rakenduse Play Marketisse üles laadimiseks autor peab tegema mõned muudatused ja parandused, näiteks:

- muuta rakenduse kujundust,
- täita puuduva funktsionaali,
- parandada väikseid vigu,
- muuta andmebaasi,

- lisada kõike nupud toodetega.

Kuid rakendus ja ülejäänud süsteem saab kasutada ning autori arvates on saavutatud head tulemused. Autor kavatseb jätkata selle süsteemi ja eriti Androidi rakenduse arendamist.

Autor arvab, et see töö andis talle palju kogemusi Android-rakenduste väljatöötamisel, andmebaaside loomisel ja päringute esitamises rakendusliideste (API) ning Python'i programmeerimiskeele abil.

REFERENCES

Oracle Corporation 2021. *MySQL Reference Manual: What is MySQL*. Available at https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html, accessed in April 7, 2021.

Oracle Corporation 2021. MySQL Workbench Manual: *What Is New in MySQL Workbench*. Available at https://dev.mysql.com/doc/workbench/en/wb-intro.html, accessed in April 7, 2021.

Richard Becker (2021) *Structured Query Language (SQL).* Available at https://www.techopedia.com/definition/1245/structured-query-language-sql, accessed in April 7, 2021

Android Studio 2021. Meet Android Studio. Available at https://developer.android.com/studio/intro, accessed in April 7, 2021

Guru99 2021. *What is PHP*. Available at https://www.guru99.com/what-is-php.html, accessed in April 16, 2021

Oracle Corporation 2021. *What is Java technology and why do I need it*. Available at https://java.com/en/download/help/whatis_java.html, accessed in April 16, 2021

Amazon 2021. *Getting started with Amazon S3*. Available at https://aws.amazon.com/s3/getting-started/?nc1=h_ls, accessed in April 16, 2021

Wikipedia 2021. *HTML*. Available at https://en.wikipedia.org/wiki/HTML, accessed in April 16, 2021

Python 2021. What is Python? Executive Summary. Available at https://www.python.org/doc/essays/blurb/, accessed in April 16, 2021

Eileen Roche (2000). *Explaining XML*. Available at https://hbr.org/2000/07/explaining-xml, accessed in April 16, 2021

Amazon 2021. What is Amazon Relational Database Service. Available at https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Welcome.html, accessed in April 16, 2021

Python 2021. TkInter. Available at https://wiki.python.org/moin/TkInter, accessed in April 16, 2021

Navdeep Singh Gill (Aug 24, 2020). Kotlin vs Java: Which is Better for Android App Development. Available at https://www.xenonstack.com/blog/kotlin-andriod/, accessed in April 17, 2021

RussianFood. Recipes for functionality testing. Available at https://www.russianfood.com/, accessed in April 25, 2021

APPENDICES

ZIP file with the system contains the following items:

- Folder with Android Studio project
- Folder with a program for generating recipes and sending all data to the database
- Folder with API
- Folder with database creation script