

Information System for Calorie Calculation

Valentyna Pleskach¹, Yevheniia Vakulenko¹, Anatoliy Serdyuk²

¹ Taras Shevchenko National University of Kyiv, Bohdan Havrylyshyn str. 24, Kyiv, 04116, Ukraine

² Warsaw University of Technology, Nowowiejska str. 24, Warsaw, 00-665, Poland

Abstract

The theoretical basis of building software systems for calculating calories was investigated in the work, architectural decisions regarding the creation of software systems for calculating calories were analyzed, and a software system for calculating calories was designed, implemented and tested. As a result of a comparative analysis of software systems, a monolithic architecture was chosen, on the basis of which a software system for counting calories with a weekly menu was implemented using the Microsoft Visual Studio 2022 development environment; .Net Framework version 7.0.0; ADO.NET technologies; Windows Presentation Foundation (WPF); C# programming language; XAML markup languages; Microsoft SQL Server 2019 database in order to improve the efficiency and convenience of providing services in the field of nutrition and health.

Keywords

Information system, meal plan, monolithic architecture, calorie counting, diet optimization

1. Introduction

Over the past half-century, obesity has become a major public health problem. According to experts from the World Health Organization (WHO), the health of citizens depends on the health care system in the country by 8-12%, by 18-20% on a person's genetic predisposition, and by 68-74% on lifestyle, one of the most important components of which is nutrition [1]. According to the classification by the WHO, a person is considered obese if their body mass index (BMI) is more than 30 kg/m². Therefore, counting a person's calories is a key element in the implementation of a healthy diet and lifestyle.

Research related to the development of applied calorie counting information systems is important for a number of reasons, including improved quality of life and health, personalised recommendations, increased nutrition awareness, accessibility, feedback, social impact and economic benefits. Proper nutrition is a key factor in maintaining health and preventing many diseases, such as obesity, diabetes, and cardiovascular disease. Calorie counting systems help users monitor their food intake and make healthier choices. Research is enabling the development of systems that take into account individual user characteristics such as age, gender, physical activity, dietary preferences and medical conditions, which provides more accurate and effective recommendations to help users achieve their personal goals. Information systems can provide users with knowledge about the composition of foods, their calorie content, macro- and micronutrient content, which really helps people make more informed food choices and understand the impact of different foods on their health. Modern technology allows us to create user-friendly mobile apps and web platforms that make calorie counting easy and accessible to a wide range of users. Calorie counting systems can provide users with the ability to track their progress, receive recommendations and adjust their diet based on changes in weight, activity level and other factors, which helps to keep them motivated and achieve better results. The new systems contribute to the

14th International Scientific and Practical Conference from Programming UkrPROG'2024, May 14-15, 2024, Kyiv, Ukraine

✉ v.pleskach64@gmail.com (V. Pleskach); zhenia2607.cool@gmail.com (E. Vakulenko); Anatolii.Serdiuk@pw.edu.pl (A.Serdiuk)

📄 0000-0003-0552-0972 (V. Pleskach); 0009-0002-4256-4464 (E. Vakulenko); 0000-0002-0787-3889 (A. Serdiuk)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

development of new methods and algorithms for more accurate calorie counting, diet analysis and recommendations. This helps to create innovative solutions based on modern scientific achievements. Calorie counting information systems can have a significant impact on society by helping to reduce the prevalence of obesity and other diet-related diseases, which will reduce the burden on healthcare systems and improve the overall health of the population. Improved population health can reduce healthcare costs and increase productivity, which has a positive impact on the economy as a whole.

Thus, an urgent issue is the development of universal information technology, which in real time calculates the number of calories of any client's food every day and allows you to optimize the diet. The most pragmatic solution to this problem is to design, develop and test such a software system for calculating the calorie content of food products using the latest technologies, so that it has a friendly intuitive interface, and the client himself can adjust the data, create his own menu for the day, week, and observe the change in his indicators towards improving his own health. There are many studies related to the recognition and identification of food images using, for example, neural networks, computer vision. Specifically, it is a method based on parameter-optimized convolutional neural networks (CNNs) to detect images of traditional dishes using a handheld camera [2].

These approaches are typically associated with deep learning – for example, the DEEPFIC system, which uses deep learning to classify and evaluate foods calorie content from images, which involves pre-processing images and the application of CNN neural networks for accurate food recognition and calorie estimation; mobile apps and web-based systems exploring the creation of calorie counters on mobile platforms such as Android to help users track their daily calorie intake effectively. These systems often combine food databases and nutritional information, which allows users to enter their meals and get calorie intake calculations, and algorithms such as YOLOv5 are used to recognize food and estimate calories.

The method of segmentation and image processing to recognize parts of food is used in the study [3].

When building a software system for calculating calories and analysing nutritional intake based on personal factors, it is important to take into account, in particular, the principles of personalization related to age, weight, height, and gender, because these basic parameters determine the basal metabolic rate (BMR) and daily calorie requirements and physical activity level, which affects the number of calories needed to ensure the body's vital functions depending on the goal (weight loss, weight gain, maintenance of current weight), so the calorie requirement may vary.

It is also a balance of macro- and micro-nutrients of proteins, fats, carbohydrates, because it is important not only to count the total number of calories, but also to ensure the correct balance of macronutrients according to personal needs and to ensure sufficient levels of micronutrients to maintain health, taking into account personal food preferences, allergies, medical restrictions (e.g. diabetes, lactose intolerance, vegetarianism), and taking into account eating habits related to cultural or religious beliefs.

The principle of monitoring and adaptability of the calorie counting information system implies feedback, i.e. the ability of the user to track their progress and receive recommendations based on changes in weight, health, energy levels. The programme should be able to adjust the diet depending on the progress and changes in the user's physical condition.

The app should have an easy-to-use interface that allows for easy entry of data on food consumption and physical activity.

Ensuring the secure storage and processing of personal information of users of such a system helps to create a secure, personalised software system for calorie counting and nutritional analysis that takes into account the unique needs of each person.

Assessing the composition of a product, i.e. the amount of essential macronutrients and micronutrients in foods, is one of the most important principles for designing such programmes. It

is also important to take into account personal characteristics such as age, gender, weight, height, activity level and medical conditions that may affect energy and nutrient requirements. With the advent of new technologies, energy calculation methods have also changed, allowing the use of energy expenditure estimates based on fitness trackers and sensors to more accurately determine calorie requirements.

It is desirable that the recommendations are in line with local traditions and the availability of foods according to seasonal and regional characteristics. For accurate results, the analysis of household habits data, which can include statistical methods or machine learning to find dietary patterns, should take into account a person's lifestyle and daily schedule. Taking into account personal preferences means taking into account taste preferences and restrictions, such as vegetarianism, allergies, etc. This implies the need to determine the presence of allergens in products and provide personalised recommendations for people with certain restrictions or allergies. It is imperative to take into account the needs of families by expanding the functionality of the information system to serve groups where everyone can have their own diet and specific recommendations to enhance the taste [4].

A reporting and analytics system is important for successful dietary adjustments. The use of data visualisation technology, with tools such as graphs, tables, or interactive elements, allows users to present the results of data analysis and calorie assessments in a user-friendly way.

The system should also regularly update product composition and calorie data to ensure that the analysis and assessment remains relevant. Additionally, collaboration with nutrition and health experts such as nutritionists, food technologists, and healthcare professionals is essential to maintain the accuracy and relevance of data analysis and calorie estimates. Guidelines should be consistent with those of relevant authoritative health organisations such as WHO, the American Heart Association and the Academy of Nutrition and Dietetics.

To study this topic, an analysis of existing research was conducted, which shows the relevance of modern scientific efforts to develop accurate and convenient tools for monitoring nutrition and physical activity, improve health data processing methods, integrate with other medical and fitness equipment, and create personalised recommendations to support a healthy lifestyle. Many studies focus on the recognition and identification of food images using neural networks [5] and computer vision. For example, some of them are based on convolutional neural networks that detect traditional dishes using portable cameras, which, according to the authors of one of the publications in the Multidisciplinary Digital Publishing Institute, provides high accuracy and can significantly simplify existing manual calorie estimation procedures by turning them into an automated real-time process.

Other studies use deep learning to classify food products and estimate their calorie content from images, such as MobileNetV3 and EfficientDet, which, according to the conclusions of the reviewed studies, significantly increase the accuracy and efficiency of image recognition at low computational costs [6]. Mobile applications that help track calorie intake based on food databases and their nutritional value are also considered [7], here apps were selected using a keyword search of related terms and reviewed: text entry apps Cronometer, DiaryNutrition, DietDiary, FoodDiary, Macros, and MyPlate; image entry apps FoodView and MealLogger; and text plus image entry apps Bitesnap, myCircadianClock, and MyFitnessPal.

The disadvantage of these systems is the lack of flexibility of the theoretical apparatus and applied solutions, which are the basis of their creation.

The purpose of this study is the development of ergonomic architectural solutions, algorithms for counting and optimization, which are implemented in specialized software in order to increase the efficiency and convenience of providing services in the field of nutrition and health on the basis of the developed software system for calculating calorie content, in particular, the theoretical foundations of building software systems for calculating calories were studied, the analysis of

solutions for the creation of software systems for calculating calories and The task is to design, implement and test a software system for calculating calories with better characteristics.

The object of the research is the processes of calorie counting, the subject of the study is a software system for calculating calorie content. Research methods include system analysis and synthesis, generalization, modeling, optimization, etc.

2. Materials and methods

Domain Algorithms

The development of calorie counting software usually involves the use of algorithms to estimate the energy content of foods. The algorithmization of calorie counting in software nutrition systems is based on several stages. First, the user enters information about the consumed products or selects them from the database. The system uses data on the nutrient content of these foods, such as proteins, fats, carbohydrates (BFAs), as well as energy content.

In the second stage, the number and volume of servings that the user consumes are taken into account, which may include weight measurement, volume, and the choice of standard portion sizes. After entering all the metrics, the system calculates the total number of calories to eat. Calorie calculation software can be implemented on a variety of platforms, programming languages, and libraries. Calculating the number of calories in foods can be based on different formulas, however, the basic idea is to determine the energy content of a product based on its chemical composition. The two most well-known formulas are the Harris–Benedict formula and the Muffin–Saint-Jeor formula.

Although both formulas are used to calculate the number of calories, they differ in their approaches to determining the body's energy needs. The Harris–Benedict formula takes into account gender, age, height, and weight: women: $BMR = 655 + (9.6 * \text{weight in kg}) + (1.8 * \text{height in cm}) - (4.7 * \text{age in years})$; men: $BMR = 66 + (13.7 * \text{weight in kg}) + (5 * \text{height in cm}) - (6.8 * \text{age in years})$.

To find out your active metabolic rate, you need to additionally multiply this number of calories by a factor that depends on the level of activity: 1.2 – sedentary lifestyle; 1.375 – light exercise one to three days a week; 1.55 – moderate exercise six to seven days a week; 1.75 – strenuous physical exercise daily; 1.9 – strenuous exercise two or more times a day.

The advantages are that the formula is easy to calculate and understand, which explains its widespread application. Among the disadvantages, one can single out the fact that the client's personal indicators are not taken into account and there is no information about the level of activity, which can lead to inaccurate results. The Muffin–São Jeora formula also takes into account gender, age, height and weight, and immediately uses the activity indicator: For women: $(10 * \text{weight (kg)} + 6.25 * \text{height (cm)} - 5 * \text{age (g)} - 161) * A$; for men: $(10 * \text{weight (kg)} + 6.25 * \text{height (cm)} - 5 * \text{age (g)} + 5) * A$; where A is the level of activity of a person, which is usually distinguished by five levels of physical activity per day: minimum activity: $A = 1.2$; weak activity: $A = 1.375$; average activity: $A = 1.55$; high activity: $A = 1.725$; extra activity: $A = 1.9$. The difficulties of calculation and the need for more accurate information are its main disadvantages.

While the Harris-Benedict formula is commonly used for an overall estimate, the Muffin–Saint-Jeor formula can provide more personalized results, especially when determining the number of calories needed to reach a specific weight.

According to H. Mitchell, food nutrients can be calculated using the Atwater system, based on the following formulas: proteins: 30% of the caloric content of the diet; fats: 30% of the caloric content of the diet; Carbohydrates: 40% of the caloric content of the diet]. The Atwater system is used in the food industry to determine the total calorie content of food using the 4 – 9 – 4 method. This system uses energy conversion factors to the macronutrients of carbohydrates, fats, proteins, and fiber. Average energy values are given as the number of calories per 1 gram of macronutrient: 1

gram of protein = 4 kcal; 1 gram of fat = 9 kcal; 1 gram of carbohydrates = 4 kcal. These values are based on the average energy output of these nutrients during digestion and metabolism.

Based on the algorithmic efficiency and the potential for increasing the ability to optimize dietary decisions, aggregation methods were used in our study.

Architectural solutions

The most common software system architectures today are: client-server, monolithic, service-oriented, microservice architecture, etc.

An example of a calorie counting app used by a client-server architecture would be MyFitnessPal. After all, its client part interacts with the server, where the database of food products and calorie counting algorithms are located. However, even such a common software product had security problems. Nearly 150 million users of Under Armour's MyFitnessPal fell victim to a data breach in February 2018 that gave hackers access to usernames, passwords, and email addresses.

Multi-component architecture is another approach to the development of software systems based on the division of functionality into small autonomous components. Each component can perform its specific function and interact with other components to achieve the overall goal of the system. The main characteristics of a multi-component architecture are modularity, which makes it easier to work on individual components, reuse of code, which contributes to the efficient use of resources, and simplifies system development due to the independence of work on components. On the other hand, managing a large number of components can be a challenge, especially in complex systems. Ensuring interoperability between components and addressing compatibility issues may require additional effort and Resources. An example of a multi-component application is the Fitbit system, where an activity tracker, a mobile app, and a website perform different functions, and their interactions form an integrated information system for tracking health and physical activity. In 2019 Fitbit was the fifth-largest wearable technology company (wireless devices, fitness monitors, and activity trackers such as smartwatches, pedometers, and heart rate and sleep quality monitors) by volume of shipments. In 2020, more than 120 million Fitbit devices were sold to 30 million users in 100 countries.

Distributed systems use components that operate in infocommunication networks and are characterized by decentralization, distributed interactions between computing nodes. Communication between computing nodes can be a source of delays and costs, and solving consistency issues becomes more challenging. In addition, ensuring security in distributed systems is an important task due to the large number of computing nodes and access points. An example is Nutrino, a web-based platform that uses distributed systems to collect and process nutrition data and provide personalized recommendations. Microservice architecture is a novel approach to software development in which an application is divided into small, standalone services, each of which performs a limited number of functions. All components of the system interact with each other through an API, which allows them to function independently of each other. The main idea is to divide functions into separate components that can be developed, scaled, and released separately. Among the disadvantages of such an architecture, first of all, there is a decentralized structure, which can cause difficulties in managing communications and the network, versions and updates. Also, if choose "microservices", you need more sophisticated monitoring tools to track status and performance when compared to a monolithic architecture. For example, HealthyTrack is a mobile health app that uses microservice architecture to optimize various aspects of calorie counting and healthy eating.

Monolithic architecture is a traditional approach to building software systems, in which all functionality is concentrated in a codebase and executed in a single process [8,9].

Running and upgrading a system in a monolithic architecture requires minimal effort because all components are together, making the deployment process relatively simple and straightforward for developers. Thanks to the centralized structure, monitoring and debugging processes also become easier. However, with the development of the software system, there are problems of flexibility,

scalability, and debugging. For example, the Cronometer system is an application for keeping a daily food log, where all the functionality (data entry, analysis, statistics) is built into a single monolithic application. This macronutrient tracking application was created in 2005 by Director A. Davidson.

In addition to food tracking, the app provides the ability to monitor exercise, which is part of the functionality of the free version. If upgrade to the paid version of Cronometer Gold, it will be possible to track biometric data or various body and cardio measurements, as well as laboratory test results.

Table 1 provides a comparative description of monolithic and microservice architectures.

Table 1
Comparative Characteristics of Monolithic and Microservice Architectures

Setting	Monolithic architecture	Microservice Architecture
Flexibility	Low	High
Scalability	Limited	High
Control	Simple	Complex
Dependencies		
Isolation & Renewal	Complex	Simple

Taking into account the above-mentioned advantages and disadvantages of software systems with different architectural solutions, a monolithic architecture was chosen as a satisfactory option in terms of balancing the criteria in Table 1.

Functional properties

Table 2 provides a comparative description of the above software systems.

Table 2
Comparative Analysis of Calorie Counting Systems

Functional	Cronometer	MyFitnessPal	Fitbit
Calorie counting	Yes	Yes	Yes
Keeping a food journal	Yes	Yes	Yes
Creating a Personalized Menu	No	No	No
Personalized Nutrition Recommendations	Yes	No	Yes
Tracking physiological parameters	No	Yes	Yes
Integration with other devices	Yes	Yes	Yes
Mood Tracking	No	No	Yes
Statistics in the form of pie charts	No	Yes	Yes

As a result of the comparative analysis, the basic functionality of calorie counting software systems was investigated and it was found that although all software systems provide many functions, they lack a clear and user-friendly interface and functions such as creating a weekly menu, compiling a list of products from a selected meal plan and the ability to make real-time changes by the client to the application application.

A calorie counting software system should provide users with a convenient and effective tool for controlling their diet and maintaining a healthy lifestyle. It allows you to register with the entry of

personal data, determines calorie norms and helps in meal planning. The system also provides an opportunity to monitor eating habits and physical activity, contributing to the achievement of specific health goals.

In our study, the functionality was significantly increased by implementing the functions of advanced settings change, compiling a weekly menu and grocery list, adding/removing dishes, etc.

3. Results

Functionality and security

A software system with a monolithic architecture is useful for those users who have a medical need for daily tracking of food intake taking into account the number of calories and macronutrients, or a desire to learn more about the meals they consume daily. For convenient use of the program, a simple, understandable, user-friendly interface has been developed, which allows the user to quickly get all the necessary information. The developed information system provides the following opportunities: registration with the introduction of such personal data that will help in compiling a unique menu, in particular, these are indicators of height, weight, age, tendency to allergies, etc.; ensuring the security of stored user data; login to the information system using a username and password; change of personal data; automatic determination of the client's calorie norms; search and add dishes; formation of a list of favorite or recommended dishes; tracking eating habits and physical activity with the help of a journal; getting a table from the menu; saving the menu in html, pdf format; getting a list of products, etc.

The software system has the following functions: registration; authorization; changing settings; data saving; output of data in the form of charts; search and adding/deleting information; automatic generation of a table based on human physiological indicators; saving a table and list; calculation of kilocalories that a person needs to consume; automatic generation and output of a list of products, based on food formed in a pre-compiled nutrition plan. Software Security Requirements The systems are as follows: user passwords must be encrypted using the pwncrypt algorithm before being stored in a database to prevent the possibility of unauthorized access to personal data; using stored procedures to perform data operations to restrict direct access to tables and providing access for control and monitoring only through defined procedures.

Non-Functional Requirements

The performance requirements of a calorie counting software system are as follows: ensuring efficient use of resources to quickly calculate calories and other data, which will allow users to get results quickly without undue delay; Using database queries and caching results to reduce server load and speed up the application. The usability requirements for a calorie counting software system are as follows: providing an intuitive and user-friendly interface that contributes to the convenience of interacting with the application; interface development based on design principles such as simplicity, consistency and accessibility to ensure the acquired user experience; clear distribution of the logic of business processes, presentation and management of data to ensure the simplicity of developing and ensuring the functioning of the application information system.

Software System Architecture

The monolithic architecture of the system is implemented in the work. Within the framework of this architecture, the functional elements of the calorie counting software system were modeled and designed on the basis of the corresponding UML diagrams. To model the functionality of the software system in terms of its interaction with the user, a use case diagram was developed. The usage diagram of this application consists of one actor – the user and eight main use cases, containing Involved use cases with the "include" relationship (Figure 1).

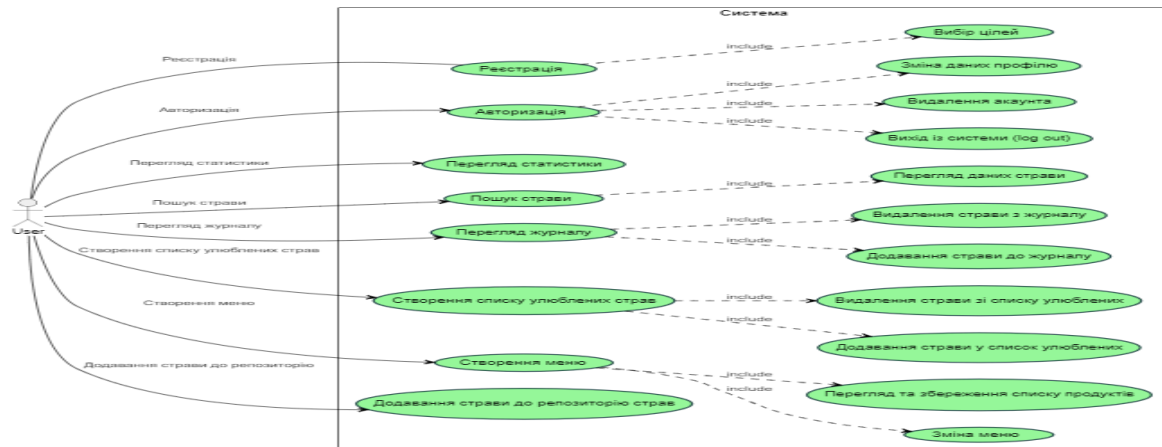


Figure 1: Use Case Diagram According to the Functional Requirements of the Calorie Counting Software System.

As stated above, a monolithic software system architecture is a traditional model of software development where all software is developed as a single piece of code. All components of the architecture are interdependent and interconnected, and each component is required to run the application. The advantages of this approach are ease of development and implementation, easy scalability in the initial stages, and the disadvantages are the difficulty of scaling if the software system becomes large, higher risks associated with support and modification. Finally, it can lead to longer deployment times, difficulties in implementing new features, and more difficult testing due to the large

interdependence of components. Modeling the efficiency of the process of developing and modifying a software system of monolithic architecture for specified functionality and typical users showed the advantage of monolithic architecture by -16-20% compared to microservice.

Figure 2 depicts the architecture of a calorie counting software system.

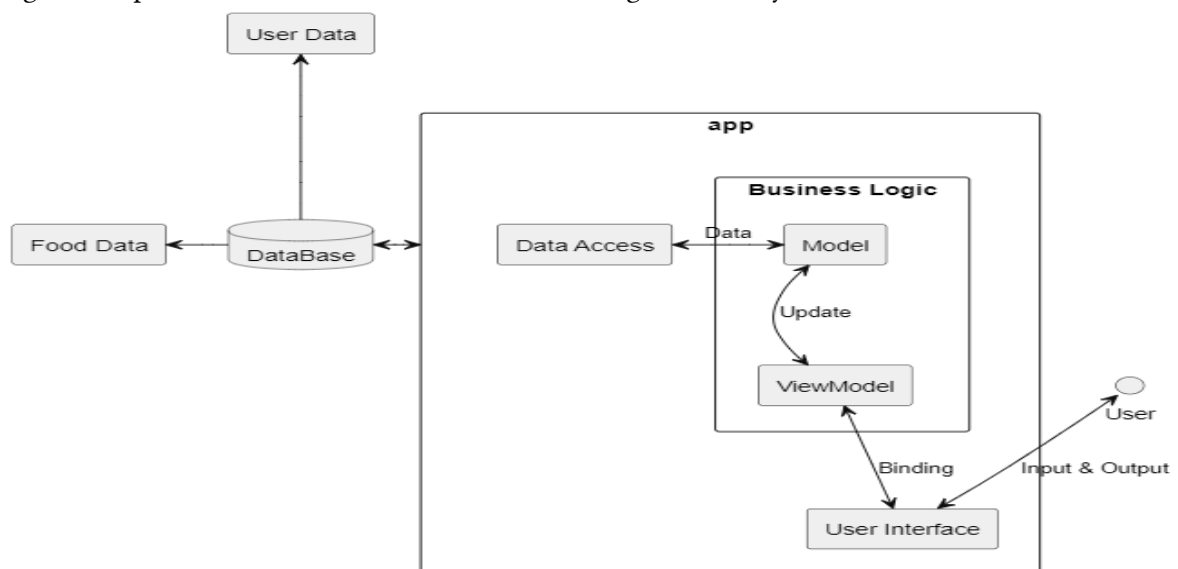


Figure 2: Calorie Counting Software System Architecture.

Development and customization technologies

To reduce the complexity of the program code, the MVVM (Model – View – ViewModel) architectural pattern was used. In the process of creating a calorie counting software system, the following was applied: Microsoft Visual Studio 2022 development environment; .Net Framework version 7.0.0; ADO.NET methodology; Windows Presentation Foundation (WPF) technology C#

programming language; XAML markup language; Microsoft SQL Server 2019 database, Microsoft SQL Server Management Studio 19.

To implement the calorie counting information system, an application for Windows was created. The organization of the system architecture is shown in Figure 3.

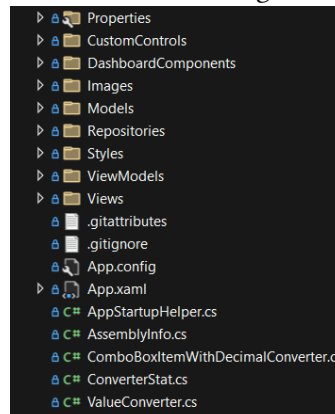


Figure 3: Organization of the System Architecture.

Due to the fact that the MVVM template provides a View-ViewModel-Model structure, the software system has been divided into appropriate categories, which can be presented in the form of project catalogs. All pages and windows have xaml files in the Views folder, and the logic classes are in the ViewModels folder. To represent data objects, there are user models and dishes in the Models catalog, as well as interfaces for working with the database in the Repositories folder. The styles of the app's visual elements, such as colors and size, are located in the Styles folder. The application contains two windows – authorizations, where you need to log in to the information system or register, and the main window, where statistics, dishes and customer profile settings are displayed. The view model is responsible for switching the pages of the main window, and the view is responsible for the display. The application consists of such elements as an authorization window: login; registration; Main window: statistics page, dish search page, add/remove dishes page, favorite (recommended) dishes page, menu page, settings page. The statistics page is a wrapper for children placed in the Dashboard Components folder.

Database

The system's database contains thirteen tables, each of which has a primary key. Tables are divided into two main groups: those that belong to user data and those that store information about dishes. The user group includes three tables: "Users", "Activity", "UserAllergy". The "UserAllergy" table contains two foreign keys and connects the user to the allergen from the "Allergens" table, which in turn is linked to the "ProductAllergens" table, which contains information about the allergenic products. The relationship "UserAllergy" matches the username to the type of allergy, and the "ProductAllergens" relationship between the name of the product in the list of products and the name of the food ingredients. In this way, allergy information is linked with the aim of further drawing up a nutrition plan based on the stored data. The group has five tables in relation to meals, namely: "Products", "Meals", "MealNutrition", "MealProducts", "ProductAllergens", interconnected by the ID of the dish. The structure of the database is presented in Figure 4.

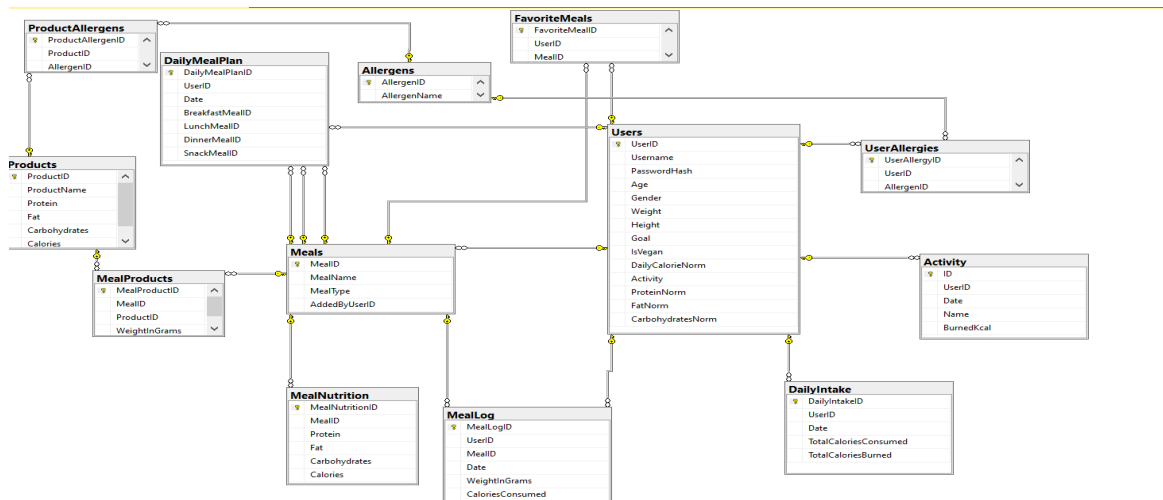


Figure 4: Calorie Counting Software Database Schema.

Authorization

In order to work with the software system, the user needs to log in to it, so when booting, an authentication window is called, which receives the values of the login and password and passes them to the view model, which in turn uses the user repository that is connected to the database to check the existence of a user with specified indicators. If there is no such client, then a corresponding message appears. After that, you can try to log in again. If there is a user with the specified username and password, depending on the entered parameters, the interface implemented in the main program window is called and the authentication form is closed. The page named "Registration" opens in the "Login" window. The user registration page contains 10 fields for filling in user data, a registration confirmation button and a link to the authorization page. The fields require you to enter the following information: username; password; age; height; weight; gender; allergies (other diseases); vegetarianism; activity; goal. All fields except "vegetarianism" are mandatory.

After launching the application, the login window opens with the authorisation page. If the user does not have an account, he or she is redirected to the registration page by clicking the 'Register' link, where he or she enters the required data. All fields except for the vegetarian checkbox are mandatory, so if at least one of them is not filled in, the confirmation button will not become active. There are also age restrictions, ranging from 16 to 100, height restrictions, ranging from 150 to 210, and weight restrictions, ranging from 40 to 120, and if you do not meet these restrictions, a message will appear and the field for re-entering will be set to zero.

Data Analytics

The statistics page is one of the main ones, as it displays several different statistical elements to facilitate the understanding of the user's calorie intake. The chart displays the percentage of calories consumed to the daily value. The information to the right of the diagram indicates the proteins, fats and carbohydrates consumed, as well as their norm. The data in the middle of the page shows the number of kilocalories consumed and burned. And the bottom bar chart gives you an idea of the trend in calorie intake over the last 7 entries. All information is updated automatically after appropriate changes are made to the database. The program provides an opportunity for the user to contribute to the expansion of the database of dishes. To do this, on the corresponding "Contribute" page, there is a form for adding a dish containing a name field, a drop-down list of the type of dish, a checkbox to determine whether this dish is vegetarian, a "container" for adding products. The software system also has the ability to save dishes to the list of favorites. On the settings page, the user can change separately such indicators as age, height, weight, goal, activity level, as well as delete their account. In contrast to the existing developments, this system at the methodological level provides for the possibility of connecting an optimization unit based on the solution of the classical problem of linear programming of diet selection.

Testing

To check the performance of the software system, two methods were chosen: automated unit testing and manual testing of various scenarios using the "black box" method. Particular attention is paid to testing the behavior of ViewModels, which are responsible for the logic of the application and link the user interface to the data. This testing was done using NUnit Test and ReSharper.

Conclusions

The article highlights the results of existing research, highlights the unresolved parts of the problem and proposes methodological approaches to the creation of software systems for calorie counting and further optimisation of the diet, and offers engineering solutions that have provided an increase in some estimates by twenty per cent compared to existing solutions. This information system uses an intuitive interface and convenient data entry functions, and ensures reliable protection of users' personal data, which is an important factor in the trust in the calorie counting information system and its long-term use. Modern calorie counting software systems should be comprehensive, personalised, flexible and intuitive, while ensuring the protection of personal data and scientific validity, enabling users to achieve their personal goals in the field of healthy eating and fitness. Further scientific research is planned to support linguistic and geographical differences with the possibility of adapting the software system to different languages and culinary traditions of different regions of the world.

References

- [1] O. Fediv, M. Patrati Principles of rational healthy eating, 2015. URL: <http://surl.li/smryp>.
- [2] R. Haque, R. Hhan, N. A. Shihavuddin, M. Mahbubul Syeed, M. Faisal Uddin, Lightweight and Parameter-Optimized Real-Time Food Calorie Estimation from Images Using CNN-Based Approach, in: *Appl. Sci.* 2022, 12(19), 9733. doi.org/ 10.3390/app12199733.
- [3] R. Jayasingh J., B. Nagajayanthi, M. Neebha, M. Sagayam, A.A. Elngar, New calculation of calorie content and determining nutritional level from day-to-day intake of food using Image Processing, in: *Research Square*. 2022. doi.org/10.21203/rs.3.rs-1766525/v1.
- [4] S. Klemm, 7 Ways to Enhance the Flavor of Your Meals, 2019. URL: <https://www.eatright.org/food/food-preparation/cooking-tips/enhancing-the-flavor-of-your-meals>.
- [5] R. An, J.M. Perez-Cruet, X. Wang, Y. Yang, Build Deep Neural Network Models to Detect Common Edible Nuts from Photos and Estimate Nutrient Portfolio in: *Nutrients*. 2024. 16(9). pp. 1294 -1303. doi.org/10.3390/nu16091294.
- [6] J. Patel, K. Modi, Indian Food Image Classification and Recognition with Transfer Learning Technique Using MobileNetV3 and Data Augmentation in: *Eng. Proc.* 2023. 56(1). pp. 197-203. doi.org/10.3390/ASEC2023-15341.
- [7] S. Gioia, I.M. Vlasac, D. Babazadeh, et al. Mobile Apps for Dietary and Food Timing Assessment: Evaluation for Use in Clinical Research in: *JMIR Formative Resources*. 2023. 7. <https://doi.org/10.2196/35858>.
- [8] R. C. Martin Clean, Architecture: A Craftsman's Guide to Software Structure and Design, 1st. ed., Pearson, 2017, 432 p.
- [9] M. Kleppmann, Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems 1st. ed., O'Reilly Media, 2017, 611 p.