

2. Polyanin, A. D., Manzhirov, A.V. (2003) Handbook of integral equations [*Spravochnik po integralnim uravneniyam*]. Moscow: FIZMATLIT.
3. Zarubin, A. N. (2017). The Tricomi Problem for a nonlinear mixed-type equation with functional delay and advance [*Zadacha Tricomi dlya nelineinogo uravneniya smeshannogo tipa s funkcionalnim zapazdivaniem i operejeniem*] *Differential equations*. Vol. 53(8). Pp.1064-1073.

УДК  
004.5

**РАЗРАБОТКА МОБИЛЬНОГО СЕРВИСА ДЛЯ  
ПЕРСОНИФИКАЦИИ ПИТАНИЯ СПОРТСМЕНОВ НА ОСНОВЕ  
УЧЕТА ИХ ИНДИВИДУАЛЬНЫХ ПОТРЕБНОСТЕЙ**

**Юлия Игоревна Иванова**

студентка  
ivanova788@mail.ru  
г. Москва

**Михаил Леонидович Рысин**

к.п.н., доцент  
m.l.rysin@mgutm.ru  
г. Москва

Московский государственный университет  
технологий и управления им.  
К.Г. Разумовского (ПКУ)

**Аннотация.** Статья посвящена описанию практических приёмов проектирования и программной реализации компонентов мобильного сервиса, предназначенного для персонализации питания спортсменов и людей, ведущих активный образ жизни. Приведены особенности использования инструментальных средств разработки программных модулей, реализующих алгоритмы учета индивидуальных потребностей в питании пользователей мобильного сервиса.

**Ключевые слова:** мобильное приложение, архитектура «клиент-сервер», IDEF0, ER-модель, инфологическая модель, Android, пользовательский интерфейс, программирование, SQL, Java, PHP.

Питание является фактором поддержания тела человека в здоровом и нормально функционирующем состоянии. В настоящее время в научном мире актуальна тема персонализации питания. Существует большая зависимость рациона человека от пола, возраста, массы тела, вида и уровня физической активности, личных и культурологических предпочтений, непереносимости и/или пищевой аллергии и т. д.[4]. Имеет практический смысл объединить в группы людей, у которых особенности организма, активность и потребности различаются не критично. Например, можно выделить такую группу как спортсмены. Зачастую перед спортсменами и людьми, ведущими активный образ жизни, стоят проблемы, решение которых напрямую зависит от оптимальности рациона питания (например, набор массы, сброс веса или поддержание организма в конкретных рамках).

Таким образом, имеется возможность объединить таких людей в группу и составить некоторое количество рационов для группы, но при этом остается проблемой необходимость автоматизации учета потребностей и формирования рациона питания для каждого участника. Кроме того, человек питается несколько раз в день, поэтому весь сервис подбора рациона должен быть доступен потребителю в любой момент времени.

Сегодня огромную роль в жизни большинства людей играют мобильные девайсы – телефоны, смартфоны, планшеты. Посредством смартфонов происходит интенсивное общение в виде живых коммуникаций (аудио- и видеозвонки), письменных диалогов, а также

хранение фотографий, книг, иных данных. Большую часть дня человек имеет доступ к возможностям этих устройств. Поэтому решать задачу автоматизации составления рациона питания для спортсменов целесообразно с помощью специализированного мобильного сервиса, основу которого составляет мобильное приложение.

Мобильное приложение – это компьютерная программа, разработанная для функционирования на основе определенной мобильной платформы. У мобильной разработки существуют особенности, т.к. устройства, для которых разрабатывается приложение, функционируют от батареи и наделены процессорами с меньшей, чем у десктопов, производительностью. Мобильные приложения разрабатываются на различных языках программирования высокого уровня, после чего транслируются в промежуточный или нативный код мобильной платформы [3]. Современные мобильные устройства снабжены различными дополнительными возможностями, например, камерой или гироскопом, что позволяет расширить и дополнить приложения уникальным набором функций.

Оптимальным выбором платформы для мобильного сервиса Android (см. табл. 1). Она имеет открытый исходный код и инструментальные среды разработки. К тому же, охват аудитории с выбором Android будет наибольшим из возможных.

*Таблица 1.  
Сравнительный анализ существующих мобильных платформ.*

критерии \ ОС	Windows	IOS	Android
<b>Открытый код</b>	нет	нет	да
<b>Доступная среда разработки</b>	да	нет (только для устройств компании Apple)	да
<b>кол-во приложений, доступных в магазинах</b>	менее 500 тыс	около 2 млн.	более 3,5 млн.
<b>Официальная поддержка системы</b>	нет	да	да
<b>Кроссплатформенность</b>	да	нет (только устройства компании Apple)	да

Также, проанализировав основные инструменты для разработки под мобильную операционную систему Android (см. табл. 2), становится ясно, что целесообразно выбрать IDE AndroidStudio. Также она является бесплатной и единственной официально распространяемой средой от компании-разработчика ОС Android [6].

*Таблица 2.  
Сравнительный анализ инструментальных средств разработки для Android*

критерии \ IDE	Eclipse	NetBeans	IntelliJIDEA	Android Studio
<b>набор плагинов</b>	+	+	+	+
<b>корректность работы плагинов между собой</b>	–	+	+	+
<b>кроссплатформенность</b>	+	+	+	+
<b>бесплатность</b>	+	+	–	+
<b>официальность</b>	–	–	–	+

Разработка сколько-нибудь серьезного программного решения предполагает в качестве основы составление технического задания [1, 2], согласованного с заказчиком – АО «Торговый дом "Биоснабсбыт"» (Московская обл.).

В соответствии с техническим заданием в составе мобильного сервиса будет присутствовать база данных, что предполагает архитектуру «клиент-сервер», в которой программное обеспечение разделено на клиентскую и серверную части.

Клиентская часть отвечает за взаимодействие с пользователем (пользовательский интерфейс), отправку запросов на сервер, получение ответов от сервера и представления их в привычном для пользователя виде.

На серверной стороне, соответственно, скрипт и платформа для его выполнения, СУБД и база данных. Серверная часть принимает запросы от клиента, обрабатывает их, выдает ответы и обеспечивает доступ к данным.

На этапе проектирования базы данных для построения её ER-модели [5] можно выделить шесть сущностей (рецепты, завтрак, второй завтрак, обед, полдник и ужин) и необходимые атрибуты для их описания (рис. 1).

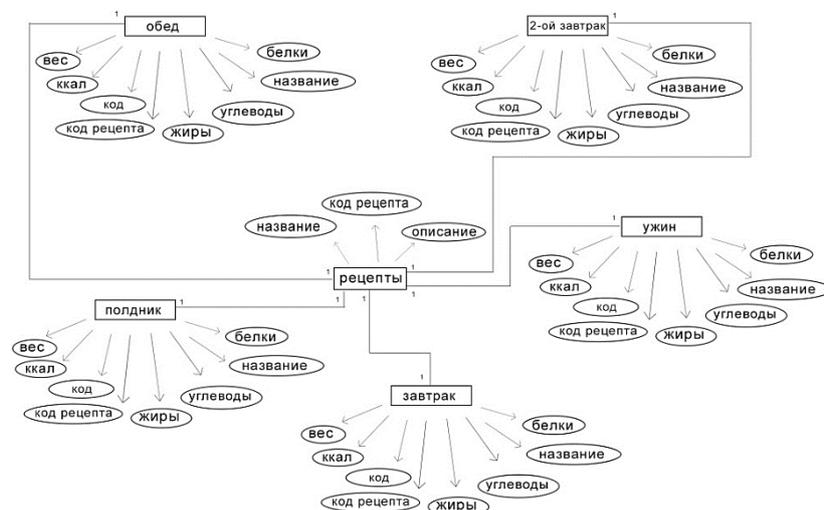


Рис. 1. ER-модель предметной области мобильного сервиса.

Исходя из ER-модели сформирована инфологическая схема базы данных (рис. 2).

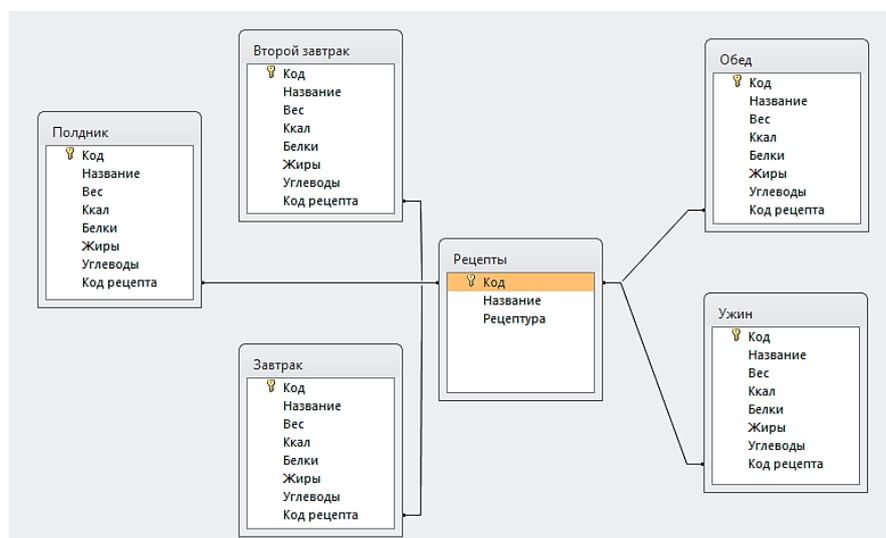


Рис. 2. Инфологическая модель базы данных мобильного сервиса.

Для разработки компонентов мобильного сервиса были использованы следующие языки:

- Java – для описания всех вычислений и работы клиентской части;
- PHP – для реализации API и взаимодействия клиента с сервером и базой данных;
- язык запросов SQL – для взаимодействия с СУБД.

Вся визуальная часть и «активити» (экраны) мобильного приложения хранятся в трёх xml-файлах:

- activity\_main.xml – вся визуальная часть экрана «калькулятор», в том числе необходимые стили и атрибуты, идентификаторы (id) элементов;
- activity\_food.xml – всё необходимое для визуализации экрана «результат»;
- activity\_recept.xml – данные экрана «рецепт».

Название базы данных, в которой хранятся необходимые таблицы - «ulivaa». Каждая таблица создается SQL-запросом, в котором указывается, в какой базе данных создается таблица, её название, все столбцы с их названиями и типами данных, а также кодировка. Так, таблица «Lunch» с данными об обедах, создается следующим образом:

```
CREATE TABLE `ulivaa`.`Lunch` ( `ID` INT(11) NOT NULL , `Name` TEXT NOT NULL , `Ccal` DOUBLE NOT NULL , `Weight` INT(11) NOT NULL , `ID_recept` INT(11) NOTNULL , `Proteins` DOUBLE NOT NULL , `Fats` DOUBLE NOT NULL , `Crds` DOUBLE NOTNULL ) ENGINE = MyISAM CHARSET=utf8 COLLATE utf8_general_ci;
```

Модуль вычислений представляет собой шесть основных файлов:

- AndroidManifest.xml – объявляются все активити, назначается главная из них, а также под-ключаются необходимые атрибуты для правильного функционирования приложения и доступа к некоторым действиям;
- MainActivity.java – все необходимые функции «калькулятора», считывания введенных данных, а также взаимодействия с остальными файлами и активити;
- FoodActivity.java – описаны все действия, благодаря которым клиент взаимодействует с файлами activity\_food.xml и Connect.java, а также, позволяющие пользователю просматривать список необходимых к употреблению блюд и обеспечивающие переход к экрану «рецепт»;
- ReceptActivity.java – функции, описывающие взаимодействие с файлами activity\_recept.xml и ConnectRecept.java и, позволяющие пользователю просматривать рецептуру выбранного приема пищи;
- Connect.java и ConnectRecept.java – файлы, обеспечивающие взаимодействие с сервером на стороне клиента;

В файле MainActivity.java: при нажатии на кнопку «рассчитать калории» из каждого элемента поля ввода считываются введенные пользователем данные, в зависимости от выбора гендера, в переменную genderChoice записывается цифра 1 или 2 – мужчина или женщина соответственно, в переменную Activ заносится коэффициент активности для формулы.

Следующий код выводит всплывающее сообщение, если какое-то поле не заполнено, а если все заполнено, то в зависимости от пола пользователя, применяется одна из двух формул, вычисляющая необходимое количество калорий, а также вывод результата на экран:

```
if (Age==0||Weight==0||genderChoice==0||Height==0){
    Toast.makeText(MainActivity.this, "Пожалуйста, заполните все поля!",
        Toast.LENGTH_SHORT).show();
} else {
    if (genderChoice==1){
        ResultCal=(88.362+(13.397*Weight)+(4.799*Height)-
            (5.677*Age))*Activ;
        ResultCal+=ResultCal*0.1;
    } else if (genderChoice==2){
```

```

        ResultCal=(447.593+(9.247*Weight)+(3.098*Height)-
            (4.330*Age))*Activ;
        ResultCal+=ResultCal*0.1;
    }
    String formattedDoubleCal = new
        DecimalFormat("#0.000").format(ResultCal);
    textViewResultCal.setText(formattedDoubleCal);
}

```

Файлы Connect.java и ConnectRecept.java содержат очень похожий друг на друга код, с небольшими несоответствиями.

Следующий метод вызывается в активити, запрашивающей соединение и передающей нужные параметры для поиска результата. Здесь параметры записываются в локальные переменные и происходит запуск соединения:

```

public void start(String IntakeF, Double StartF, Double EndF)
{
    this.Intake= IntakeF;
    this.Start= Double.toString(StartF);
    this.End= Double.toString(EndF);
    this.start();
}

```

При запуске соединения, программа формирует лист для отправки данных на сервер (по аналогии со второй строкой указываются все необходимые для передачи параметры):

```

ArrayList<NameValuePair>nameValuePairs = new ArrayList<NameValuePair>();
nameValuePairs.add(new BasicNameValuePair("Intake", Intake));

```

С помощью следующего кода, создается новое соединение с файлом PHP(в случае запроса рецепта указывается файл get\_recept.php) на сервере, который является API для приложения и в это соединение передается лист с данными:

```

HttpClient httpClient = new DefaultHttpClient();
HttpPost httppost = new
    HttpPost("http://mydiplom.kl.com.ua/get_data.php");
httppost.setEntity(new UrlEncodedFormEntity(nameValuePairs, "UTF-8"));
HttpResponse response = httpClient.execute(httppost);
HttpEntity entity = response.getEntity();
is = entity.getContent();

```

После соединения и обработки информации на сервере начинается прием ответа в формате json. Ответ считывается построчно, пока строки ответа не закончатся:

```

BufferedReader reader = newBufferedReader(new InputStreamReader(is,
    "UTF-8"), 8);
StringBuilder sb = new StringBuilder();
while((line= reader.readLine()) != null) {
    sb.append(line+ "\n");
}
is.close();
result= sb.toString();

```

Когда ответ принят, программа обрабатывает полученный файл json и записывает, по аналогии со второй строкой, значения в переменные:

```

JSONObject json_data = new JSONObject(result);
Name = (json_data.getString("Name"));

```

С помощью следующей функции (или нескольких подобных), полученный параметр возвращается в активити:

```
public String rename ()
{ return Name;}
```

API, благодаря которому клиент может взаимодействовать с базой данных, написан на языке программирования PHP с использованием программы SublimeText. Он содержит четыре файла, загруженных на сервер, к которым обращаются файлы Connect.java и ConnectReceipt.java.

Файл db\_config.php, содержащий информацию для соединения с базой данных – логин панели БД, ее пароль, название базы данных, в которой находятся таблицы и сервер, на котором, непосредственно, лежит вся информация БД.

Файл db\_connect.php содержит функцию обращения к файлу db\_config.php для импорта переменных для соединения с бд, подключается к серверу, если все верно, или удаляет соединение при ошибке, подключается к базе данных и возвращает соединение, а также функцию закрытия соединения.

К файлам get\_data.php и get\_receipt.php обращается клиент. В результате работа приложения может быть продемонстрирована следующим образом (рис. 3):

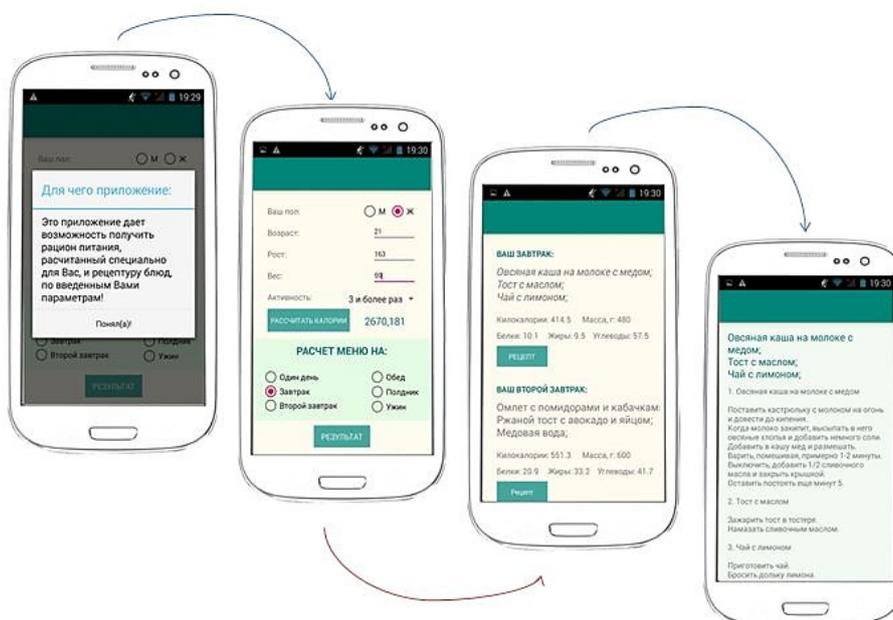


Рис. 3. Вид экрана смартфона на различных этапах исполнения мобильного приложения.

Таким образом, все поставленные в техническом задании требования были реализованы. Эффективность предложенного решения подтверждается экспертным заключением со стороны заказчика АО «Торговый дом "Биоснабсбыт"» в форме акта о внедрении.

### Список литературы

1. ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.
2. Орлов С. А. Программная инженерия. Учебник для вузов. 5-е изд. СПб: Питер, 2016.
3. Шакирова Ю.К., Савченко Н.К., Абилдаева Г.Б., Зайцева С.В., Мартыненко О.В. Проектирование мобильных приложений и облачных сервисов // Молодой ученый. 2018. №2. С. 254-258.
4. Портнов Н.М., Карпов В.И. Задача оптимизации меню в системе персонализированного питания // Системный анализ в проектировании и управлении сборник научных трудов XXIII Международной научно-практической конференции.

- Санкт-Петербургский политехнический университет Петра Великого. СПб, 2019. С. 102-109.
5. Burbank D. Data modeling made simple with CA ERwin Data Modeler R8. [Текст] / Burbank D., Hoberman S. – Technics Publications, 2011. 536 p.
  6. Android Studio and SDK tools. [Электронный ресурс] URL: <https://developer.android.com/studio> (дата обращения 21.07.19).

## THE MAKING MOBILE SERVICES TO ATHLETES'S NUTRITION PERSONALIZE BY TAKING INTO ACCOUNT THEIR INDIVIDUAL NEEDS

<p><b>J.S. Ivanova</b> student ivanova788@mail.ru Moscow</p> <p><b>M.L. Rysin</b> Cand. Sci. (Pedagogy), associate professorm.l.rysin@mgutm.ru Moscow</p>	<p>K.G. Razumovsky Moscow State University of technologies and management (FCU)</p>
---	---

**Abstract.** The article shows a description of practical techniques for designing and making software components of a mobile service designed to personify the nutrition of athletes and people leading an active lifestyle. Features of the use of development tools of software modules that implement the algorithms for accounting for individual nutritional needs of mobile service users are presented.

**Keywords:** mobile application, client-server architecture, IDEF0, ER-model, infological model, Android, user interface, programming, SQL, Java, PHP.

### References

1. GOST 34.602-89. Informatsionnaya tekhnologiya. Kompleks standartov na avtomatizirovannyye sistemy. Tekhnicheskoye zadaniye na sozdaniye avtomatizirovannoy sistemy. [*Information technology. Set of standards for automated systems. Terms of reference for the creation of an automated system*]. Moscow.
2. Orlov, S.A. (2016) Programmnaya inzheneriya. Uchebnyk dlya vuzov. St. Petersburg: Piter.
3. Shakirova, Y.K., Savchenko, N.K., Abildaeva, G. B., Zaitseva, S.V., Martynenko, O.V. (2018) Proyektirovaniye mobilnykh prilozheniy i oblachnykh servisov [*Designing mobile applications and cloud services*]. *Young scientist*. Vol. 2. Pp. 254-258.
4. Portnov, N.M., Karpov, V.I. (2019) Zadacha optimizatsii menyuu v sisteme personifitsirovannogo pitaniya [*The task of optimizing the menu in the system of personified nutrition*]. System analysis in design and management collection of scientific papers of the XXIII International scientific and practical conference. St. Petersburg Polytechnic University of Peter the Great. Pp. 102-109.
5. Burbank, D. (2011) Data modeling made simple with CA ERwin Data Modeler R8. Technics Publications.
6. Android Studio and SDK tools. URL: <https://developer.android.com/studio> (accessed 21.07.19).