

УДК 004.415

***РАЗРАБОТКА ФУНКЦИОНАЛА ЛИЧНОГО КАБИНЕТА
ПОЛЬЗОВАТЕЛЯ В JAVA SPRING-ПРИЛОЖЕНИИ***

Сафина Г.Ф.

к. ф.-м. н, доцент,

ФГБОУ ВО «Уфимский университет науки и технологий», Нефтекамский филиал, Нефтекамск, Россия

Кириллова Е.А.

студент,

ФГБОУ ВО «Уфимский университет науки и технологий», Нефтекамский филиал, Нефтекамск, Россия

Аннотация

В работе представлена практическая реализация функционала личного кабинета пользователя в Java Spring-приложении, включающего управление профилем, верификацию электронной почты и безопасную смену пароля. Описаны методы генерации и проверки временных кодов подтверждения email. Продемонстрирована интеграция фасадов и сервисов для обработки HTTP-запросов и формирования корректных ответов клиенту.

Ключевые слова: Java, Spring Data, Spring Framework, базы данных, трёхслойная архитектура.

***DEVELOPING THE FUNCTIONALITY OF A USER'S PERSONAL
ACCOUNT IN A JAVA SPRING APPLICATION***

Safina G.F.

PhD, Associate Professor,

*Neftekamsk branch of the Ufa University of Science and Technology,
Neftekamsk, Russia*

Kirillova E.A.

student,

Neftekamsk branch of the Ufa University of Science and Technology,

Neftekamsk, Russia

Abstract

The paper presents a practical implementation of the user's personal account functionality in a Java Spring application, including profile management, email verification and secure password change. Methods for generating and checking email confirmation time codes are described. The integration of facades and services for processing HTTP requests and generating correct responses to the client was demonstrated.

Keywords: Java, Spring Data, Spring Framework, databases, three-layer architecture.

Личный кабинет пользователя любой информационной (цифровой, образовательной и т.п.) системы выступает центральной точкой взаимодействия пользователя с системой, обеспечивая управление персональными данными, контроль безопасности учетной записи и выполнение операций, связанных с конфиденциальной информацией. Разработкам таких личных кабинетов (ЛК) для различных систем посвящено множество работ, в том числе труды [1-5].

Использование пользователем личного кабинета в любой сфере деятельности предполагает не только взаимодействие с данной системой (платформой), а работу с учетом информационной безопасности этого взаимодействия, т.е. защиты как персональных данных пользователя, защиты документации, защиты функциональных возможностей работы в личном кабинете и т.п.

К ключевым пунктам, необходимым для безопасного функционирования ЛК и работы в ней, можно отнести:

– компетенции пользователя в управлении профилем;

- умения и опыт в обеспечении защиты данных (в том числе персональных);
- опыт работы с различными информационными и образовательными системами и их функционалом;
- навыки в сфере применения и защиты (шифрование, хеширование и другое) информационных систем, приложений, платформ, технологий и языков программирования.

В рамках представленной работы предлагается алгоритмическая реализация функционала личного кабинета пользователя в Java Spring-приложении с поддержкой изменения профиля, верификации электронной почты и безопасной смены пароля на основе подтвержденного email.

Представленное решение соответствует техническому заданию, согласно которому пользователь после успешной аутентификации должен иметь возможность редактировать сведения о себе, инициировать процедуру верификации почты и изменять пароль исключительно при подтвержденном статусе email. Все методы управления размещены в фасаде `SsaUserLkApiFacade`.

Функциональность личного кабинета на уровне логики приложения представлена API фасадом, сервисный слой инкапсулирует бизнес-логику, а слой хранения реализован через фасад доступа к данным `SsaStorageFacade`. Такой подход позволяет обеспечить слабую связанность компонентов и высокую масштабируемость решения [1,2].

Основной управляющий компонент модуля – класс `SsaUserLkApiFacadeImpl`, который агрегирует вызовы к нескольким сервисам, включая `SsaUserLkService`, `SsaMailVerificationService` и `SsaAuthService`. Он отвечает за обработку HTTP-запросов и формирование корректных ответов клиенту [3-6].

Приведем листинг алгоритма обновления профиля пользователя:

```
@Override
public ResponseEntity<SsaUserLkJson>
updateUserProfile(SsaUserLkUpdateRequest request)
{
    String userId = userUtil.getUserId();
    SsaUserLkJson toRet = userLkService.updateUserLk(request,
userId);
    return ResponseEntity.ok(toRet);
}
```

Листинг 1 – Пример метода обновления профиля пользователя

Верификация email построена на механизме временных кодов подтверждения. После вызова метода `sendVerificationCode` пользователю генерируется случайный цифровой код, который хэшируется и сохраняется в таблице `ssa_mail_verification` с указанием срока действия.

Процесс генерации и отправки кода будет иметь следующий листинг:

```
final String code = generateCode(4);
SsaMailVerification verification =
storage.newSsaMailVerification(
    UUID.randomUUID().toString(),
    userId,
    email,
    SsaRestoreType.V,
    encrypt.bcryptHash(code),
    times.nowZ().plusHours(1)
);
storage.save(verification);
```

Листинг 2 – Процесс генерации и отправки кода

Смена пароля возможна только при подтвержденном email, что повышает уровень безопасности [7]. Данная проверка реализована в методе `changePassword` сервиса `SsaAuthServiceImpl`. Перед сохранением нового пароля осуществляется несколько последовательных проверок: подтвержден ли email, совпадает ли старый пароль, и не дублирует ли новый текущий.

Приведем далее листинг кодирования безопасной смены пароля:

```
if (user.getEmailVerified() == null ||
!user.getEmailVerified())
{
    webExcept.throwByStatusAndDesc(HttpStatus.FORBIDDEN,
"Email not verified");
}
if
(!user.getPassHash().equals(encrypt.bcryptHash(request.getPassword(
))))
{
    webExcept.throwByStatusAndDesc(HttpStatus.FORBIDDEN, "Old
password mismatch ");
}
user.setPassHash(encrypt.bcryptHash(request.getNewPassword()))
storage.save(user);
```

Листинг 3 – Реализация безопасной смены пароля

Таким образом, в работе показана реализация функционала личного кабинета пользователя в приложении на языке программирования Java Spring. Функционал содержит:

- управление профилем пользователя в ЛК;
- верификацию электронной почты;

– безопасную смену пароля.

Разработанный функционал личного кабинета демонстрирует эффективность структурированного подхода к организации бизнес-логики и взаимодействия с пользователем.

Использование фасадов, сервисного слоя и фасада хранения данных позволяет обеспечить безопасное управление персональными данными, контролировать операции с конфиденциальной информацией и поддерживать расширяемость системы.

Реализация верификации email и проверки условий смены пароля повышает уровень защиты учетной записи, а представленная архитектура способствует дальнейшему развитию и интеграции функционала без значительного усложнения кода.

Библиографический список:

1. Основы программирования на языке Java: учебное пособие / А. Ю. Богачев, В. Л. Шапошников, А. А. Минаков [и др.]. – Краснодар: ФГБУ «Российское энергетическое агентство» Минэнерго России Краснодарский ЦНТИ- филиал ФГБУ «РЭА» Минэнерго России, 2023. – 135 с.
2. Пономарчук, Ю. В. Программирование на языке Java: учебное пособие / Ю. В. Пономарчук, И. В. Кузнецов. – Хабаровск: ДВГУПС, 2021. – 103 с.
3. Java 2019 – The state of Developer Ecosystem in 2019 infographic / JetBrains. – URL: <https://www.jetbrains.com/Ip/devecosystem-2019/java/> (дата обращения: 20.01.2026).
4. Spring Guides / Broadcom. – URL: <https://spring.io/guides> (дата обращения: 20.01.2026).
5. Habr.com | Dependency Injection in Spring, Service Locator [Электронный ресурс] // <https://habr.com/>. – URL: <https://habr.com/ru/all/> (Дата

обращения 19.12.2025). Spring framework [Электронный ресурс] // <https://spring.io/>. – URL: <https://spring.io/> (Дата обращения 19.12.2025).

6. Website with information on Spring annotations, resource injection, and autowire [Электронный ресурс] // <https://www.baeldung.com/>. – URL: <https://www.baeldung.com/spring-annotations-resource-inject-autowire> (Дата обращения 19.12.2025).

7. Кириллова Е.А., Сафина Г.Ф. Преимущества применения Spring Data Jdbc для упрощения работы с базами данных / В сборнике: Актуальные вопросы современной науки и образования. Материалы Всероссийской научно-практической конференции, посвященной 25-летию Нефтекамского филиала УУНиТ. – Уфа, 2025. – С. 75-79.