

# **Бизнес-процессы**

## **БП-1: Регистрация пользователя**

### **Цель:**

Обеспечить новому пользователю возможность создать аккаунт и заполнить профиль с музыкальными предпочтениями для получения персонализированных рекомендаций.

1. Незарегистрированный пользователь открывает форму регистрации.
  2. Пользователь вводит имя, email, пароль и подтверждение пароля.
  3. Система валидирует данные (формат email, требования к паролю, совпадение паролей).
  4. Система проверяет уникальность email в базе данных.
  5. При существующем email система отображает ошибку, процесс завершается.
  6. При уникальном email система создает запись User и перенаправляет на заполнение профиля.
  7. Пользователь заполняет профиль: возраст, музыкальные жанры, инструменты, локация, описание, любимый артист, любимая песня.
  8. Система проверяет заполнение обязательных полей.
  9. При незаполненных полях система отображает ошибку, пользователь возвращается к форме.
  10. При заполненных полях система создает запись Profile и связывает с User.
  11. Система устанавливает `is_active = true` и выполняет автоматический вход.
  12. Система перенаправляет пользователя на главную страницу.
- 

## **БП-2: Аутентификация пользователя**

### **Цель:**

Обеспечить безопасный вход зарегистрированного пользователя в систему с защитой от несанкционированного доступа.

1. Пользователь открывает форму входа.
2. Пользователь вводит email и пароль.
3. Система проверяет заполнение полей.
4. При незаполненных полях система отображает ошибку, процесс завершается.
5. Система ищет пользователя по email в базе данных.
6. При отсутствии пользователя система отображает ошибку, процесс завершается.
7. Система проверяет статус аккаунта (`is_active`).

8. При неактивном аккаунте система отображает ошибку, процесс завершается.
  9. Система проверяет корректность пароля (сравнение хеша).
  10. При неверном пароле система увеличивает счетчик неудачных попыток.
  11. При счетчике  $\geq 5$  система блокирует аккаунт на 15 минут и отображает ошибку, процесс завершается.
  12. При счетчике  $< 5$  система отображает ошибку, процесс завершается.
  13. При верном пароле система сбрасывает счетчик, генерирует JWT токены и сохраняет refresh token.
  14. Система устанавливает токены в cookies/headers и обновляет last\_login.
  15. Система перенаправляет пользователя на главную страницу.
- 

## **БП-3: Управление профилем**

**Цель:**

Обеспечить пользователю возможность редактировать информацию профиля, включая базовые данные, музыкальные предпочтения и фотографии.

1. Пользователь открывает страницу редактирования профиля.
  2. Система загружает текущие данные профиля и отображает форму.
  3. Пользователь изменяет данные (имя, email, возраст, жанры, инструменты, локация, описание, артист, песня).
  4. При изменении email система проверяет уникальность нового email.
  5. При существующем email система отображает ошибку, процесс завершается.
  6. Пользователь может загрузить новую фотографию.
  7. Система валидирует формат (JPEG, PNG, WebP) и размер ( $\leq 10$  МБ).
  8. При невалидных данных система отображает ошибку, процесс завершается.
  9. При валидных данных система сохраняет изображение в File Storage и создает запись UserImage.
  10. Пользователь может удалить фотографию или установить главную.
  11. Система валидирует все изменения.
  12. При невалидных данных система отображает ошибки, процесс завершается.
  13. При валидных данных система сохраняет изменения в базе данных.
  14. Система отображает сообщение об успешном обновлении и обновляет отображение профиля.
- 

## **БП-4: Просмотр колоды профилей и свайп**

**Цель:**

Обеспечить пользователю возможность просматривать рекомендованные профили и выражать интерес через механизм свайпа.

1. Пользователь открывает колоду профилей.
  2. Система получает рекомендации: анализирует профиль пользователя, историю свайпов и вычисляет совместимость.
  3. Система формирует список рекомендованных профилей и проверяет наличие рекомендаций.
  4. При отсутствии рекомендаций процесс завершается.
  5. Система отображает профиль из колоды.
  6. Пользователь выполняет свайп вправо (лайк) или влево (дизлайк).
  7. Система сохраняет решение в базе данных (создает/обновляет запись Swipe).
  8. При лайке система проверяет наличие обратного свайпа от другого пользователя.
  9. При взаимном лайке система создает запись Match и Chat, отправляет уведомления и отображает экран совпадения.
  10. Система переходит к следующему профилю в колоде.
  11. При наличии профилей процесс возвращается к шагу 5.
  12. При отсутствии профилей система загружает новые рекомендации, процесс возвращается к шагу 2.
- 

## БП-5: Обмен сообщениями в чате

**Цель:**

Обеспечить пользователям возможность обмениваться сообщениями в рамках активного мэтча.

1. Пользователь открывает чат с совпавшим пользователем.
2. Система проверяет наличие активного мэтча между пользователями.
3. При отсутствии мэтча система отображает ошибку, процесс завершается.
4. Система загружает историю сообщений из базы данных и отображает пользователю.
5. Пользователь вводит текст сообщения и отправляет.
6. Система создает запись Message в базе данных.
7. Система обновляет запись Chat (lastMessageId, lastMessageAt).
8. Система отправляет сообщение получателю через WebSocket или push-уведомление.
9. Система обновляет отображение чата для отправителя.
10. При открытии чата получателем система помечает сообщения как прочитанные и обновляет счетчик непрочитанных.

---

## **БП-6: Получение рекомендаций**

### **Цель:**

Генерировать персонализированные рекомендации профилей других пользователей на основе музыкальных предпочтений и истории взаимодействий.

1. Пользователь запрашивает рекомендации (явно или при открытии колоды).
2. Система собирает данные: профиль пользователя, историю свайпов, существующие мэтчи.
3. Система анализирует профили других пользователей, исключая просмотренные, мэтчи и неактивных.
4. Система вычисляет совместимость для каждого профиля (совпадение жанров, инструментов, артистов, песен, локация, возраст).
5. Система сортирует профили по убыванию совместимости.
6. Система формирует список рекомендаций (колоду).
7. Система кэширует рекомендации в Redis для быстрого доступа.
8. Система возвращает список рекомендаций пользователю.