**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ФАКУЛЬТЕТ РАДИОФИЗИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ**

**Кафедра телекоммуникаций и информационных технологий**

**РАЗРАБОТКА СИСТЕМЫ ДЛЯ АНАЛИЗА РОДИНОК**

Курсовой проект

Игнатенко Анастасии Ивановны

студентки 2 курса 9 группы, специальность «Прикладная информатика»

Научный руководитель:

ассистент кафедры телекоммуникаций и информационных технологий  
Афоненко Анатолий Александрович

Минск, 2025

# Техническое задание

Необходимо разработать веб-приложение для проверки новообразований кожи: определения риска того, что родинка является меланомой. Функциональные требования к приложению:

1. Пользователю предоставляется возможность пройти регистрацию или войти в существующий аккаунт;
2. Пользователь может просмотреть и изменить данные аккаунта:

* имя;
* пол;
* дата рождения.

1. Пользователь может загрузить файл с родинкой, выбрать местоположение родинки и получить результат: вероятность того, что родинка является злокачественной;
2. Сохраняется история каждой совершенной пользователем проверке, включающая:

* фотографию родинки;
* дату проверки;
* уровень риска меланомы;
* местоположение родинки.

Исходя из функциональных требований должны быть реализованы следующие процессы:

**Авторизация пользователя.** При запуске приложения пользователь попадает на страницу авторизации, где в веб форму должен ввести свой логин и пароль и нажать кнопку «Войти». Если пользователь с таким логином и паролем существует, то пользователь будет перенаправлен на главную страницу приложения. Иначе будет отображено сообщение «Неверный логин!» или «Неверный пароль!», в зависимости от того, какие данные были введены некорректно. Если у пользователя нет аккаунта, то он должен нажать на гиперссылку «Зарегистрироваться», после чего будет перенаправлен на страницу регистрации.

**Регистрации пользователя.** Пользователь заполняет свои данные на веб форме и нажимает кнопку «Зарегистрироваться». Если регистрация прошла успешно, то пользователь будет перенаправлен на главную страницу приложения. Если пользователь введет логин, который уже занят другим пользователем, то будет отображено сообщение «Логин уже существует!».

**Отображение данных аккаунта.** На главной странице приложения пользователь может нажать на гиперссылку со своим именем, после чего будет перенаправлен на страницу, которая содержит данные аккаунта пользователя, гиперссылку «Изменить данные», при нажатии на которую пользователь будет перенаправлен на страницу, где он может изменить данные аккаунта, и гиперссылку «Назад», при нажатии на которую пользователь будет перенаправлен на главную страницу приложения.

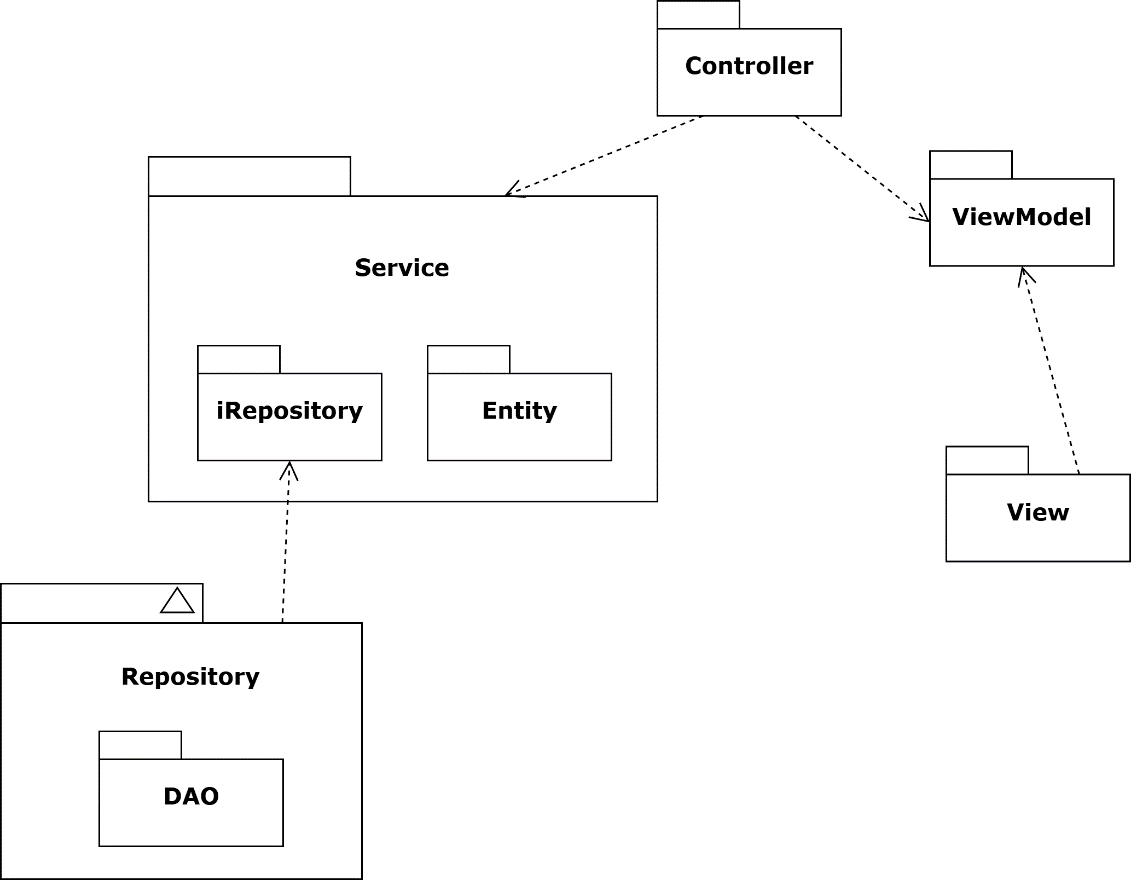
**Изменение данных аккаунта.** Пользователь может изменить данные в веб форме и нажать кнопку «Сохранить изменения», после чего будет перенаправлен на страницу с данными аккаунта. Также пользователь может попасть на страницу с данными аккаунта не изменяя данные аккаунта, нажав на гиперссылку «Назад».

**Просмотр истории проверок.** На главной странице отображен список всех совершенных пользователем проверок. О каждой проверке отображается следующая информация: фото родинки, риск злокачественности родинки. Также под каждой проверкой есть кнопка «Показать дополнительную информацию», после нажатия на которую пользователь попадет на страницу, содержащую полную информацию о проверке, кнопку «Удалить проверку», при нажатии на которую проверка будет удалена, а пользователь будет перенаправлен на главную страницу приложения, и гиперссылку «Вернуться на главную страницу»,при нажатии на которую пользователь будет перенаправлен на главную страницу приложения.

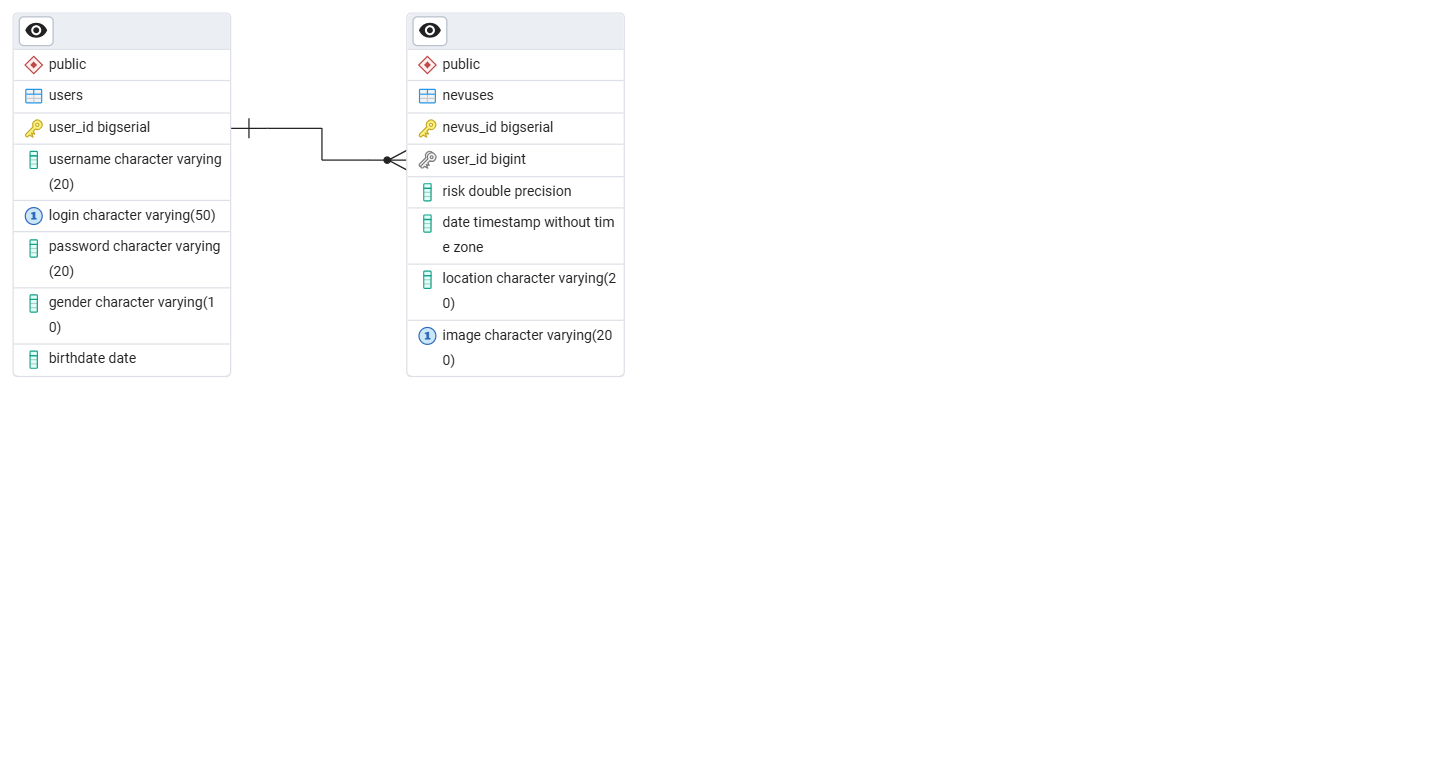
**Проверка родинки.** На главной странице приложения пользователь может нажать на кнопку «Выполнить проверку» после чего будет перенаправлен на страницу, на которой пользователь может загрузить файл с родинкой и выбрать местоположение родинки из списка и нажать кнопку «Выполнить проверку», после чего пользователь будет перенаправлен на страницу, содержащую результат проверки: риск того, что родинка является меланомой. Пользователь может вернуться на главную страницу приложения не выполнив проверку, нажав на гиперссылку «Отмена».

# Архитектура приложения

На рисунке 1 представлена диаграмма связи компонентов, согласующаяся с структурой исходного кода (java packeges).



*Рис. 1 Диаграмма связи компонентов*

**В приложении используется база данных, физическая модель которой представлена на рисунке 2.

*Рис. 2 Физическая модель базы данных*

🧩 Логика работы компонентов

1. View (представление)

Компонент View реализован с использованием JSP-страниц, расположенных в директории webapp/WEB-INF.

JSP-страницы отвечают за отображение данных пользователю. Каждая страница связана с соответствующей моделью представления (ViewModel), которая содержит все необходимые данные для подстановки в шаблон.

2. Controller (контроллеры)

Контроллеры находятся в пакете Controller. Каждый контроллер обрабатывает определённый маршрут (URL) и выполняет следующие действия:

Принимает HTTP-запрос (GET/POST)

Обрабатывает входные данные

Вызывает методы соответствующего сервиса

Формирует ViewModel и передаёт его в JSP для отображения

Пример:

Контроллер Register принимает данные формы регистрации, вызывает UserService для создания пользователя, и в зависимости от результата либо перенаправляет на страницу входа, либо отображает сообщение об ошибке.

3. Service (бизнес-логика)

Сервисы (UserService, CheckService, AuthService) инкапсулируют бизнес-логику приложения.

Они взаимодействуют с репозиториями (iRepository) и отвечают за валидацию, хранение и обработку данных.

4. Repository и DAO (доступ к данным)

Для доступа к данным используется двухуровневый подход:

Интерфейсы в пакете iRepository определяют контракт (например, iUserRepository)

Реализации находятся в Repository.DAO (UserRepository, MoleCheckRepository) и используют подключение к базе данных, предоставляемое классом Database.

5. Entity (модели данных)

Сущности (UserEntity, MoleCheckEntity) представляют таблицы базы данных. Они используются в слоях Service и Repository.

6. ViewModel (модели представления)

Каждая страница получает данные через соответствующий ViewModel, расположенный в пакете ViewModel. Эти модели передаются от контроллера к JSP-страницам.