МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«Российский экономический университет им. Г.В. Плеханова»

Московский приборостроительный техникум

**Курсовой проект**

ПМ 01 Разработка модулей программного обеспечения для компьютерных систем

МДК 01.01 Разработка программных модулей

Специальность 09.02.07 «Информационные системы и программирование»

Квалификация: Программист

Тема: «Разработка программного комплекса для криптовалютного обучения и новостных статей»

**Техническое задание**

Листов: 26

|  | Руководитель  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ / А.А. Шимбирёв  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 год  Исполнитель  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ / С.А. Мельников  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 год |
| --- | --- |

2025

АННОТАЦИЯ

В данном программном документе приведено техническое задание на разработку программного комплекса для детского нейроцентра «Природа разума»

В данном программном документе, в разделе «Введение» указано наименование, краткая характеристика области применения программы.

В разделе «Основания для разработки» указаны документы, на основании которых ведется разработка, наименование и условное обозначение темы разработки.

В разделе «Назначение разработки» указано функциональное и эксплуатационное назначение программы (программного изделия).

Раздел «Требования к программе» содержит следующие подразделы:

* требования к функциональным характеристикам;
* требования к надежности;
* условия эксплуатации;
* требования к составу и параметрам технических средств;
* требования к информационной и программной совместимости;
* специальные требования.

В разделе «Требования к программной документации» указаны предварительный состав программной документации и специальные требования к ней.

В разделе «Технико-экономические показатели» указаны: ориентировочная экономическая эффективность, предполагаемая годовая потребность.

В разделе «Стадии и этапы разработки» установлены необходимые стадии разработки, этапы и содержание работ.

В разделе «Порядок контроля и приемки» должны быть указаны виды испытаний и общие требования к приемке работы.

СОДЕРЖАНИЕ

[1. ВВЕДЕНИЕ 8](#_Toc168459449)

[1.1. Наименование программы 8](#_Toc168459450)

[1.2. Краткая характеристика области применения программы 8](#_Toc168459451)

[2. ОСНОВАНИЕ ДЛЯ РАЗРАБОТКИ 8](#_Toc168459452)

[2.1. Основание для проведения разработки 8](#_Toc168459453)

[2.2. Наименование и условное обозначение для разработки 8](#_Toc168459454)

[3. НАЗНАЧЕНИЕ РАЗРАБОТКИ 8](#_Toc168459455)

[3.1. Функциональное назначение программы 8](#_Toc168459456)

[3.2. Эксплуатационное назначение программы 8](#_Toc168459457)

[4. ТРЕБОВАНИЯ К ПРОГРАММЕ 8](#_Toc168459458)

[4.1. Требования к функциональным характеристикам 8](#_Toc168459459)

[Данная программа должна иметь следующий функционал: 8](#_Toc168459460)

[ Просмотр, удаление, изменение и создание Статей внутри Базы Данных 9](#_Toc168459461)

[ Просмотр, удаление, изменение и создание Тэгов внутри Базы Данных 9](#_Toc168459462)

[ Просмотр, удаление, изменение и создание Пользователей Телеграмм Бота внутри Базы Данных 9](#_Toc168459463)

[ Просмотр, удаление, изменение и создание Ролей внутри Базы Данных 9](#_Toc168459464)

[ Вывод текущей стоимости Биткоина на главной странице сайта Target not f](#_Toc168459465)

[ Отдельные страницы с уникальным дизайном для регистрации и авторизации 9](#_Toc168459466)

[ Возможность создания и редактирования статей прямо на сайте 9](#_Toc168459467)

[ Фильтрация статей по выбранному Тэгу Target not f](#_Toc168459468)

[ Возможность просмотра всех Статей на отдельной странице сайта Target not f](#_Toc168459469)

[ Возможность просмотра всех Тэгов на отдельной странице сайта Target not f](#_Toc168459470)

[ API – Возможность создания, изменения, удаления и получения данных с помощью API, реализованной по стандарту GraphQL. Программист может обратиться к API с помощью запроса на эндпоинт и получить как и полноценные статьи, так и лишь часть информации из статей – только название, только дату публикации, только содержание, только время на прочтение, только превью, только дату создания, только тэги, только краткое описание или же только дату публикации, также можно реализовать запрос таким образом, что можно получить несколько аргументов, например: название, тэги и краткое описание статей. Также можно сделать и с тэгами: получить лишь названия тэгов или же только описания. 10](#_Toc168459471)

[ Возможность просмотра команд для обращения к API на отдельной странице сайта 10](#_Toc168459472)

[ Возможность просмотра информации о собственной учетной записи в личном кабинете на отдельной странице сайта 10](#_Toc168459473)

[ Возможность включения и выключения уведомлений в личном кабинете и в Телеграмм боте Target not f](#_Toc168459474)

[ Уведомления на сайте: пользователь может получать уведомления на почту, уведомления должны иметь следующее содержание: название статьи, краткое описание, дата публикации для телеграмм бота и время на прочтения в обоих случаях. Пользователь может включать или выключать уведомления в личном кабинете на сайте. Также уведомления должны содержать ссылку на новую статью Target not f](#_Toc168459475)

[ Возможность просмотра всех статей в мобильном приложении Target not f](#_Toc168459476)

[ Возможность просмотра всех тэгов в мобильном приложении Target not f](#_Toc168459477)

[ Возможность добавление криптовалюты в избранное: в Телеграмм боте пользователь может добавить криптовалюту в избранные криптовалюты: в главном меню необходимо нажать на кнопку «Выбрать криптовалюты для наблюдения», после нажатия выведется меню, в котором можно будет перемещаться по страницам. Всего выводится по 10 криптовалют на страницу, также всего должно быть всего 5 страниц. В меню выводятся топ 50 криптовалют с сайта CoinGecko. При добавлении криптовалюты – нажатия на неё в Избранных криптовалютах появляется кнопка с название выбранной монеты. Также должны быть возможность вернуться обратно по нажатию на кнопку «Вернуться обратно». Target not f](#_Toc168459478)

[ Избранные криптовалюты: в Телеграмм боте должна быть возможность перехода в меню избранные криптовалюты, для этого необходимо в главном меню нажать на кнопку «Избранные криптовалюты». При пустом листе избранных криптовалют пользователь получит сообщение «Ваш список избранных криптовалют пуст», которое удалится через секунду после отправки. В другом случае в меню с избранными криптовалютами должны быть выведены кнопки с избранными монетами, при нажатии на кнопку с активом появляется подробная информация о криптовалюте. Target not f](#_Toc168459479)

[ Уведомления в Телеграмм боте: в основном меню пользователь имеет возможность перейти в меню для включения или же выключения уведомлений о новых статьях по нажатию на кнопку «Уведомлять о новых статьях?». В меню должно быть 2 кнопки – «Да» и «Нет». В соответствии от кнопки будут включены или выключены уведомления. Есть возможность выхода из данного меню по нажатию на кнопку «Вернуться обратно». При включенных уведомлениях пользователь будет получать сообщение со следующим содержимым: Название статьи, время на прочтение, краткое описание, ссылка на статью. Target not f](#_Toc168459480)

[ Вывод FAQ – наиболее часто – задаваемых вопросов на отдельной странице в мобильном приложении. Target not f](#_Toc168459481)

[ Фильтрация статей по отдельному тэгу в мобильном приложении Target not f](#_Toc168459482)

[ Регистрация – пользователь на отдельной странице сайта может зарегистрироваться, введя логин, электронную почту и пароль с подтверждением. При успешной регистрации пользователь должен быть перенаправлен в личный кабинет, в другом случае выведется причина отказа в регистрации. Target not f](#_Toc168459483)

[ Авторизация – пользователь на отдельной странице сайта может авторизоваться в систему, введя логин и пароль. При неудачном процессе аунтефикации выведется причина. 11](#_Toc168459484)

[ Избранные криптовалюты: в Телеграмм боте по нажатию на кнопку «Избранные криптовалюты» пользователь должен перейти в меню с избранными криптовалютами. Если пользователь до данного момента не выбрал ни одну криптовалюту для наблюдения, то выведется сообщение о том, что список избранных криптовалют пуст. Если же ранее пользователь выбрал криптовалюты – то ему откроется меню с ранее выбранными криптовалютами. В данном меню будет выведена кнопка для каждого актива, при нажатии на кнопку с соответствующим активом пользователю выведется подробная информация о данной монете с многими её метриками, которые будут получены при обращении к CoinGecko API. Будет выведена следующая информация: цена на данный момент, текущая капитализация, ATH – максимальная цена за последние 24 часа, ATL – максимально низкая цена за последние 24 часа, ATH за всё время, а также процент в изменении в цене за последние 24 часа. Также есть возможность выхода из данного меню по нажатию на кнопку «Вернуться обратно». 15](#_Toc168459485)

[4.2. Требования к надежности 19](#_Toc168459486)

[4.3. Условия эксплуатации 20](#_Toc168459487)

[4.4. Требования к составу и параметрам технических средств 20](#_Toc168459488)

[4.5. Требования к информационной и программной совместимости 20](#_Toc168459489)

[4.6. Специальные требования 21](#_Toc168459490)

[1. Требования к контролю версий 21](#_Toc168459491)

[4. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ 21](#_Toc168459492)

[a. Предварительный состав программной документации 21](#_Toc168459493)

[b. Специальные требования к программной документации 22](#_Toc168459494)

[5. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ 23](#_Toc168459495)

[a. Ориентировочная экономическая эффективность 23](#_Toc168459496)

[b. Предполагаемая годовая потребность 23](#_Toc168459497)

[c. Экономические преимущества разработки 23](#_Toc168459498)

[6. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ 23](#_Toc168459499)

[a. Стадии разработки 23](#_Toc168459500)

[b. Этапы разработки 23](#_Toc168459501)

[c. Содержание работ по этапам 23](#_Toc168459502)

[d. Исполнители 25](#_Toc168459503)

[7. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ 25](#_Toc168459504)

[a. Порядок контроля 25](#_Toc168459505)

[b. Порядок приемки 25](#_Toc168459506)

## ВВЕДЕНИЕ

### Наименование программы

Наименование - «Программный комплекс для детского нейроцентра «Природа разума»».

### Краткая характеристика области применения программы

Приложение предназначено к применению в детском нейроцентре.

## ОСНОВАНИЕ ДЛЯ РАЗРАБОТКИ

### Основание для проведения разработки

Основанием для проведения разработки является задание по ПМ 01 Разработка модулей программного обеспечения для компьютерных систем и МДК 01.01 Разработка программных модулей

### Наименование и условное обозначение для разработки

Наименование темы разработки - «Цифровая платформа для автоматизации работы с документами».

## НАЗНАЧЕНИЕ РАЗРАБОТКИ

### Функциональное назначение программы

Функциональным назначением программы является автоматизация работы с документами

### Эксплуатационное назначение программы

Программа должна эксплуатироваться в компании “Природа разума”.

Конечным пользователем программы будет Администратор и Педагог.

## ТРЕБОВАНИЯ К ПРОГРАММЕ

### Требования к функциональным характеристикам

### Данная программа должна иметь следующий функционал:

* Просмотр, удаление, изменение и создание Пользователей внутри Базы Данных

### Просмотр, удаление, изменение и создание Ролей внутри Базы Данных

### Просмотр, удаление, изменение и создание Пациентов внутри Базы Данных

### Просмотр, удаление, изменение и создание Документов внутри Базы Данных

### Отдельная страница с уникальным дизайном для авторизации

### Отдельная страница с уникальным дизайном для просмотра статистики

### Отдельная страница с уникальным дизайном для личного кабинета

### Страница для работы с каждой моделью на сайте

### Страница для работы с каждой моделью в мобильном приложении

### Возможность поиска всех моделей на сайте

### Возможность поиска всех моделей в мобильном приложении

### Возможность фильтрации Пользователей по статусу на сайте

### Возможность фильтрации Пользователей по ролям на сайте

### Возможность фильтрации Пользователей по статусу в мобильном приложении

### Возможность фильтрации Пользователей по ролям в мобильном приложении

### Возможность фильтрации Документов по пациентам на сайте

### Возможность фильтрации Документов по директориям на сайте

### Возможность фильтрации Документов по пациентам в мобильном приложении

### Возможность фильтрации Документов по директориям в мобильном приложении

### API – Возможность создания, изменения, удаления и получения данных с помощью RESTful API. В API должен быть реализован следующий функционал: создание, изменение, удаление, получение по ID и получение всех моделей в базе данных, а также авторизация и обновление JWT токена, получение статуса запланированной задачи по её ID, получение статистики по дням о следующих моделях: документы, пользователи и пациенты. Получение статистики по директориям о документах и получение статистики по ролям о пользователях, выгрузка любой статистики в формат .csv или .xlsx. Также для модели «Документы» должна быть возможность скачивания документа в исходном формате, для модели «Пользователи» должна быть возможность получение, загрузки и удаление фотографии личного кабинета.

### Возможность просмотра документации к API как минимум в двух форматах - Swagger и Redoc

### Возможность доступа к базе данных при помощи PgAdmin4

### Возможность изменения пароля: пользователь может изменить пароль в личном кабинете на сайте и в мобильном приложении, для этого он должен ввести старый пароль, а также ввести новый пароль два раза. После этого необходимо нажать на кнопку «Изменить пароль».

### Возможность просмотра информации о собственной учетной записи в личном кабинете на сайте

### Возможность просмотра информации о собственной учетной записи в личном кабинете в мобильном приложении

### Возможность работы с документами: пользователь имеет возможность редактирования - создания, изменения, удаления, а также просмотра документов. Также пользователь имеет возможность скачивание документа. Пользователь может работать с данном моделью как и в мобильном приложении, так и на сайте

### Возможность работы с пациентами: пользователь имеет возможность редактирования - создания, изменения, удаления, а также просмотра документов. Пользователь может работать с данном моделью как и в мобильном приложении, так и на сайте

### Возможность работы с пользователями системы: пользователь имеет возможность редактирования - создания, изменения, удаления, а также просмотра пользователей системы, если у него есть роль «Администратор». Пользователь может работать с данном моделью как и в мобильном приложении, так и на сайте

### Возможность работы с ролями: пользователь имеет возможность редактирования - создания, изменения, удаления, а также просмотра ролей, если у него есть роль «Администратор». Пользователь может работать с данном моделью как и в мобильном приложении, так и на сайте

### Реализация Drag’n’drop на сайте для загрузки документов –Пользователи могут загружать файлы напрямую в приложение, перетаскивая их из проводника компьютера или с рабочего стола в специально обозначенную зону интерфейса - всю область открытого браузера

### Авторизация – пользователь на отдельной странице сайта может авторизоваться в систему, введя логин и пароль. При неудачной попытке входа выведется причина или ошибка

### Выход из системы – пользователь может выйти из системы и авторизоваться заново, для этого ему необходимо нажать на кнопку в боковом меню или же личном кабинете.

### Изменения фотографии личного кабинета – пользователь в личном кабинете может изменить фотографию, для этого ему необходимо нажать на собственный аватар и загрузить новую фотографию

#### Требования к таблицам базы данных

#### База данных должна иметь 1 НФ и в ней должно быть как минимум 4 Таблицы: Пользователи, Роли, Пациенты и Документы.

#### Таблицы Роли и Пользователи должны быть соединены между собой, также таблицы Пользователи и Документы, Документы и Пациенты также должны быть соединины между собой

#### Все Таблицы по умолчанию должны содержать поля с created\_at - дата создания модели, и updated\_at - дата обновления модели

#### Все таблицы должны иметь названия в стиле sneak case - все буквы в назваии должны иметь нижний регистр, а также необходимые слова должны быть разъеденены нижними подчёркиваниями

#### Требования к таблице Пользователи в базе данных

#### В данной таблице должны реализовываться следующие поля: ФИО, логин, пароль, электронная почта, роль, статус активности, ссылка на фотографию личного кабинета, документы, прикреплённые к нему, а также Уникальный Идентификатор.

#### Роль должна браться из ранее созданных ролей в таблице «Роли» с помощью связи «Один ко многим»

#### ФИО должно иметь тип данных varchar, а также иметь длину от 3 до 255 символов и содержать только латиницу. Оно обязательно должно быть заполнено

#### Логин должен иметь тип данных varchar, а также состоять только из латиницы и цифр, а также он должен иметь длину от 5 до 50 символов. Он обязательно должен быть заполнен

#### Электронная почта должна иметь тип данных varchar, а также состоять только из латиницы, цифр и специального знака «Собака», а также он должен иметь длину от 5 до 255 символов

#### Пароль должен храниться в виде сырых байтов, а также иметь длину равную 60 байтам - 60 символам. Он обязательно должен быть заполнен.

#### Статус активности должен иметь тип данных булево, по умолчанию должен быть равен единице, или же быть истинным. Он обязательно должен быть заполнен

#### Ссылка на фотографию личного кабинета должна иметь тип данных varchar

#### Поля «Ссылка на фотографию лично кабинета», «Электронная почта», «Документы» могут быть пустыми.

#### Требования к таблице Роли

#### В данной таблице должны реализовываться следующие поля: Название, Описание, Уникальный идентификатор

#### Название должно иметь тип данных varchar, его длина должна быть от 3 до 255 символов. Название может состоять только из Латиницы, Кириллицы, цифр, пробелов и дефисов. Оно обязательно должно быть заполнено.

#### Описание должно иметь тип данных varchar, его длина должна быть от 0 до 1000 символов. Оно может содержать цифры, латиницу, кириллицу и специальные символы.

#### Поле «Описание» может быть пустым.

#### Требования к таблице Пациенты в базе данных

#### В данной таблице должны быть реализованы следующие поля:

#### ФИО, Дата рождения и документы, прикреплённый к нему.

#### ФИО должно иметь тип данных varchar, а также иметь длину от 3 до 255 символов и содержать только латиницу. Оно обязательно должно быть заполнено.

#### Требования к таблице Документы в базе данных

#### В данной таблице должны быть реализованы поле Название, Данные, Тип поддиректории, ID Документа, ID Пациента.

#### Поле название должно содержать до 255 символов и иметь тип данных varchar. Оно обязательно должно быть заполнено.

#### Поле данные должен храниться в виде сырых байтов. Они обяазтельно должно быть заполнены.

#### Между документами и пользователями должна быть организована связь «Один ко многим»

#### Между документами и пациентами должна быть организована связь «Один ко многим»

#### Требования к функциям

#### В программе должны быть реализованы следующие функции:

#### Вывод данных из таблиц: в зависимости от роли сотруднику должна быть предоставлена возможность просмотра всех данных из всех таблиц

#### Поиск ролей: сотруднику должна быть представлена возможность поиска ролей по их названию. Для поиска необходимо ввести необходимый запрос в поле, после чего выведется результат.

#### Поиск документов: сотруднику должна быть представлена возможность поиска ролей по их названию. Для поиска необходимо ввести необходимый запрос в поле, после чего выведется результат.

#### Поиск детей: сотруднику должна быть представлена возможность поиска детей по их ФИО. Для поиска необходимо ввести необходимый запрос в поле, после чего выведется результат.

#### Поиск пользователей: сотруднику должна быть представлена возможность поиска пользователей по их ФИО. Для поиска необходимо ввести необходимый запрос в поле, после чего выведется результат.

#### Фильтрация пользователей: сотруднику должна быть предоставлена возможность фильтрации пользователей по двум параметрам: статус активности и роль. Для фильтрации независимо от платформы необходимо нажать на выпадающее меню и выбрать необходимый параметр

#### Фильтрация документов: сотруднику должна быть предоставлена возможность фильтрации документов по двум параметрам: ребёнок, к которому привязан документ и директория. Для фильтрации независимо от платформы необходимо нажать на выпадающее меню и выбрать необходимый параметр

#### Авторизация: в окне с авторизацией должна быть возможность авторизоваться в систему при помощи ввода логина и пароля и нажатия на кнопку «Войти». При неправильном вводе данных под соответствующим полем выведется текст с причиной о отказе авторизации. При успешной авторизации пользователь переходит в личный кабинет.

#### Удаление, Создание, Изменение и просмотр всех таблиц: в панели управления сайта сотрудник в зависимости от роли может реализовать все данные функции с определёнными таблицами - Администратор может работать со всеми таблицами в системе, а Педагог может работать только с таблицами «Пациенты» и «Документы».

#### Изменение пароля: в личном кабинете должна быть предоставлена возможность изменения пароля, для этого необходимо нажать на кнопку «Изменить пароль», после чего откроется модальное окно, в котором необходимо один раз ввести старый пароль и два раза ввести новый пароль, после чего необходимо нажать на кнопку «Изменить пароль»

#### Загрузка фотографии для профиля: в личном кабинете должна быть предоставлена возможность загрузки фотографии для профиля, для этого необходимо нажать на иконку в личном кабинете и выбрать необходимую фотографию

#### Экспорт статистики: на странице «Статистика» администратору должна быть предоставлена возможность выгрузки статистики в 2 форматах: CVS и Excel - .xlsx. Для выгрузки статистики ему необходимо выбрать график, временной промежут, далее нажать на кнопку «Экспорт» и выбрать необходимый формат

#### Страницы веб-сайта должны быть реализовано след. Образом:

#### Каждая страница, кроме страниц для авторизации и регистрации должны иметь нижнее и верхнее меню со всеми ссылками на сайте

#### Каждая страница, кроме страниц для регистрации и авторизации должны иметь белый фон и быть адаптивными

#### Требования к ролям

#### В программе изначально должна быть 2 роли – «Администратор» и «Педагог».

#### Требования к планировщику задач

#### Планировщик задач должен быть реализован при помощи Celery

#### Планировщик задач должен иметь в зависимостях брокер сообщений, которым должен выступать Redis или RabbitMQ.

#### Планировщик задач должен планировать задачи не реже одной недели и не чаще одного часа

#### Планировщик задач должен отвечать за бэкап базы данных, который должен происходить при старте приложения, а также каждый день после инициализации.

#### Требования к контейнеризации

#### Клиент - серверное приложение - сайт, а также база данных и такие сервисы, как pgAdmin4, Redis, nginx и RabbitMQ должны быть контейнеризированы при помощи сервиса Docker.

#### Требования к инициализации базы данных

#### При сборке программы должна быть реализована логика для изначального применения миграций для сохранения актуальности базы данных

#### При изначальном запуске программы должно быть создано 2 роли - «Администратор» и «Педагог», а также должен быть создан пользователь с ролью «Администратор» со следующими данными: ФИО - Фамилия Имя Отчество, статус активности - истинно, логин - admin123, пароль - admin123

#### Требования к мобильному приложению:

#### Мобильное приложение должно быть реализовано при помощи языка Kotlin.

#### Мобильное приложение должно быть реализовано при помощи фреймворка Kotlin Multiplatform и поддерживать как минимум одну операционную систему

#### Мобильное приложение можно скачать при помощи сканирования QR-кода.

#### Расширение мобильного приложения должно являться .apk или .ipa

#### Требования к переменным

#### Переменные должны иметь соответствующий тип доступа в соответствии с их назначением и положением в иерархии программы.

#### Переменные должны быть описаны комментариями если их название не очень ясно передает предназначение данной переменной

#### Требования к функциям приложения

#### Функции должны быть семантически названы с помощью латиницы, заглавных и строчных букв и цифр.

#### Каждая функция должна иметь название, отражающая функционал, которая она содержит

#### Каждая функция в зависимости от ее расположения и использования должна иметь свой уровень доступа – public, private или protected.

#### Стиль названия функции должен зависеть от языка и фреймворка, в котором она реализована

#### Требования к потоку выполнения функций и методов

#### Сетевые запросы и неконтролируемые Input / Output запросы должны быть реализованы асинхронно или параллельно

#### Требования к защите хранилищ данных

#### Все хранилища данных должны быть защищены как минимум при помощи базовых методов аутентификации - логина и пароля

#### Требования к выводу данных о Пользователях

#### Пароль при вводе для авторизации или его изменения должен быть закрыт точками. При этом должна быть реализована возможность его просмотра по нажатию на специальную кнопку

#### Пароль, как данные не должен никуда выводиться ни в каком виде

#### Поля для данных должны быть организованы в соответствии с типом данного столбца

#### Требования к подключению базы данных

#### База данных должна быть полностью реализована и иметь исправную и рабочую логику при подключении

#### Требования к организации входных данных

#### Все входные данные должны проходить валидацию в зависимости от их допустимых значений и типов данных в базе данных

#### Все файлы должны проверяться на соответствующие расширения – фотографии должны иметь лишь тип данных .jpg, .jpeg, или .png, видео могут иметь лишь тип данных .mp4, .mov или .mkv. Документы в системе могут храниться только в формате .pdf или .docx

#### Валидация для всех данных должна быть реализована на трёх уровнях: клиент, сервер и база данных.

#### При вводе электронная почта должна иметь тип [название@доменное.имя](mailto:название@доменное.имя).

#### Требования к API

#### Логика для API должно реализовываться в отдельном файле

#### API должен быть доступен только пользователям, которые есть в системе и имеет статус активности - Истинный.

#### API должно иметь тип RESTful

#### Требования к документации API

#### API должно иметь как минимум 1 вид документации - Swagger

#### API должно описать все существующие эндпоинты как минимум 1 словом

#### Документация к API должна быть общедоступной

#### В документации должны быть приведены все возможные статус - коды, которые могут придти в ответе от сервера

#### Требования к хосту

#### Сервер хоста должен иметь Unix - подобную операционную систему

#### Операционная система должна поддерживать все необходимые технологии для развёртывания приложения

#### Требования к адаптивной верстке окон

#### Адаптивная верстка должна позволять расширять окно на всю ширину экрана

#### При уменьшении или увеличении окна обязательно должны сохраняться кнопки для закрытия, скрытия или увеличения окна во весь экран

#### Адаптивная верстка должна быть реализована на всех окнах

#### Требования к веб - серверу:

#### Веб - сервер для клиента должен быть реализован при помощи nginx

#### Веб - сервер для серверной части приложения должен быть реализован при помощи uvicorn

#### Требования к используемым технологиям

#### Для реализации клиент - серверного приложения необходимо использовать следующие фрейморки и библиотеки: Клиент - язык программирования: Typescript, Фреймворк: React, Локальный сервер - Vite, библиотека стилей: Bootstrap. Серверная часть - язык: Python, фреймворк - FastAPI, ORM - SQLAlchemy, инструмент для миграций - Alembic, сервис для кэширования - Redis, линтер - Black, форматтер - mypy, библиотека для валидации - Pydantic

#### Для реализации мобильного приложения необходимо использовать технологию Kotlin Multiplatform и язык программирования Kotlin

#### Требования к библиотекам для разработки

#### Библиотеки должны быть общедоступны и желательно иметь открытый код

#### Должна быть возможность закачки библиотек при помощи наиболее популярных и нативных менеджеров пакетов – для python: pip, для Kotlin: gradle, при использовании Android Studio: установка с помощью добавления имплементации в Gradle, для Typescript - npm.

#### Требования к названиям файлов

#### Файлы должны иметь семантические названия, состоящие из латиницы, заглавных и строчных букв.

#### Файлы с версткой для вебсайта должны иметь название в соответствии с содержанием страницы

#### Файлы, содержащие реализацию интерфейса или класса должны начинаться с большой буквы

#### Требования к хранению личных данных пользователей:

#### Все пароли должны быть захэшированы и засолены

### Требования к надежности

#### Требования к обеспечению надежного (устойчивого) функционирования программы

Надежное (устойчивое) функционирование программы должно быть обеспечено организацией бесперебойного питания технических средств.

#### Время восстановления после отказа

Время восстановления после отказа, вызванного сбоем электропитания технических средств не должно превышать времени, необходимого на перезагрузку операционной системы и запуск программы, при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

#### Отказы из-за некорректных действий оператора

Отказы программы возможны вследствие некорректных действий пользователя при взаимодействии с операционной и информационной системой.

Во избежание возникновения отказов программы по указанной выше причине следует провести инструктаж с конечным пользователем о правильном эксплуатации информационной системы и убедиться в том, что конечный пользователь имеет опыт использования ЭВМ.

### Условия эксплуатации

#### Климатические условия эксплуатации

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации.

#### Требования к видам обслуживания

Требования к видам обслуживания не предъявляются.

#### Требования к численности и квалификации персонала

Сотрудники должны иметь базовое умение работы с персональным компьютером и таблицами.

### Требования к составу и параметрам технических средств

В состав технических средств должен входить IBM-совместимый персональный компьютер (ПЭВМ), включающий в себя:

а) процессор с тактовой частотой, 1.6 ГГц, не менее;

б) оперативную память объемом, 8 ГБ, не менее;

в) жесткий диск объемом 50 Гб, и выше;

г) оптический манипулятор типа «мышь»;

### Требования к информационной и программной совместимости

#### Требования к информационным структурам и методам решения

Требования к информационным структурам на входе и выходе, а также к методам решения не предъявляются.

#### Требование к исходному коду и языкам программирования

Исходные коды программы должны быть реализованы на языке Python, Typescript или Kotlin в зависимости от области разработки. В качестве интегрированной среды разработки программного комплекса должны быть использованы PyCharm Professional как минимум 2023 года выпуска или Visual Studio Code с минимальной версией 1.97.0, Android Studio с минимальной версией 2024.2.1 и Docker с минимальной версией 27.2.1.

#### Требования к программным средствам, используемым программой

На компьютере должна быть операционная система Windows 10 или Windows 11, или Windows 11 Pro, или же любая Unix-подобная система, поддерживающая необходимые версии приложений для развёртывания. Также должен быть запущен сервер с базой данных PostgreSQL, и установлены Android Studio 2024.2.1 Ladybug, Visual Studio Code и Docker Desktop.

#### Требования к защите информации и программ

Требования к защите информации и программ не предъявляются.

### Специальные требования

### Требования к контролю версий

#### После завершения разработки весь программный комплекс должен быть запушен на GitHub в отдельный репозиторий

#### Каждый отдельным комплекс должен иметь уникальную директорию

#### Каждый отдельный проект должен иметь файл .gitignore

#### Требования к IDE Visual Studio Code

#### Год выпуск приложения должен быть как минимум 2020 год.

#### IDE должно исправно работать и не иметь внутренних ошибок при компиляции.

#### Требования к подключение сторонних библиотек

#### Библиотека должна быть в открытом доступе

#### Библиотека должна быть скачана в пакетном менеджере Pip, maven или же в другом официальном пакетном менеджере

#### Библиотека не должна скачиваться со сторонних неофициальных и нелицензированных ресурсов.

## ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

### Предварительный состав программной документации

Состав программной документации должен включать в себя:

1. Приложение А. Техническое задание;
2. Приложение Б. Пояснительная записка;
3. Приложение Б.1. Технический проект;
4. Приложение Б.2. Эскизный проект;
5. Приложение В. Сценарий тестовых испытаний;
6. Приложение Г. Результаты тестовых испытаний;
7. Приложение Д. Руководство пользователя;
8. Приложение Е. Текст программы;
9. Приложение Ж. Скрипт базы данных.

### Специальные требования к программной документации

Техническое задание оформлялось по ГОСТам: 19.201-78 и 19.106-78.

Поля макета технической записки:

* правое поле – 20 мм,
* левое поле – 35 мм
* верхнее поле – 25 мм,
* нижнее поле – 20 мм.

Текст должен быть отпечатан машинным способом на принтере на одной стороне листа.

Шрифт должен быть с "засечками". Рекомендуется Times New Roman. Цвет шрифта должен быть черным.

Размер шрифта (его высота в кеглях):

* для основного текста: 14,
* для заголовка 14,
* для таблиц, схем и приложений 12,
* для колонтитулов (к каждому листу П3) 12.

Межстрочный интервал равен:

* для основного текста 1,5;
* для таблиц, схем и приложений 1.

Текст содержит отступ каждого абзаца - 1,25 см.

Основной текст работы (проекта) печатается 1,5 междустрочным интервалом компьютерного набора.

Выравнивание текста устанавливается «По ширине страницы».

## ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

### Ориентировочная экономическая эффективность

Ориентировочная экономическая эффективность не рассчитывается.

### Предполагаемая годовая потребность

Предполагаемая годовая потребность не рассчитывается.

### Экономические преимущества разработки

Экономические преимущества разработки не рассчитываются.

## СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

### Стадии разработки

Разработка должна быть проведена в соответствии со следующими стадиями:

1. Разработка технического задания;
2. Разработка технического проекта;
3. Разработка эскизного проекта;
4. Рабочий проект;
5. Внедрение.

### Этапы разработки

Сроки сдачи этапов разработки формируются исходя из требований заказчика.

Окончательный срок сдачи всего проекта – 06.05.2024.

### Содержание работ по этапам

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

1. Постановка задачи;
2. Определение и уточнение требований к техническим средствам;
3. Определение требований к программе;
4. Определение стадий, этапов и сроков разработки программы и документации на неё;
5. Выбор языков программирования;
6. согласование и утверждение технического задания.

На этапе разработки эскизного проекта должны быть выполнены перечисленные ниже работы:

1. Предварительная разработка входных и выходных данных;
2. Уточнение методов решения задачи;
3. Разработка общего описания алгоритма решения задач;
4. Согласование и утверждения эскизного проекта.

На этапе подготовки готового программного проекта должна быть выполнена работа по подготовке и передаче программы и программной документации в эксплуатацию.

На этапе разработки пояснительной записки должны быть выполнены перечисленные ниже работы:

1. Общая часть;
2. Специальная часть;
3. Технологическая часть;
4. Заключение.

На этапе разработки результатов тестовых испытаний должны быть выполнены перечисленные ниже виды работ:

1. Разработка, согласование и утверждение программы и методики испытаний;
2. Проведение приемо-сдаточных испытаний;
3. Корректировка программы и программной документации по результатам испытаний.

На этапе разработки руководства пользователя должны быть описаны следующие пункты:

1. Описание установки информационной системы;
2. Описание эксплуатации информационной системы;
3. Описание деинсталляции информационной системы.

На этапе подготовки скрипта базы данных и текста программы необходимо выполнить работу по созданию соответствующего приложения.

### Исполнители

Руководитель разработки Фамилия. Е.А.

Исполнитель Мельников. С.А.

## ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

### Порядок контроля

Контроль выполнения работ проводится менеджером проекта и заказчиком проекта в момент завершения каждого этапа /стадии.

Стадии разработки проекта указаны в разделе «7.1 Стадии разработки».

### Порядок приемки

Менеджер проекта передает заказчику техническую документацию и разработанный проект в электронном виде.

Все документы должны быть подписаны исполнителем, менеджером проекта и утверждены заказчиком.

Состав документации определен в разделе 5.1. «Предварительный состав программной документации».