**Авторизация:**

1. Ввод логина и пароля в форму на сайте
2. Передача логина и хэш-пароля на сервер
3. Сервер отправляет запрос БД (с помощью готовых процедур?), передавая логин
4. БД возвращает хэш-пароля на сервер
5. Сервер сравнивает хэш-пароли
6. Если они совпадают:
   1. Сервер передает положительный ответ сайту (+ куки?)
   2. Сайт авторизирует пользователя и переводит его на начальную страницу (личный кабинет/главная страница)
7. Если они не совпадают:
   1. Сервер передает причину ошибки сайту
   2. Сайт выводит причину отказа в понятном для пользователя виде

**Аутентификация:**

1. Ввод регистрационных данных
2. Преобразование данных в запрос на сервер через JSON/XML
3. Сервер парсит данные и формирует их в запрос для БД (через процедуру?)
4. Если БД может добавить пользователя:
   1. Сервер получает положительный ответ о добавлении нового пользователя и передает его сайту (+ кука?)
   2. Сайт авторизирует пользователя и переводит его на начальную страницу (личный кабинет/главная страница)
5. Если БД не может добавить пользователя:
   1. Передача «кода ошибки» серверу
   2. Сервер интерпретирует ошибку для сайта
   3. Сайт выдает ошибку пользователю в понятном для него виде

**Добавление интересов:**

1. Ввод интереса в поле
2. Передача интереса серверу
3. Сервер передает интерес БД (через процедуру?)
4. БД производит поиск данного интереса у пользователя
5. Если такой интерес есть:
   1. ???
6. Если такого интереса нет:
   1. Добавление интереса в список в БД
   2. Передача положительного ответа серверу
   3. Сервер передает положительный ответ сайту
   4. Отображение интереса на сайте

**Удаление интересов:**

1. Удаление интереса с помощью кнопки (?) на сайте
2. Передача удаляемого интереса на сервер
3. Сервер передает интерес БД для удаления (через процедуру?)
4. БД удаляет интерес из списка интересов студента и передает положительный ответ серверу
5. Сервер передает положительный результат сайту
6. Интерес, который было необходимо удалить, пропадает из списка интересов

**Добавление контактных данных:**

1. Ввод контактных данных в поля (далее КД)
2. Передача КД серверу
3. Сервер передает КД в БД (через процедуру?) ------ нужен json?
4. БД производит поиск данных КД у пользователя
5. Если КД были введены:
   1. Изменение КД
6. Если КД нет:
   1. Добавление КД в поля в БД
   2. Передача положительного ответа серверу
   3. Сервер передает положительный ответ сайту
   4. Отображение КД на сайте

**Удаление контактных данных:**

1. Удаление контактных данных с помощью кнопки (?) на сайте (далее КД)
2. Передача удаляемых КД на сервер
3. Сервер передает КД в БД для удаления (через процедуру?)
4. БД удаляет КД из информации о студенте и передает положительный ответ серверу
5. Сервер передает положительный результат сайту
6. КД, которые было необходимо удалить, пропадают на сайте

**Получение списка проектов:**

1. Сайт передает запрос серверу на получение списка проектов
2. Сервер передает запрос БД (через процедуру?)
3. Если проекты есть:
   1. БД передает серверу список проектов
   2. Сервер передает список проектов сайту
   3. Сайт выводит список
4. Если проектов нет:
   1. БД передает «код ошибки» серверу
   2. Сервер интерпретирует ошибку для сайта
   3. Сайт выдает ошибку пользователю в понятном для него виде

**Поиск среди проектов по термину:**

1. Термин вводится в поле поиска на сайте
2. Сайт передает термин серверу
3. Сервер отправляет запрос на поиск БД
4. БД ищет совпадения по списку проектов
5. Если совпадения есть:
   1. БД передает серверу список подходящих проектов
   2. Сервер передает список сайту
   3. Сайт выводит их на экран (как кликабельные ссылки? Если да, то в зависимости от количества будет передавать данные по каждому проекту сразу?) ---------------------????
6. Если совпадений нет:
   1. БД передает «код ошибки» серверу
   2. Сервер интерпретирует ошибку для сайта
   3. Сайт выдает ошибку пользователю в понятном для него виде (Таких проектов нет)

**Создание проекта:**

1. Ввод данных о проекте на сайте
2. Преобразование данных в запрос на сервер через JSON/XML
3. Сервер парсит данные и формирует их в запрос для БД (через процедуру?)
4. Если БД может добавить проект:
   1. Сервер получает положительный ответ о добавлении нового проекта и передает его сайту
   2. Сайт показывает уведомление о том, что проект добавлен
5. Если БД не может добавить проект:
   1. Передача «кода ошибки» серверу
   2. Сервер интерпретирует ошибку для сайта
   3. Сайт выдает ошибку пользователю в понятном для него виде

**Получение списка рекомендованных проектов:**

?????

**Получение списка участников проекта:**

1. Сайт передает запрос серверу на получение списка участников проекта
2. Сервер передает запрос БД (через процедуру?)
3. Если участники есть:
   1. БД передает серверу список участников
   2. Сервер передает список участников сайту
   3. Сайт выводит список (как кликабельные ссылки? Если да, то в зависимости от количества будет передавать данные по каждому участнику сразу?)
4. Если участников нет:
   1. БД передает «код ошибки» серверу
   2. Сервер интерпретирует ошибку для сайта
   3. Сайт выдает ошибку пользователю в понятном для него виде

**Получение дополнительной информации по проекту:**

1. Сайт передает запрос серверу на получение информации по проекту
2. Сервер отправляет запрос БД на получение информации
3. БД передает серверу информацию по проекту
4. Сервер передает информацию сайту
5. Сайт выводит ее на экран

**Получение списка студентов:**

1. Сайт передает запрос серверу на получение списка студентов
2. Сервер передает запрос БД (через процедуру?)
3. Если студенты есть:
   1. БД передает серверу список студентов
   2. Сервер передает список студентов сайту
   3. Сайт выводит список
4. Если студентов нет:
   1. БД передает «код ошибки» серверу
   2. Сервер интерпретирует ошибку для сайта
   3. Сайт выдает ошибку пользователю в понятном для него виде

**Поиск среди студентов:**

1. Данные вводятся в поле поиска на сайте
2. Сайт передает данные серверу
3. Сервер отправляет запрос на поиск БД
4. БД ищет совпадения по списку студентов
5. Если совпадения есть:
   1. БД передает серверу список подходящих студентов
   2. Сервер передает список сайту
   3. Сайт выводит их на экран (как кликабельные ссылки? Если да, то в зависимости от количества будет передавать данные по каждому проекту сразу?)
6. Если совпадений нет:
   1. БД передает «код ошибки» серверу
   2. Сервер интерпретирует ошибку для сайта
   3. Сайт выдает ошибку пользователю в понятном для него виде (Таких студентов нет)

**Получение информации по конкретному студенту по его** **идентификатору в домене @edu.hse.ru:**

1. Идентификатор вводится в поле поиска на сайте
2. Сайт передает идентификатор серверу
3. Сервер отправляет запрос на поиск БД и получение информации
4. БД ищет совпадения по списку студентов
5. Если совпадения есть:
   1. БД передает серверу информацию по студенту
   2. Сервер передает информацию сайту
   3. Сайт выводит ее на экран
6. Если совпадений нет:
   1. БД передает «код ошибки» серверу
   2. Сервер интерпретирует ошибку для сайта
   3. Сайт выдает ошибку пользователю в понятном для него виде (Такого студента нет)