**1 слайд (титульный)**

Здравствуйте, уважаемые члены ГЭК, целью данной работы является проектирование и реализация веб-приложения сервиса для организации игр в лазертаг. Полученная система упростит и улучшит процесс организации игр для лазертаг-клубов, а также упростит игрокам поиски игр.

**2 слайд (Постановка задачи)**

Система должна содержать данные о местах проведения игр, об игроках, командах, клубах, турнирах, самих играх и об игровой экипировке и ее аренде.

Работа с системой должна происходить через браузер и иметь интуитивно понятный интерфейс для пользователя.

**3 слайд (Системы-аналоги)**

На данном слайде представлены 2 системы-аналога:

- Makeready.ru – Это платформа для организации и проведения турниров по стрельбе. Из-за специфики платформы она ориентирована на регистрацию участников, составление расписаний и обработку результатов соревнований. Однако данная система не предназначена для сопровождения командных игр типа лазертага, не поддерживает управление снаряжением, распределение игроков по командам.

- Gamra.ru – Сервис, специализирующийся на анонсах страйкбольных игр и регистрации участников. Платформа предоставляет базовый функционал для отображения информации об играх и списков игроков, но не предусматривает автоматизацию процессов аренды снаряжения, отслеживания результатов игр или формирования команд, что критически важно для полноценной организации лазертаг-мероприятий.

Таким образом, задача разработки собственной системы организации игр в лазертаг является актуальной.

**4 слайд (Логическая модель БД)**

На данном слайде представлена логическая модель базы данных автоматизированной информационной системы. Она включает в себя 10 сущностей, у каждой из которых присутствует первичный ключ и атрибуты, наиболее полно отображающие характеристики каждой сущности.

**5 слайд (Диаграмма ВИ)**

Был разработан логический проект системы по методологии UML.

На диаграмме вариантов использования можно увидеть, что в системе присутствует несколько ролей.

Вход в систему определяет какая роль принадлежит пользователю.

Администратор создает и редактирует игры, турниры, игровые предметы и локации в рамках клуба.

Игрок создает и редактирует команды, участвует в играх или турнирах, и арендует игровое оборудование.

**6 слайд (Диаграмма деятельности)**

На данном слайде представлена диаграмма деятельности для варианта использования «Создать игру». На ней показаны действия каждого актора: Админстратора, клиентской части приложения и серверной части.

**7 слайд (Диаграмма последовательности)**

На данном слайде представлена диаграмма последовательности для варианта использования «Пригласить в команду». Тут акторами является: игрок, клиентская часть и серверные части приложеня.

**8 слайд (Выбор средств проектирования и реализации)**

В качестве средства проектирования был выбран язык UML,

В качестве средства для построения UML-диаграмм был выбран онлайн-сервис «Draw.io»

Проект является одностраничным приложеним (SPA), написанном на языке программирования TypeScript. Особанность проекта в том, что Typescript используется и на клиенте, и на сервере.

В качестве основной архитектуры используется монолитная архитектура. На бекенде используется паттерн MVC.

В качестве СУБД выбрана PostgreSQL, в качестве ORM была выбрана ORM Prisma.

**9 слайд (Схема алгоритма создания локации)**

На данном слайде представлена Схема алгоритма создания локации. Показана схема работы сервера и схема работы клиента. Остальные сущности создаются по аналогичным принципам.

**10 слайд (Схема алгоритма просмотра локации)**

На данном слайде представлена Схема алгоритма просмотра локации, как на клиенте, так и на сервере.

**11 слайд (Схема алгоритма просмотра статистики игрока)**

На данном слайде представлена Схема алгоритма просмотра статистики игрока по датам на сервере. Схема алгоритма просмотра на клиенте аналогична схеме просмотра любой сущности.

**12 слайд (Пример работы: Вход и регистрация администратора)**

Рисунок 1 – страница входа в приложение.

Для создания аккаунта администратора нужно перейти на страницу создания аккаунта и зарегистрироваться с ролью «администратор».

**13 слайд (Пример работы: Создание клуба администратором)**

Одна из основных сущностей системы – клуб. Для того, чтобы создать клуб, администратор переходит с главной страницы, на страницу с клубами и нажимает кнопку «создать», после чего в диалоговом окне вбивает данные о клубе и нажимает кнопку «создать клуб», затем клуб появляется в списке клубов.

**14 слайд (Пример работы: Создание локации администратором)**

Обязательной сущностью для функционирования клуба является сущность – локация. Поэтому для создания локации для клуба администратор переходит на страницу «места» и создает локацию, выбрав клуб, которому принадлежит локация.

**15 слайд (Пример работы: Создание игры администратором)**

Ну и главная сущность всей системы – Игра. Для этого администратор проводит такие же действия, как и для других сущностей, выбрав нужную локацию, указав время проведения.

Игра появляется на главной странице администратора и показывает статус «Создана» и находится в ожидании регистрации игроков.

**16 слайд (Пример работы: Регистрация игрока в системе)**

Чтобы зарегистрироваться в системе игроком, достаточно при регистрации указать роль «игрок». Заметим, что интерфейс приложения поменялся – появились другие пункты меню, и общая цветовая гамма поменялась на более светлую.

**17 слайд (Пример работы: Регистрация игрока на игру)**

Игрок заходит на страницу «Игры» и видит ранее созданную игру. Щелкает на нее и появляется диалоговое окно в котором игрок выбирает оборудование, которое он хотел бы арендовать на эту игру. После выбора кликает на кнопку «Зарегистрироваться», и игра появляется на главной странице игрока.

**18 слайд (Пример работы: Процесс проведения игры администратором)**

Администратор заходит с главной страницы на игру, которую создал.

Видит зарегистрированных игроков. После клика на кнопку «Завершить регистрацию» игра переходит в статус «Активная», зарегистрироваться на игру больше нельзя, и администратору показывается оборудование, которое следует подготовить к аренде на игру.

Когда все игроки собрались – администратор нажимает кнопку «начать игру». Игра переходит в статус «в процессе».

После окончания игры администратор нажимает кнопку «завершить игру», после чего выбирает выигравшую команду, игрокам которой начисляются очки.

**19 слайд (Заключение)**

Система реализована на языке TypeScript c использванием интерпретатора nodejs, и preact. Использовалась СУБД PostgreSQL. В качестве ORM была выбрана Prisma. Разработка велась в среде VSCode.

По методологии UML разработан логический проект системы с помощью средства Draw.io. Вся требуемая функциональность была разработана.

**20 слайд (Спасибо за внимание)**

Доклад окончен. Спасибо за внимание.