



**МЕЖДУНАРОДНАЯ ОЛИМПИАДА
ПО ПРОМЫШЛЕННОЙ РАЗРАБОТКЕ**

Т БАНК

**ЦЕНТРАЛЬНЫЙ
УНИВЕРСИТЕТ**

**ВЫСШАЯ ШКОЛА
ЭКОНОМИКИ**

Платформа привилегий клиентов Т-Банка

Мы создаем новую точку входа наших клиентов в возможности привилегий подписок Pro и Premium. Сейчас некоторые привилегии не видны клиентам, а многие вообще не знают об их существовании. Наша задача — сделать точку входа, в которой будут собраны доступные привилегии для клиента.

Цели проекта:

1. Все привилегии на одном экране.
2. Минимизация шагов для того, чтобы понять, какие привилегии доступны и как ими пользоваться.
3. Лаконичный интерфейс в стиле Т-Банка.

Требования к реализации:

- Авторизация
- При регистрации клиент должен указывать свои ФИО, дату рождения, телефон, почту и другие данные, если требуется для вашей реализации..
- Данные по привилегиям и клиентам хранятся в базе данных. Часть данных можно хранить в cookies
- Учет подписки и грейда клиента. Они напрямую влияют на наполнение привилегий. Есть три варианта грейда подписки:
 - без подписки;
 - подписка T-Pro;
 - подписка T-Premium.
- Bronze (платный, с ежемесячной оплатой 2990p)

- Silver (баланс 3млн+)
- Gold (баланс 5млн+)
- Diamond (баланс 10млн+)
- Грейды должны предоставляться за выполнение условий, указанных выше. При этом, вы можете изменить условия на свое усмотрение, но сохранением общей логики распределениях привилегий по грейдам (например — срок жизни клиента может менять базовое наполнение). Грейд должен проставляться автоматически при регистрации на основании введенных данных/данных из базы..

Все опции на одном экране:

- Должно быть понятно, что доступно, а что нет.
- Есть возможность “провалиться” в опцию и почитать про нее подробнее.

Учет использования привилегий:

- Клиент должен видеть, какие привилегии он использовал и что нужно, чтобы использовать их вновь (например, счетчик обновляется раз в 30 дней / каждый день).
- Учет лимитов сумм привилегий (например, компенсация ресторана не более 500 рублей в неделю и за такую опцию списывается проход в бизнес-зал).
- После использования привилегии клиент должен понимать, что она ему недоступна или оставшееся количество применений уменьшилось (например, даем 2 бизнес-зала, клиент использовал 1 — значит, 1 остался).
- Реализовать простую админку, в которой можно апдейтить данные клиентов, например — изменять остаток денег. После таких изменений в админке — клиент должен видеть обновленные привилегии.
- Реализовать возможность ручного изменения количества доступных привилегий для клиента. Их можно увеличить, уменьшить или вообще закрыть доступ.

Привилегии, которые клиент может получить (можно посмотреть в приложении или [на сайте](#)):

- Подписки на сервисы просмотра кино.
- Подписки на антивирус.
- Бизнес-залы.
- Скидки в ресторанах и кафе.
- Бесплатные мероприятия
- И т. д.



МЕЖДУНАРОДНАЯ ОЛИМПИАДА
ПО ПРОМЫШЛЕННОЙ РАЗРАБОТКЕ



Платформа для управления ресурсами GPU в Университете

Где используется GPU*:

- Игры и мультимедиа — для отрисовки реалистичной графики.
- Машинное обучение и ИИ — GPU ускоряет обучение нейросетей.
- Научные исследования — для моделирования сложных систем.
- Криптовалюты — добыча (майнинг) криптовалют.
- Графические приложения — видеомонтаж, 3D-анимация, дизайн.

Проблема:

- Ресурс GPU дорогой и покупка даже 10—20 GPU может обойтись в миллионы рублей.
- Если распределять ресурсы неоптимально, счета по GPU будут расти и со временем могут обанкротить Университет.
- Неоптимальное распределение ресурсов может привести к тому, что из 100 студентов воспользоваться GPU для работ смогут 10 студентов, а остальные будут сидеть ждать, когда кто-то освободит.

Поэтому важно иметь способ управления, чтобы ресурсы не простаивали и никто не занимал сразу все GPU, блокируя работу других студентов.

Цель: создать платформу, которая позволит гибко управлять выделением ресурсов GPU.

* GPU — графический процессор

Такая платформа:

- Поможет Университету понять, сколько GPU закуплено и сколько на деле используется. Исходя из данных вуз сможет оценить, сколько нужно оплачивать GPU и не переплачивать за неиспользованные ресурсы.
- Оптимально распределять ресурсы между студентами.
- Позволит отслеживать, как используются ресурсы, какая нагрузка.
- Позволит вводить ограничения на время работы.

Основные роли:

- Администратор, который управляет ресурсами. Указывает, какое количество ресурсов есть, задает ограничения, видит, как используются ресурсы.
- Студенты и преподаватели, которые могут получить доступ к ресурсам и понимать, какие ограничения по использованию у них есть.

Функциональные требования

Администратор:

1. Управление доступными ресурсами. Администратор создает группу и указывает, сколько GPU доступно, какое распределение. Пример: группа «Машинное обучение», 1 курс. Доступно: 4 GPU. Распределение ресурсов: 25%.
2. Настройка ограничений на использование: максимальное время выполнения задачи (сколько часов), ограничение по времени суток (какие дни недели/время).
3. Добавление пользователей, проставление ролей (студент/преподаватель) и группы.
4. История запусков. Каждый запуск фиксируется в истории использования GPU с данными. Пример: Запуск_1, дата и время запуска, сколько времени было запущено, пользователь.

Студент/Преподаватель

1. Использование ресурсов GPU: Студент/Преподаватель видит, какое количество ресурсов в его группе доступно и с какими ограничениями.
2. Есть возможность запустить свое решение с использованием GPU — упростим и оставим кнопку «Использовать» / «Завершить использование». При этом важно учитывать очередь.
3. Если свободного ресурса нет, то Студент/Преподаватель должен иметь возможность встать в очередь. После нажатия кнопки «Встать в очередь» необходимо отображать кто сейчас в очереди и на какой позиции находится пользователь.

4. Платформа должна показывать пользователю, сколько времени на использование осталось. Например: максимальное время выполнения задач — 1 час, после нажатия кнопки «Использовать» осталось 59 минут.
5. Отображение информации о том, сколько времени на выполнение задачи осталось.

Технические рекомендации:

1. Нужно эмулировать процесс работы с GPU
2. Должна быть возможность воспользоваться «машиной времени» — специальный функционал в интерфейсе, который позволит перемещаться вперёд по таймлайну



**МЕЖДУНАРОДНАЯ ОЛИМПИАДА
ПО ПРОМЫШЛЕННОЙ РАЗРАБОТКЕ**

Т БАНК

**ЦЕНТРАЛЬНЫЙ
УНИВЕРСИТЕТ**

**ВЫСШАЯ ШКОЛА
ЭКОНОМИКИ**

Единая корзина для заказов из разных супермаркетов в Т-Банке

Сервис «Супермаркеты» в приложении Т-Банка позволяет пользователям заказывать продукты онлайн из популярных магазинов, таких как ВкусВилл, Перекресток, Магнит и других. Однако существует ограничение: в одном заказе можно выбрать товары только одного партнера. Это создает несколько проблем:

1. Неудобства для пользователей:

- Невозможность собрать один список покупок с товарами из разных магазинов.
- Необходимость делать несколько заказов, что занимает больше времени.
- Невозможность выбирать самые выгодные предложения из разных магазинов.
- Отсутствие единого окна отслеживания статусов заказов.

2. Ограничения для партнеров:

Пользователи могут отказаться от заказа у менее удобного магазина, если он не предлагает нужный ассортимент.

Цель

Задача — создать единый интерфейс заказа продуктов, который решает указанные проблемы и позволяет пользователю:

1. Добавлять товары из разных магазинов в одну корзину.
2. Оформлять один заказ, который будет разделен на подзаказы по партнерам.

Обязательные функции

1. Отображение ленты товаров от различных партнеров:
 - Товары могут быть одинаковыми у разных партнеров, важно избежать дублей одного товара от разных партнеров.
 - Товары должны быть сгруппированы, чтобы пользователь видел разные варианты одного товара от разных партнеров.
2. Фильтры товаров:
 - Фильтрация товаров по конкретному магазину.
3. Поиск товаров:
 - Возможность искать товары по названию или категории.
4. Сортировка по цене:
 - Сортировка товаров по цене среди всех доступных партнеров, с отображением самого выгодного варианта.
5. Добавление товаров в корзину:
 - Возможность добавлять товары из разных магазинов в одну корзину.
6. Автоматическое разделение заказа:
 - После оформления заказа система должна автоматически разделить его на подзаказы по партнерам при отображении.
7. История заказов:
 - Созданный заказ должен отображаться в истории заказов пользователя.
8. Планирование логистики:
 - Выбор способа получения заказа