

Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплины:

Технологии разработки программного обеспечения
и Технологии обработки финансовой информации.

Техническое задание

“Веб приложение для онлайн покупки билетов на концерты”

Выполнил:
студент гр. 953502
Михнюк Я. Ю.
Севко П.Р

Минск 2022

Содержание

1. Введение	3
1.1. Назначение проекта	3
1.2. Предполагаемая аудитория	3
1.3. Последовательность восприятия	4
2. Общее описание	5
2.1. Функциональность продукта	5
2.2. Классы и характеристики пользователей	5
3. Функциональность системы	6
3.1. Функциональный блок Back-End	6
3.1.1. Описание и приоритет	6
3.1.2. Функциональная карта	6
3.1.3. Функциональные требования	7
3.1.4. Используемые сущности	7
4. Требования к внешним интерфейсам	9
4.1. Программные интерфейсы	9
5. Нефункциональные требования	10
5.1. Требования к сохранности данных	10
5.2. Используемые технологии	10

1. Введение

1.1. Назначение проекта

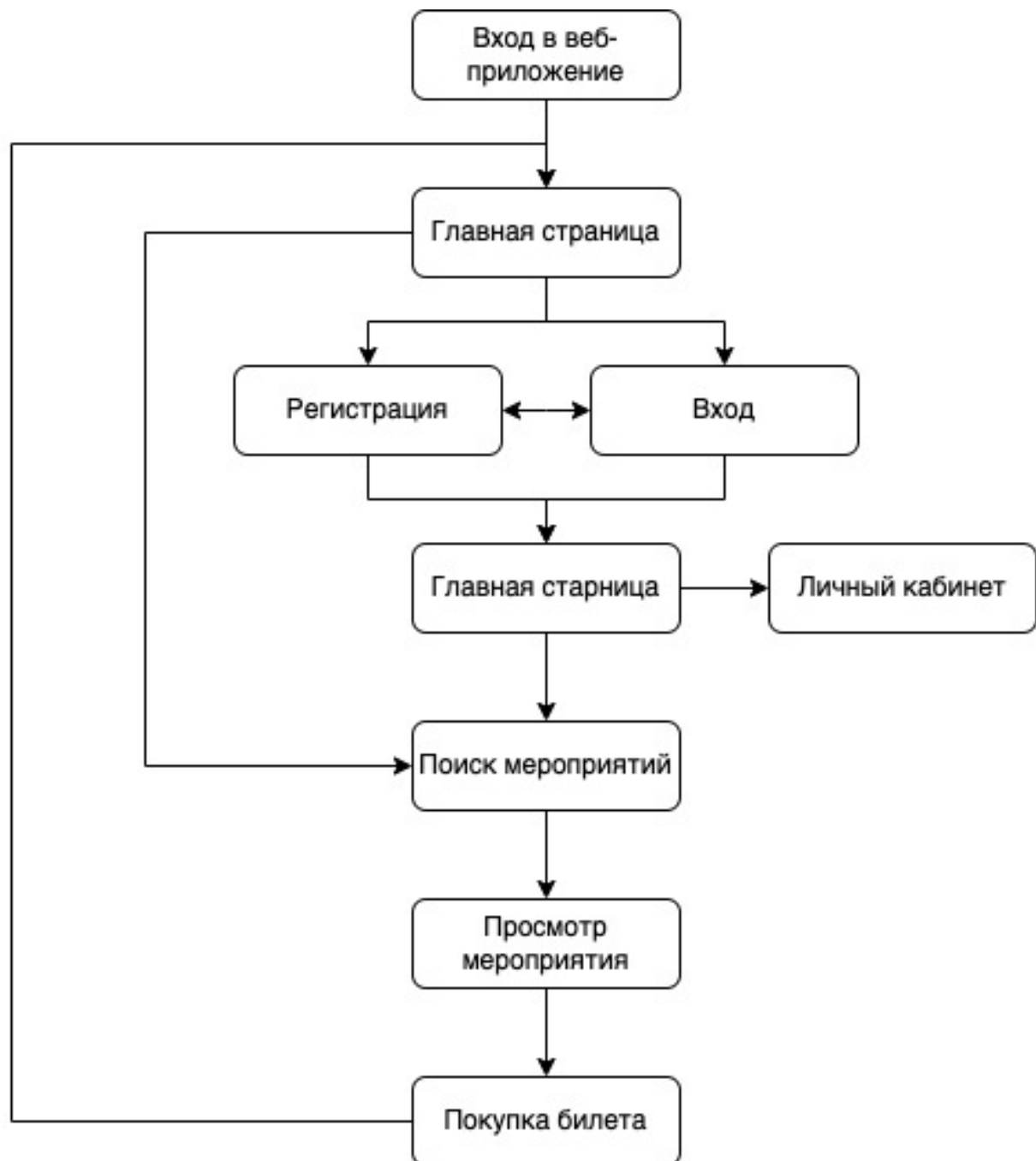
Наша миссия состоит в том, чтобы предоставлять людям невероятные впечатления, начиная с билетов на живые мероприятия.

Мы считаем, что живые мероприятия — одни из самых важных и искренних моментов в нашей жизни, и что, давая людям возможность посещать и проводить время на таких мероприятиях, мы поможем создать более искренний и сплоченный мир. Таким образом, целью проекта является разработка прогрессивного веб-приложения, которое сочетало бы в себе все функции и процессы, необходимые для организации и продажи билетов на концерты. В процессе разработки будут задействованы современные инструменты для создания веб-приложения.

1.2. Предполагаемая аудитория

Целевой аудиторией будут являться люди, предпочитающие посещать мероприятия, концерты и слушать любимых исполнителей вживую.

1.3. Последовательность восприятия



2. Общее описание

2.1. Функциональность продукта

Функции:

- Регистрация пользователя
 - Авторизация пользователя
 - Просмотр списка мероприятий
 - Покупка билетов
 - Система промокодов
 - Система наград и «ачивок»
-

2.2. Классы и характеристики пользователей

Веб-приложение будет иметь единственный вид пользователей - клиент, который будет иметь возможность регистрироваться, входить в систему, просматривать доступные мероприятия, приобретать билеты, применять промокоды, а также просматривать полученные награды, например за посещение определенного количества концертов некоторого исполнителя или за посещение мероприятий в определенных местах.

3. Функциональность системы

3.1. Функциональный блок Back-End

3.1.1. Описание и приоритет

Серверная часть приложения представляет собой выделенный сервер, который будет собирать данные о мероприятиях при помощи API, собирать данные о покупке билетов и получении наград. После сбора данных они будут сохраняться в базу данных и предоставляться по запросу клиентской части.

Приоритет будет ставиться на удобство запроса данных, простоту и понятность использования клиентской частью.

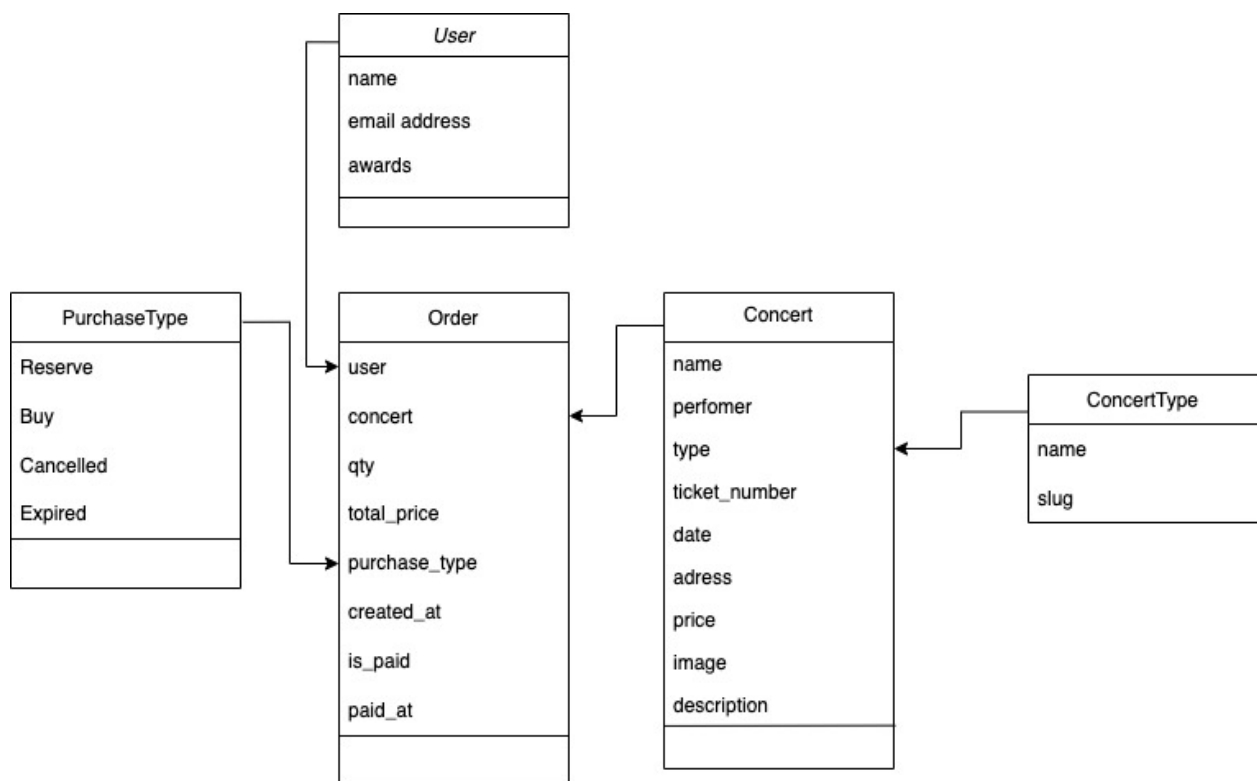
3.1.2. Функциональная карта



3.1.3. Функциональные требования

1. Запрос данных о доступных мероприятиях у API и сохранение их в базу данных.
2. Покупка билетов
3. Предоставление данных о купленных билетах и полученных наградах по запросу клиентской части.

3.1.4. Используемые сущности



Описание сущностей:

User – пользователь приложения.

Consert – описание мероприятия.

Order – заказ, создается при покупке билета/билетов.

ConcertType – описание типа мероприятия.

PurchaseType – описание состояния оплаты корзины .

Допускается изменение/дополнение сущностей в процессе разработки.

4. Требования к внешним интерфейсам

4.1. Программные интерфейсы

При обращении к конечным точкам API будет собираться информация с этих точек и вся информация из вложенных конечных точек.

Примеры конечных точек:

- *Serve/concert*
- *Server/concert/:id*
- *Server/purchase*

Допускается изменение/дополнение конечных точек API во время разработки.

4.2. Интерфейсы пользователя (UX)

Figma: [раскладка фигмы](#)

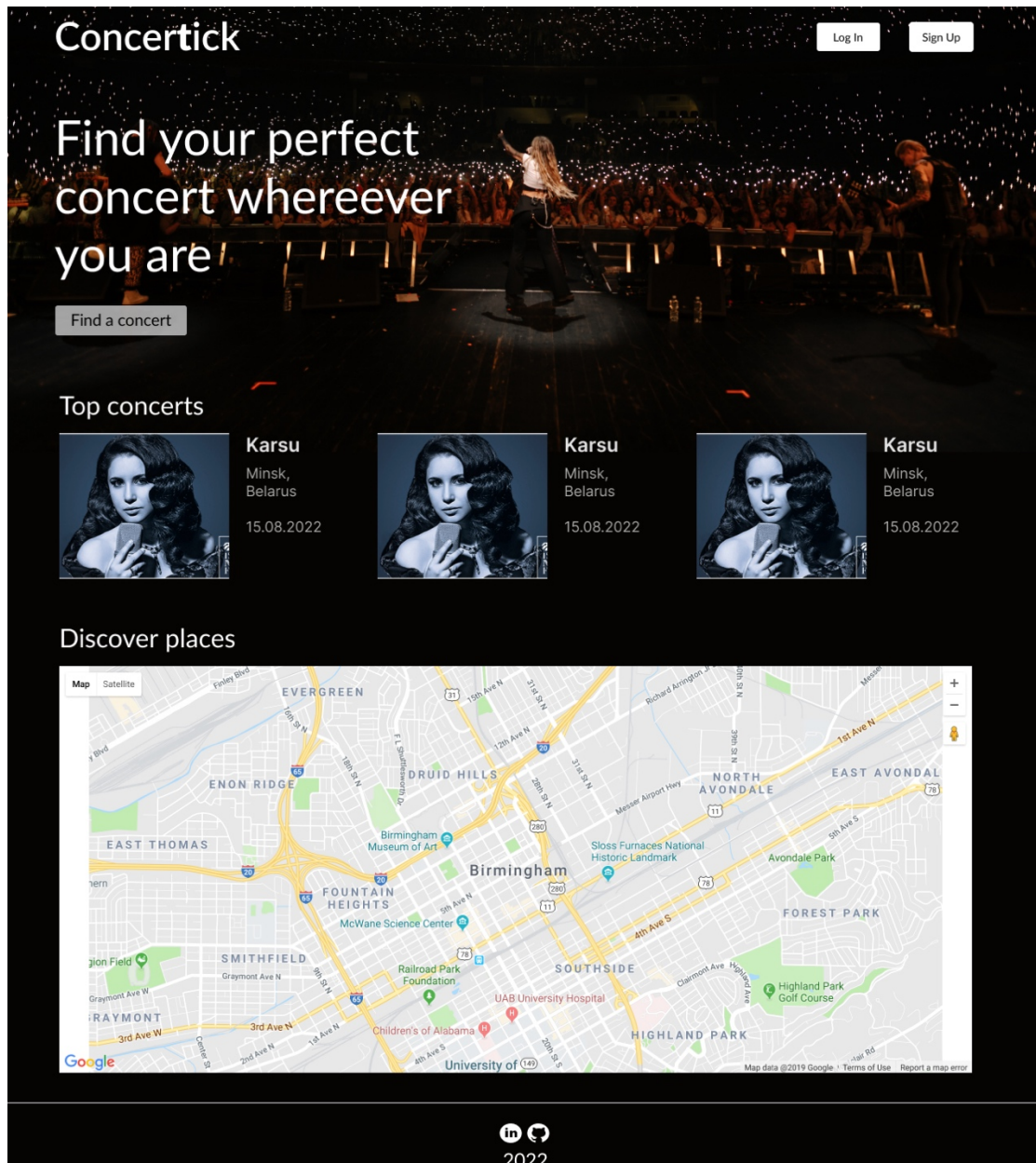


Рис. 1. Целевая страница

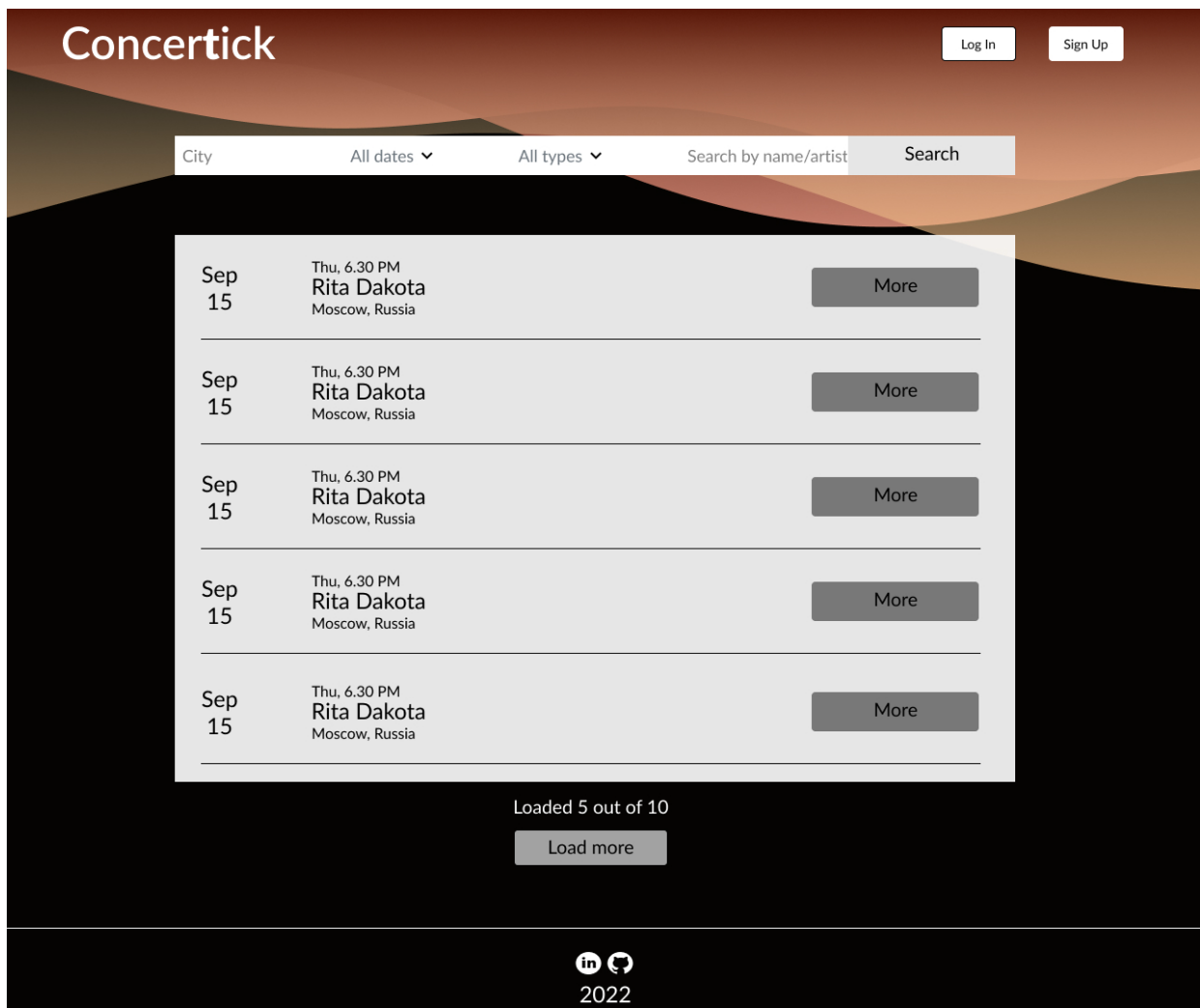


Рис. 2. Список мероприятий


Concertick

Log InSign Up



Rita Dakota

Thursday, 18.05.2022, 8.00 PM
Moscow, Russia, Barclays Arena
100 \$

Buy



Type	Classic
Voice	Tenor
Antract	Yes



2022


Рис. 3. Страница мероприятия

Concerttick

[Log In](#)[Sign Up](#)

Rita Dakota

Thursday, 18.05.2022, 8.00 PM
Moscow, Russia, Barclays Arena
100 \$



Apply

Tickets available: 100

Number of tickets:

-



 1

+

Type of purchase: ☒ Reserve ☐ Buy

Continue

100 \$



2022


Рис. 4. Страница заказа билетов

Concertick

Log InSign Up

Rita Dakota

Thursday, 18.05.2022, 8.00 PM, 6.30 PM
Moscow, Russia, Barclays Arena
100 \$





Number of tickets:1

Total price, \$:100

Payment method:
☒ Paypal

Buy/Reserve



2022

Рис. 5. Страница оплаты

5. Нефункциональные требования

5.1. Требования к сохранности данных

Действия, применяемые для сохранности данных:

- Экранирование данных, приходящий с клиентской части, для избежания инъекций.
 - Определенное максимальное количество запросов за определенный промежуток времени.
 - Запрос данных с сервера по токену.
 - Приложение не должно хранить данные оплаты в чистом виде.
-

5.2. Используемые технологии

- Python
- Django / Django REST framework
- PostgreSQL
- Songkick API
- PayPal API