

Решение кейса от компании «Ахепіх»

Команда: «Пандури, чонгури и дудук»

Наша команда

Студенты 3-го курса СПбГЭУ

Направление: «Экономика»

Профиль: «Математическое моделирование и анализ данных в экономике»

1

Анализ существующих решений, выявление основных аспектов пользовательского опыта

2

Написание логики кода, продумывание алгоритмов и оптимизация кода

3

UX/UI дизайн – исследование пользовательского опыта, создание пользовательского сценария, создание прототипа сайта

Катя

Егор

Аким



Анализ существующих решений

Критерий	Tutu.ru	Яндекс Путешествия	Trip and Fly
Сильные стороны			
Широкий выбор билетов	Да (авиа, ЖД, автобусы, электрички)	Да (авиа, ЖД)	Да (все виды транспорта)
Гибкие фильтры	Да (цена, время, пересадки)	Да (расширенные фильтры для авиа)	Да (широкий функционал)
Рейтинг перевозчиков	Да	Нет	Нет
Возможность бронирования отеля при пересадке	Да (ночные пересадки)	Да (отель можно забронировать через сервис)	Нет
Опция заказа такси или трансфера	Нет	Да	Нет
Удобство поиска	Да (интуитивно понятный интерфейс)	Да	Да
Сортировка билетов	Да (по разным критериям)	Да (более расширенная)	Да
Мультимодальные маршруты	Нет	Нет	Да
Слабые стороны			
Отсутствие предложений по снижению стоимости	Да	Нет	Да (ограниченные рекомендации)
Отсутствие мультимодальных маршрутов	Да	Да	Нет
Ограниченные настройки пересадок	Да (нет информации о смене вокзалов, трансферах)	Да (нет фильтров для сложных пересадок на ЖД)	Да (нет информации о трансферах)
Недостаточная детализация пересадок в аэропортах	Да	Да	Да
Нет интеграции с пробками и задержками рейсов	Да	Да	Да

На основе анализа конкурентов мы пришли к выводу, что наш продукт должен обладать следующими ключевыми свойствами:

- Возможность комбинирования различных видов транспорта: авиа, ж/д, автобусы, такси, трансферы.
- Расширенные настройки пересадок
 - Фильтр «без смены вокзала/терминала».
 - Возможность выбора длительности пересадки более нескольких суток.
 - Отображение расстояний между терминалами и расчет времени на перемещение.
 - Информация о доступных вариантах трансфера (такси, общественный транспорт, пешком).
 - Возможность бронирования отеля при длительной или ночной пересадке.
- Гибкая система фильтров и сортировки
 - Фильтры по цене, количеству пересадок, времени в пути, уровню комфорта.
 - Возможность исключать неудобные варианты (ночные рейсы, короткие пересадки).
- Интеграция с сервисами реального времени
 - Учет пробок, загруженности аэропортов и вокзалов.
 - Отображение задержек рейсов и поездов.
 - Обновление времени в пути с учетом текущих дорожных условий.
- Умные подсказки и рекомендации
 - Предложения более дешевых вариантов при изменении дат или маршрута.
 - Советы по наиболее удобным маршрутам на основе пользовательских предпочтений.
- Четкий интерфейс с интуитивным поиском и быстрым доступом к фильтрам.

Часть кода

Библиотеки/модули

```
import os
import json
import logging
import requests
import pandas as pd
import networkx as nx
from networkx.algorithms.simple_paths import shortest_simple_paths
from datetime import datetime, timedelta
from typing import Any, Dict, List, Tuple, Union
from time import sleep
from IPython.display import display
```

Код реализует систему поиска оптимальных маршрутов между городами, используя данные из Яндекс.Расписаний и Aviasales.

- Он загружает список станций, кэшируя их в JSON и CSV, чтобы избежать повторных запросов.
- Станции фильтруются, выделяются аэропорты и вокзалы, а затем выбираются подходящие транспортные узлы.
- Автоматически определяются коды станций (Яндекс, IATA).

После ввода пользователем города отправления, прибытия и даты код отправляет запросы к API Яндекс.Расписаний для поиска доступных маршрутов различных видов транспорта, а также к Aviasales для получения авиабилетов и их стоимости. Найденные сегменты маршрутов объединяются в единый JSON и CSV.



[Ссылка на GitHub с кодом и решением](#)

Часть кода

Создаётся граф маршрутов, где:

- Узлы – транспортные узлы
- Рёбра – маршруты с весами, зависящими от времени или стоимости.

Алгоритм поиска K кратчайших путей в NetworkX находит маршруты с учётом пересадок. Формируется таблица с деталями поездки, результаты сохраняются в CSV и выводятся на экран.

Маршруты ранжируются по времени или стоимости: алгоритм выбирает пути с минимальным значением веса, а при сортировке по цене добавляется колонка с общей стоимостью.



[Ссылка на GitHub с кодом и решением](#)

Часть кода

Код будет полезен:

- для автоматизированного поиска оптимальных маршрутов между городами, учитывая разные виды транспорта, время в пути и стоимость
- может применяться в логистике, планировании поездок и анализе транспортных потоков, обеспечивая удобный доступ к актуальной информации.

Его можно развить, добавив источники данных, требующие партнёрства или платных API, что позволит получать более точные и полные сведения о ценах и наличии билетов. Интеграция дополнительных сервисов поможет детализировать маршруты, разбивая их на районы или даже более мелкие зоны.

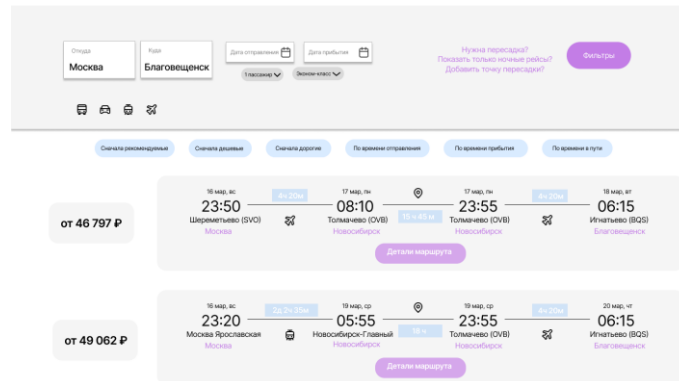
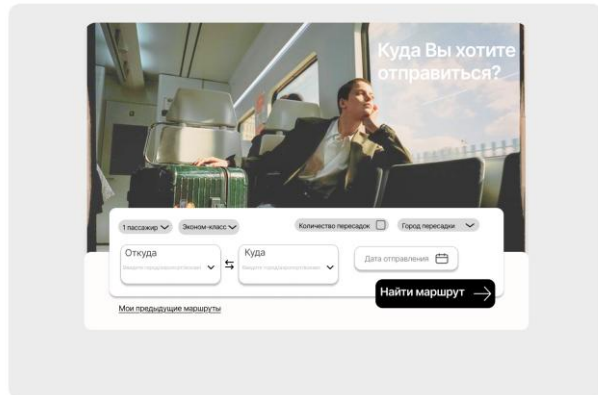
Для этого можно разработать собственный веб-сервис, сайт или приложение, которое объединит несколько API, упростит работу с маршрутами и обеспечит расширенные функции, такие как динамическое ценообразование, персонализированные рекомендации и более точный учёт местных особенностей.



[Ссылка на GitHub с кодом и решением](#)

User Journey Map - Карта пользовательского пути

	Первое взаимодействие с сервисом	Поиск маршрута	Выбор маршрута	Путешествие (использование маршрута)
Действие)	Заходит на сайт, оценивает интерфейс	Вводит точки отправления и назначения, ищет маршрут	Сравнивает варианты, вносит изменения, уточняет детали	Следует маршруту, получает обновления и уведомления.
Цели и ожидания	Понять, как работает сервис. Проверить удобство	Получить понятный, удобный и точный маршрут	Сделать маршрут более подходящим и гибким	Без проблем следовать маршруту, получать актуальные данные
Мысли	"Не слишком ли сложно? Понятен ли интерфейс?"	«Как найти оптимальный маршрут?», «Как не потеряться в сложных пересадках?» «Как доехать быстрее и дешевле?»	«Есть ли варианты с удобными пересадками?»	«Успею ли я на пересадку?»
Точки контакта	Главная страница сайта, мобильное приложение	Фильтры, сортировка, информация о пересадках	Фильтры, сравнение вариантов	Нотификации, поддержка
Эмоции	Интерес, лёгкое беспокойство	Интерес, волнение	Сомнение, расчет	Волнение, контроль
Гипотеза	Чем проще и понятнее интерфейс, тем выше вероятность, что пользователь останется в сервисе	Чем гибче фильтры, тем выше удовлетворенность пользователя	Пользователи ценят гибкость в настройке маршрута	Интеграция с сервисами реального времени снижает тревожность
Проблема (точки боли)	Сложный интерфейс, не понимает, что делать Сразу просят регистрацию – теряет интерес Не уверен, что сервис решит его задачу	Не знает, какие параметры важны (например, время, бюджет, остановки) Беспокоится, что сервис не учтет все нюансы	Сложно сравнить маршруты. Много полей – запутался. Не знает, какие места выбрать. Беспокоится о совместимости разных видов транспорта	Потерял доступ к интернету Непредвиденные изменения (задержка рейса, пробки) Нужно быстро менять маршрут
Возможности	Понятный интерфейс, использование без регистрации	Упрощенный режим с рекомендациями "быстрее", "дешевле"	Сортировка по комфорту и цене, всплывающие подсказки	Интеграция с сервисами реального времени, уведомления, офлайн-доступ, возможность сохранить маршрут для будущих поездок



UX | UI

По ссылке можно
ознакомиться с прототипом
пользовательского сценария
(Figma):



Ссылка на GitHub с UI-дизайном
(тык на ссылку)

