Прогноз пассажиропотока и рекомендации по выпуску автобусов

Технологическая задача ТОО «SMSBUS»

Техническое задание

1. Границы пилота

Пилот проводится в одном городе: Костанай.

В пилот входят все маршруты города, обслуживаемые системой SMSBUS.

В пилот не входят: автоматическое изменение расписания, подсчёт пассажиров по видео, доработка мобильных приложений.

Ограничение данных: прогноз строится по оплаченным поездкам. Доля наличной оплаты в пилотном городе — 5%.

Существующая инфраструктура заказчика: Java (JSP, сервлеты), Apache Tomcat, СУБД Postgres. Сервис получает данные пакетной выгрузкой или через API загрузки и не требует изменений в системе приёма оплаты.

2. Роли и права доступа

Действие	Аналитик перевозчика	Специалист акимата	Администратор SMSBUS	Внешняя система (API)
Просмотр прогноза	Свои маршруты	Все маршруты города	Все города	По выданному ключу
Просмотр рекомендаций по выпуску	Свои маршруты	Все маршруты города	Все города	По выданному ключу
Сценарий «что если»	Да	Да	Да	Нет
Экспорт отчёта	Свои маршруты	Да	Да	Нет
Справочники и календарь событий	Нет	Календарь своего города	Да	Нет
Пользователи, ключи API, переобучение модели	Нет	Нет	Да	Нет

3. User Stories

ID	Роль	Я хочу	Чтобы
US-1	Аналитик перевозчика	видеть прогноз поездок по маршруту по часам на неделю вперёд	заранее планировать выпуск автобусов и смены водителей

ID	Роль	Я хочу	Чтобы
US-2	Аналитик перевозчика	получать рекомендацию по числу автобусов и интервалу на каждый час	не держать лишние машины на линии и не допускать переполнения
US-3	Специалист акимата	видеть маршруты с нехваткой провозной способности на ближайшую неделю	реагировать до появления жалоб
US-4	Специалист акимата	смоделировать изменение интервала или числа автобусов	оценить загрузку до принятия решения
US-5	Специалист акимата	добавлять в календарь городские события и перекрытия	прогноз учитывал их заранее
US-6	Администратор SMSBUS	видеть точность прогноза по городам и маршрутам	понимать, где модели можно доверять
US-7	Администратор SMSBUS	подключать новый город загрузкой справочников и истории	масштабировать сервис без доработки кода
US-8	Внешняя система	получать прогноз и рекомендации по API	показывать их в диспетчерской системе перевозчика

4. Критерии приёмки (Given-When-Then)

AC-1. Прогноз на неделю (US-1). Дано: по маршруту загружена история оплат не менее чем за 8 недель. Когда: аналитик открывает маршрут и выбирает ближайшие 7 дней. Тогда: сервис показывает прогноз поездок для каждого часа работы маршрута с нижней и верхней границей и датой расчёта.

AC-2. Мало истории (US-1). Дано: маршрут работает меньше 8 недель. Когда: аналитик запрашивает прогноз. Тогда: сервис показывает прогноз с пометкой «низкая достоверность» и не выдаёт рекомендаций по выпуску.

AC-3. Рекомендация по выпуску (US-2). Дано: для маршрута заданы вместимость автобуса, время оборота и допустимая загрузка. Когда: готов прогноз на следующий день. Тогда: сервис выдаёт на каждый час число автобусов и интервал, при которых прогнозная загрузка не превышает допустимую, и показывает разницу с текущим расписанием.

AC-4. Нехватка провозной способности (US-3). Дано: прогнозная загрузка маршрута в каком-либо часе выше допустимой. Когда: специалист акимата открывает сводку по городу. Тогда: маршрут стоит в списке проблемных с указанием дня, часа и недостающего числа автобусов.

AC-5. Сценарий «что если» (US-4). Дано: открыт маршрут с готовым прогнозом. Когда: пользователь меняет число автобусов в выбранном часе. Тогда: сервис не дольше чем за 3 секунды пересчитывает интервал и загрузку и не меняет сохранённое расписание.

AC-6. Календарь событий (US-5). Дано: специалист акимата добавил событие на дату в пределах 7 дней. Когда: выполнен очередной расчёт. Тогда: прогноз на эту дату пересчитан, а в списке факторов прогноза указано добавленное событие.

AC-7. Разграничение доступа (все роли). Дано: аналитик перевозчика А авторизован. Когда: он запрашивает маршрут перевозчика Б через кабинет или API. Тогда: сервис отвечает отказом (код 403) и записывает попытку в журнал.

AC-8. Нет свежих данных (US-6). Дано: за прошлые сутки данные об оплатах не поступили. Когда: запускается ночной расчёт. Тогда: сервис оставляет предыдущий прогноз, помечает его как устаревший и уведомляет администратора.

AC-9. Подключение города (US-7). Дано: администратор загрузил справочники и историю оплат нового города за 8 недель и более. Когда: завершён первый расчёт. Тогда: прогноз по маршрутам города доступен в кабинете и по API без изменения кода сервиса.

5. Структура API

Базовый путь `/api/v1`, формат JSON, авторизация по токену (OAuth 2.0, client credentials), описание в OpenAPI 3.0. Состав методов исполнитель может уточнить на этапе проектирования.

Метод	Путь	Назначение
POST	`/ingest/validations`	Приём пакета обезличенных оплат
POST	`/ingest/gps`	Приём пакета GPS-точек
GET	`/forecasts`	Прогноз поездок по маршруту и периоду
GET	`/recommendations`	Рекомендованный выпуск и интервалы
POST	`/scenarios`	Расчёт сценария «что если»
GET	`/deviations`	Часы и маршруты, где факт разошёлся с прогнозом
GET	`/quality`	Точность прогноза по городам и маршрутам
GET, PUT	`/calendar/events`	Календарь праздников, каникул и городских событий
GET	`/health`	Состояние сервиса и время последнего расчёта

Пример запроса и ответа:

```
GET /api/v1/forecasts?city_id=17&route_id=13&from=2027-02-01&to=2027-02-07&step=hour

200 OK
{
  "city_id": 17,
  "route_id": "13",
  "calculated_at": "2027-01-31T04:10:00+05:00",
  "confidence": "normal",
  "items": [
    {"ts": "2027-02-01T07:00:00+05:00", "trips": 412, "low": 371, "high": 455}
  ]
}
```

Ошибки возвращаются в едином виде `{"code": "...", "message": "...", "request_id": "..."}` с кодами 400, 401, 403, 404, 422, 429 и 500. Пакетная загрузка идемпотентна: повторная отправка пакета с тем же `batch_id` не создаёт дублей.

6. Нефункциональные требования

Категория	Требование
Производительность	Ответ API на чтение не более 500 мс (p95); ежедневный пересчёт прогноза готов до 05:00 местного времени

Категория	Требование
Надёжность	Доступность не ниже 99,5% в месяц; сбой сервиса не влияет на приём оплаты проезда
Безопасность	В сервис не передаются ФИО, ИИН, телефоны и номера карт; TLS 1.2 и выше; ролевая модель доступа; журнал действий пользователей хранится 12 месяцев
Размещение	На серверах в Казахстане или в контуре заказчика; поставка в контейнерах
Интеграция	Без переработки существующих систем: данные поступают пакетной выгрузкой или через API загрузки
Масштабируемость	Новый город подключается настройкой и загрузкой данных; сервис выдерживает не менее 10 городов и 500 маршрутов
Языки и интерфейс	Казахский и русский; последние версии Chrome, Edge, Safari
Объяснимость	Для каждого прогноза видны дата расчёта, границы и факторы, которые на него повлияли
Передача результата	Исходный код, документация API, инструкция по развёртыванию и переобучению модели; права на результат определяются договором

7. Цели пилота

Точность измеряется на данных, которые модель не видела: последние 8 недель истории и 30 дней наблюдения.

Метрика	Цель
Ошибка прогноза по маршруту за день (WAPE)	не более 10%
Ошибка прогноза по маршруту за час (WAPE)	не более 20%
Улучшение к базовому прогнозу «неделю назад»	ошибка ниже не менее чем на 25%
Доля маршрутов пилотного города с прогнозом	100%
Готовность ежедневного прогноза к 05:00	не менее 29 дней из 30
Время ответа API (p95)	не более 500 мс
Рекомендации, признанные перевозчиком применимыми	не менее 70%

WAPE — взвешенная абсолютная процентная ошибка: сумма отклонений прогноза от факта, делённая на сумму факта.

8. Сроки и этапы

Разработка и интеграция занимают 10 недель, затем 30 дней наблюдения за пилотом.

1. Анализ данных и проектирование — 2 недели: аудит выгрузок, базовый прогноз, контракт API.
2. Модель прогноза и проверка на истории — 3 недели.
3. Расчёт рекомендаций по выпуску и сценарии «что если» — 2 недели.
4. API, веб-кабинет и интеграция с выгрузками заказчика — 2 недели.
5. Запуск пилота в одном городе и обучение пользователей — 1 неделя.