Текст презентации

Кирилл: Добрый день, уважаемая комиссия, студенты. Сейчас я, Куприянов Кирилл и мой коллега Суровцев Максим расскажем вам про курсовой проект, который мы разработали в этом году.
Максим: Тема: Клиент-Серверное Приложение для Управления Скидками в Розничных Сетях, а научный руководитель — Александров Дмитрий Владимирович.

Кирилл: В последнее время люди чаще стали покупать товары по скидкам и акциям. Это привело к появлению агрегаторов скидок и купонов, и к созданию новой соответствующей ниши для приложений и программ. Её мы и решили занять, найдя проблему и обосновав актуальность.

Кирилл: В процессе своего выступления мы будем использовать технические термины, которые стоит сразу обозначить.

Максим: Проводя исследование актуальности работы, мы столкнулись с интересной вещью. Статистика популярности запроса “скидки” в интернете имеет сезонность. Оказывается, зимой люди склонны больше покупать товаров по скидке, чем в другие времена года. В целом график показывает рост интереса людей к скидкам.
Кирилл: Второй график иллюстрирует рост популярности крупного аналога — приложения “Едадил”, что говорит о хорошей конкуренции в этой сфере.

Кирилл: Проблем мы нашли много. Бумажные каталоги использовать неудобно — что, если человек хочек посмотреть скидки, сидя дома на диване? Смотреть сайты каждого магазина по отдельности времязатратно, а существующие приложения не обладают достаточным функционалом.

Кирилл: Соответстно, мы задались целью создать модульные и масштабируемые web и Андроид приложения для работы со множеством списков покупок и акциями в розничных сетях.

Максим: На момент утверждения темы курсовой работы существовал один аналог нашего приложения — “Едадил — акции в магазинах (бета)”.

Кирилл: Перед Едадилом наше приложение выделяется удобством использования — у нас предусмотрено неограниченное количество списков покупок, добавление в списки пользовательских позиций, то есть товаров, которых нет на данный момент по скидке, и возможность использования одного аккаунта несколькими людьми. А такие архитектурные решения как модульность, масштабируемость и поддерживаемость выделяют нас как программный продукт. Перейдём к обзору алгоритма работы программы.

Кирилл: Работа нашей программы начинается с кроулера. Он собирает данные о товарах с сайтов магазинов и передаёт серверу. Поскольку кроулер — модульный, и рассматривать его лучше по модулям.

Кирилл: В модуле selectors были использованы xpath селекторы в качестве универсального средства поиска элементов в HTML дереве. Например, данный селектор найдёт все аттрибуты href ссылок с классом catalog-categories__link.

Кирилл: Модуль spiders, или пауки, работает следующим образом. Пуак заходит на сайт магазина, то есть загружает его исходный код. Ищет корневой элемент для всех товаров, и затем, для каждого товара, создаёт объект класса Item, заполняет его поля, и передаёт модулю pipelines, про который я расскажу далее. Спайдер затем переходит на следующую страницу, запускает снова свою процедуру, и так до конца, пока не кончатся страницы.

Кирилл: Каждый объект класса Item поступает на вход модулю pipelines, где одним из важнейших является класс, записывающий его в Базу Данных путём отправки POST запроса на сервер.

Максим: Пожалуй, главной сущностью в базе данных является item. Именно в таблицу item записываются данные о товарах, собранных кроулером. Пользователь может иметь много списков покупок, а список покупок в свою очередь может содержать много товаров магазинов и много пользовательских позиций. Пользовательские позиции из списка покупок отражает таблица custom_item.

Максим: Каждый список покупок может содержать как товары из магазина, так и пользовательские позиции. На первый взгляд, пользовательские позиции — это просто текст, но если среди товаров магазинов найдутся соответствующие позиции, то они будут рекомендованы пользователю.

Максим: Для взаимодействия сервера и клиентов был разработан REST API.
Список магазинов, список категорий для магазина у нас не захардкожены на клиентах — они динамически подгружаются при запуске приложения. Это значит, что если мы захотим добавить новый магазин, нам достаточно будет сделать insert в базе данных, написать спайдер с селекторами для него и запустить (без перезагрузки сервера).
На слайде приведен пример endpoint'а. Голубым цветом выделен id магазина, розовым — параметры запроса. Сервер вернет ответ, содержащий первую страницу товаров категории "Напитки" для магазина с id равным 1.
Товары загружаются постранично для экономии трафика.

Максим: Endpoint'ы для работы со списком покупок должны быть авторизованы. Для авторизации используются JSON Web Token.
Токен состоит из трех частей: header (красный), payload (розовый) и signature (голубой). Header содержит тип токена (это JWT) и способ хэширования, используемый для создания сигнатуры. В payload'е содержится основная информация — это имя пользоватея, его id и timestap создания токена. Именно отсюда сервер берет id и имя пользователя.
Сигнатура токена получается следующим образом: header и payload кодируются в формат base64 и конкатенируются через точку. После этого полученная строка передается в хэш-функцию вместе с секретным ключом.

Максим: Алгоритм авторизации пользователя следующий:
1. Клиентское приложение отправляет POST запрос, в теле которого указывает имя пользователь и пароль, введенные пользователем.
2. Если имя пользователя и пароль верны, сервер возвращает токен.
3. Клиентское прилоежение добавляет токен в заголовок (header) каждого авторизованного запроса.
4. Сервер проверяет сигнатуру токена с помощью секретного ключа, и если она оказалось ва-

лидной возвращает ответ, а иначе возвращает ошибку.

Кирилл: В разработке были использованы: Фреймворк Scrapy и язык Python для создания кроулера и его модулей, PostgreSQL и Sequelize для базы данных и ORM, и фреймворк express для создания REST API. Фреймворки VUE js и bootstrap для веб-клиента, а под Андроид разработка велась с использованием языка Java. Для задания расписания запуска кроулера использовалась стандартная Unix утилита crontab. Для создания программной документации и презентации, была использована система вёрстки текста LaTeX.

Кирилл: И конечно же, помимо серверной части, были реализованы 2 клиента. Их работу мы продемонстрируем на видео.

Кирилл: Поскольку направление разработки достаточно новое и развивающееся, путей развития может быть множество: от увеличения числа магазинов, до создания клиентах на других популярных платформах

Кирилл: Спасибо за внимание! С удовольствием ответим на ваши вопросы.