

«Виртуальный модный помощник в подборе образа». СТО

Реализация всех задумок СРО в виде продуманной архитектуры проекта

В качестве архитектуры приложения была выбрана архитектура MVC (Model – View - Controller)

Причины, по которым была выбрана данная архитектура, следующие:

- 1) чёткое разделение логики представления (интерфейса пользователя) и логики приложения;
- 2) поддержка различных типов пользователей, которые используют различные типы устройств. Модель возвращает одинаковые данные, а единственное различие заключается в том, что контроллер выбирает различные виды для вывода данных;
- 3) концепция MVC существенно уменьшает сложность больших приложений. Код получается гораздо более структурированным, и, тем самым, облегчается поддержка, тестирование и повторное использование решений.

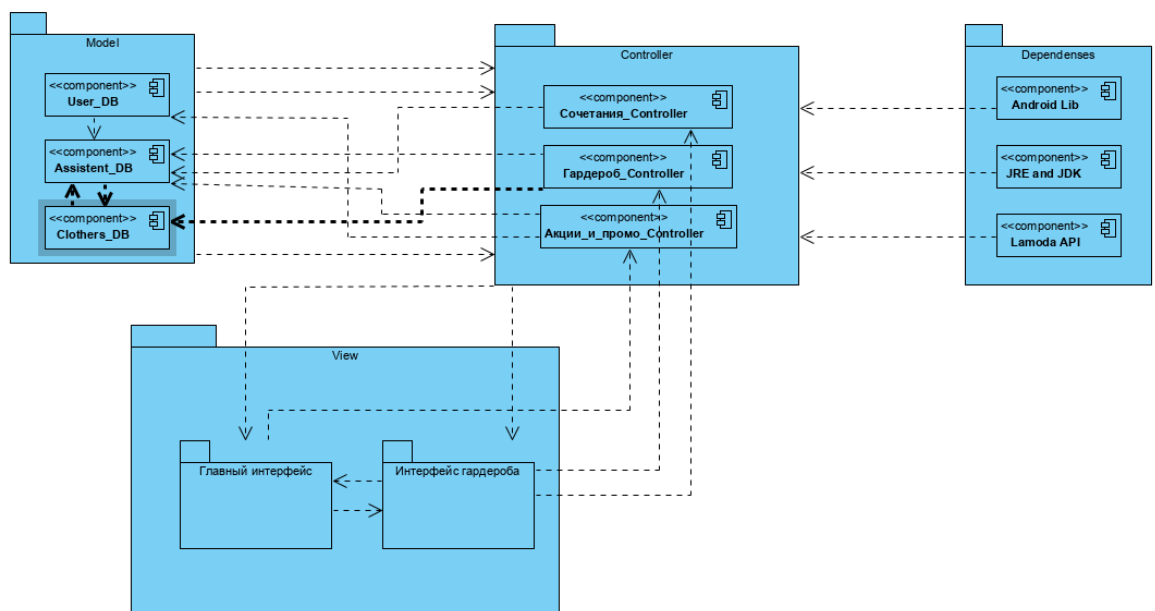
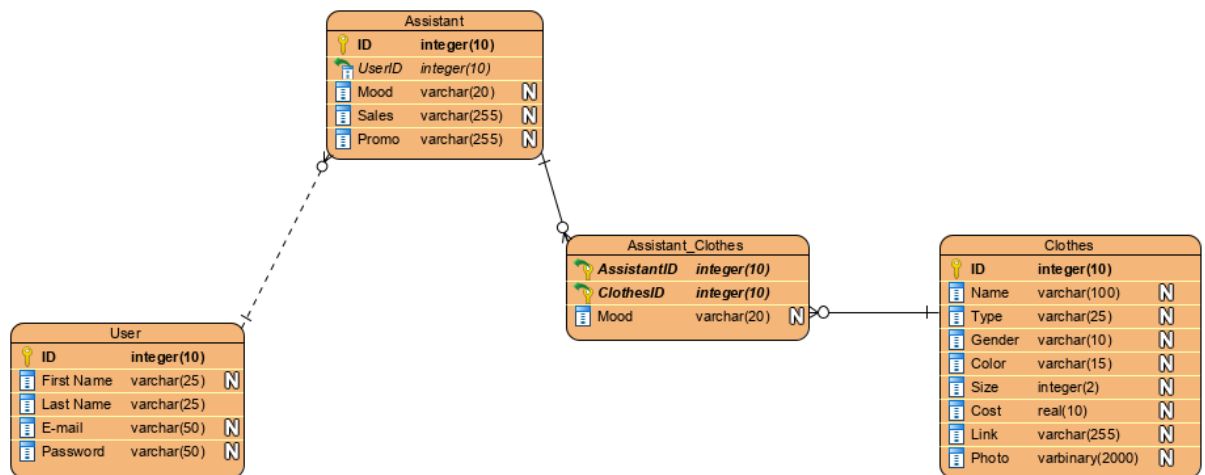


диаграмма компонентов архитектуры MVC

Архитектура хранилища данных

СУБД – Oracle DB



Представленные таблицы:

1. User – пользователь приложения. В БД хранится информация об имени и фамилии, адресе электронной почты и пароле
2. Assistant – помощник пользователя приложения. По настроению подбирает одежду из соответствующей БД, а также сообщает о скидках и промокодах в магазинах одежды
3. Clothes – база данных одежды. Содержит информацию об одежде (название, тип, для какого пола, цвет, размер, цену в магазине и фото с ссылкой на товар в магазине)

Выбор технологий для проекта

Backend: Java

Причины:

- а) Платформенно-независимый язык программирования
- б) Строгая типизация, обеспечивающая безопасность передачи данных в объекты классов
- в) Позволяет писать программы-модули

Рассматриваемые кандидатуры:

Kotlin, Go, Python

Выбор был по большей части обусловлен как популярностью языка программирования, так и широким функционалом, расширениями и библиотек

<https://dev.by/news/12-topovyh-yazykov-mobilnoi-razrabotki-po-versii-ieee>

Frontend: Java Script

Причины:

- а) Удобство пользовательских интерфейсов
- б) Простота и рациональность применения

в) Скорость работы и производительность

Рассматриваемые кандидатуры:

WordPress, Angular, SimTech

Выбор был по большей части обусловлен как популярностью использования технологий, так и широким функционалом, удобством интерфейсов и производительностью

<https://vc.ru/hr/145461-yazyk-programmirovaniya-javascript-osobennosti-i-preimushchestva>

Хранилище данных: Oracle DB

Причины:

а) СУБД Oracle хорошо обрабатывает транзакции. Система сохраняет высокую производительность, в результате чего пользователи не страдают от низкой скорости обработки

б) СУБД обладает высокой степенью готовности. В разных установках, продолжительность работы Oracle индивидуальная

в) СУБД обладает локальной управляемостью

Рассматриваемые кандидатуры:

PostgreSQL, MySQL, SQLite

Выбор был обусловлен хорошей обработкой транзакций, т.к. в приложении много обращений к БД, а также высокой связью с Java и JavaScript

<https://qna.habr.com/q/227515>

API: Lamoda API

Причины:

а) Доступ к открытому исходному коду на гитхаб

б) Возможность расширенной технической поддержки

в) Отсутствие необходимости «Парсинга сайта»

Рассматриваемые кандидатуры:

Wildberries API, Amazon API, Befree API

Выбор был обусловлен благодаря возможностью доступа к открытому исходному коду и отсутствия необходимости парсинга сайтов, что выгодно отличает от предоставленных вариантов

<https://tech.lamoda.ru/>

Релизы

Инструменты создания и автоматизации:

Среда разработки: IntelliJ Idea

Средство создания UML диаграмм: Visual Paradigm

Средство проектирования бизнес-процессов: BPMN Storm

Средство автоматизированного тестирования: TestNG

Релизы:

Основная аудитория – граждане РФ

Релиз проекта: Вторник или Среда за неделю до праздников, в первой половине дня.

Пример 18 мая 2021 года (на неделю до последнего звонка)

Релиз крупных обновлений: Понедельник или Вторник в первой половине дня (Делается это для снижения нагрузки на сервера)

Релиз патчей и хотфиксов: Не позднее 2-3 после релиза обновления, но до выходных (Также для снижения нагрузки на сервера)

Скрипт проведения собеседования

1. Начинаем собеседование с рассказа о компании, почему мы, что делаем — рассказываем релевантно той должности, на которую пришел человек. Помним, что основной мотиватор — это сама работа, поэтому импровизируйте на этом этапе, под каждого человека — пусть он задает вопросы. Расскажите, кого ищете и что ждете от кандидата на данной должности.
2. Далее начинается обратная связь от кандидата: пусть соискатель расскажет свой релевантный опыт на основе полученной информации, т.е. какие проекты или опыт работы в данной сфере у него был и как его опыт пересекается с необходимой для вакансии специальностью.
3. Выслушав кандидата, переходим к списку «Технических вопросов» по теме специальности (например, для QA - знание принципов тестирования и инструментария, а для разработчиков на Java - типы данных, интерфейсы, концепция ООП), фиксируйте ответы для того, чтобы объективно сравнить данного кандидата с другими соискателями.
4. Теперь переходим к «общим вопросам по резюме» — здесь уже делаем двусторонний диалог, вопрос-ответ, уточнение, рассказы за профессиональную жизнь.
5. Прощаемся. На собеседовании не выставляем оценок способностей (Нет школьным оценкам!!), а указываем на сильные и слабые стороны кандидата. Если есть сомнения в компетентности, то нужно дать тестовое задание и обговорив сразу, что кандидат будет делать и на какой в срок он должен прислать ответ.
6. Пишем краткое резюме по данному человеку — плюсы и минусы. Может быть, какие-то свои замечания. По результатам тестового задания даём уже окончательный ответ с указанием на пробелы и необходимостью изучения материалов.