1. Приветствие. Добрый день (Здравствуйте).

Сейчас я представлю вам веб-приложение, разработанное мною за время стажировки под руководством Владимира Некрасова.

2. Тема проекта доска объявлений о покупке, продаже, аренде недвижимости.

3. Функциональность приложения ограничена минимальными потребностями пользователя и простотой использования. В проекте реализовано три пользовательских роли: Гость (незарегистрированный либо не прошедший авторизацию пользователь), Авторизованный пользователь и частный случай авторизованного пользователя – Администратор. Гость может просмотреть информативно ограниченный список всех объявлений, без возможности сортировки и фильтрации. Пройдя регистрацию, гость автоматически становится авторизованным пользователем, что значительно расширяет его возможности. Авторизованный пользователь получает доступ к подробной информации об объявлениях, а также к комментариям. Он может добавлять объявления и комментировать объявления других пользователей. Для удобства поиска необходимого объявления авторизованный пользователь получает доступ к фильтрации объявлений по категориям и сортировке по дате добавления. Администратор вдобавок к перечисленным ранее возможностям имеет право удалять объявления, комментарии и пользователей, а также способен наделить пользователя правами администратора.

4. Использованные технологии и средства

Начну с фреймворка автоматизации сборки проектов Maven. Он позволяет собрать проект, независимо от операционной системы, поддерживает основные среды разработки на java, управляет зависимостями и собирает все необходимые библиотеки в локальный репозиторий.

Для хостинга проекта и взаимодействия с куратором, был выбран, ориентированный на совместную разработку, веб-сервис GitHub. Это так называемая социальная сеть для разработчиков с использованием системы контроля версий git, кроме размещения кода, участники могут общаться, комментировать правки друг друга, следить за новостями знакомых. Для проектов есть личные страницы, небольшие Wiki, система отслеживания ошибок и система управления задачами( issue tracking). Прямо на сервисе можно просмотреть файлы проектов с подсветкой синтаксиса для большинства языков. Для работы с git пользовался инструментом TortoriseGit.

Проект создавался в среде разработки IntelliJ IDEA, это мой первый опыт работы с данной средой, до этого я использовал Eclipse, переход на IDEA не вызвал никаких затруднений, в чем-то IDEA оказалась удобнее.

База данных проекта создана в СУБД MySql. Для управления и администрирования базы данных использовался удобный графический клиент dbForge Studio for MySQL.

Рзработка UML-диаграмм классов, макетов страниц будущего веб-приложения, велась с помощью Enterprise Architect.

До того как внедрить Hibernate работа с базой приложением осуществлялась по средствам JDBC.

Уровень представлений приложения реализован с помощью JSP-страниц.

На JSP страницах для различных операций использовались JSTL-тэги, скриплеты, JavaScript.

Шаблон (паттерн проектирования) DTO(Data Transfer Object) используется для передачи всех необходимых данных на JSP страницу, тем самым снижая количество вызовов методов.

Паттерн Template Method используется в реализации JDBC AbstractDAO, помогает избавиться от дублирования когда,  определяет основу алгоритма и позволяющий наследникам переопределять некоторые шаги алгоритма, не изменяя его структуру в целом, предупреждает  ошибки в классах наследниках.

Архитектура приложения основана на одной из вариаций модели Model-view-controller.  Моделью являются классы сущностей, представлением JSP страницы, в качестве контроллера выступают команды.

Шаблон DAO(Data access object) отделяет логику работы с базой данных от бизнес-логики и логики представления. Что значительно упрощает разработчику сопровождение, тестирование, повторное использование кода, облегчает работу с базой данных, поскольку логика работы с базой сосредоточена в одном слое. В данном приложении существует две реализации DAO: основанная на JDBC и реализованная с помощью Hibernate.

Hibernate. Целью Hibernate является освобождение разработчика от значительного объёма сравнительно низкоуровневого программирования по обеспечению хранения объектов в реляционной базе данных. Разработчик может использовать Hibernate как в процессе проектирования системы классов и таблиц «с нуля», так и для работы с уже существующей базой данных. Hibernate не только решает задачу связи классов Java с таблицами базы данных (и типов данных Java с типами данных SQL), но также предоставляет средства для автоматической генерации и обновления набора таблиц, построения запросов и обработки полученных данных и может значительно уменьшить время разработки, которое обычно тратится на ручное написание SQL- и JDBC-кода. Hibernate автоматизирует генерацию SQL-запросов и освобождает разработчика от ручной обработки результирующего набора данных и преобразования объектов, максимально облегчая перенос (портирование) приложения на любые базы данных SQL.

При помощи JUnit были проведены тесты некоторых DAO классов.

I18n. В данном приложении реализована довольно просто, поддерживается языка (русский и английский), переводы находятся в файлах свойств.

Аутентифика́ция (англ. Authentication) — процедура проверки подлинности.

проверка подлинности пользователя реализована путём сравнения введённого им пароля с паролем в базе данных пользователей; Аутентификацию не следует путать с авторизацией.

Авториза́ция (от англ. authorization — разрешение, уполномочивание) — предоставление определённому лицу или группе лиц прав на выполнение определённых действий; как я уже говорил в приложении реализовано три роли пользователей, в зависимости от роли пользователя предоставляются определенные права.

Spring. универсальный фреймворк с открытым исходным кодом для Java-платформы. Центральной частью Spring Framework является Inversion of Control контейнер, который предоставляет средства конфигурирования и управления объектами Java . Контейнер отвечает за управление жизненным циклом объекта: создание объектов, вызов методов инициализации и конфигурирование объектов путем связывания их между собой.

Объекты создаваемые контейнером также называются Управляемые объекты или Beans. Обычно конфигурирование контейнера осуществляется путем загрузки XML файлов, содержащих Определение Bean’ов и предоставляющих информацию необходимую для создания bean’ов. Внедрение зависимости — шаблон проектирования, когда контейнер передает экземпляры объектов по их имени другим объектам либо с помощью конструктора, либо свойства, либо фабричного метода. Наиболее популярной реализацией IoC является Dependency Injection Principle (Принцип внедрения зависимостей). В данном проекте используется Dependency Injection.

5. Проектирование приложения была начато с построения UML-диаграмм сущностей предметной области в Enterprise Architect, после этого по диаграмме были сгенерированы классы будущего приложения, которые постепенно изменялись и совершенствовались.

6. Позже была спроектирована база данных.

Чтобы описать работу всего приложения вполне достаточно пояснить назначение трех его классов: RealtyServlet, ServletFilter, CommandFactory…