

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ЦЕЛЕЙ ОЦЕНКИ УНИКАЛЬНЫХ ОБЪЕКТОВ

Павлова В.В., Карасева М.Г., Прокопчук М.А.

*Белорусский национальный технический университет, г. Минск, Беларусь,
v.pavlova@bntu.by, m6668358@gmail.com*

Данная статья посвящена разработке программного комплекса, который будет решать проблему поиска аналогов для уникальных объектов оценки. В настоящий момент современное программное обеспечение не в состоянии выполнить эту задачу. Авторами делается попытка предложить оценщикам продукт, призванный облегчить их работу при оценке редко встречающихся на вторичном рынке объектов.

Введение. На настоящем этапе в современных программных комплексах в области оценочной деятельности очевидна проблема недостаточного обеспечения программными продуктами для *редко встречающихся на вторичном рынке объектов*. Поэтому целью авторов данной статьи являлось выявление нереализованного в современных программных комплексах в области оценочной деятельности полезного функционала и его реализация в виде нового программного продукта. Элементом практической значимости проведенного исследования является разработанный программный комплекс, обеспечивающий возможность поиска упомянутых выше объектов.

Назначением разрабатываемой системы является предоставление пользователям инструмента для поиска аналогов для уникальных объектов, поиск аналогов для которых на сегодняшний момент является достаточно трудоемким.

Целью создания является уменьшение трудозатрат оценщиков на поиск аналогов, редко встречающихся на рынке объектов оценки. Соответственно, пользователями разрабатываемой системы станут оценщики.

Для обеспечения доступа пользователям к системе, система создается в виде веб-приложения.

Функционально система будет представлять из себя сайт с системой авторизации обладающей тремя ролями:

- администратор
- авторизированный пользователь
- гость.

Для гостя будет предусмотрена возможность просмотра информации по уже проведенным работам по оценке, а так же поиск по заданным параметрам.

После прохождения авторизации гость становится либо пользователем либо администратором.

Если ролью после авторизации является пользователь, то после прохождения авторизации открывается список работ по оценке, которые добавил данный пользователь. Также на странице присутствуют интерфейсы для фильтрации этих работ по заданным критериям и инструменты изменения или удаления работ данного пользователя и добавления новой работы.

Для администратора, в свою очередь, предусмотрен список пользователей, с возможностью фильтрации по заданным критериям и инструменты добавления, изменения и удаления пользователей. Вынесение инструмента создания новых пользователей исключительно в функционал администратора позволит осуществлять контроль за базой пользователей и решит проблему обхода оплаты за пользование

системой через погрешности в системе безопасности через такие вэб-клиенты как Postman, если в перспективе видом монетизации станет оплата за пользование сервисом.

В то же время не предусматривается система защиты паролей пользователей с помощью алгоритмов шифрования для минимизации трудозатрат на создание рабочего прототипа. Соответственно, при необходимости запуска рабочего сервиса после стадии тестирования прототипа в рабочей среде, необходимо разработать систему шифрования аутентификационных данных пользователя и систему восстановления пароля.

Также для ролей администратора и пользователя предусмотрен механизм выхода из личного кабинета и изменения их роли в системе на гость. В свою очередь, на гостевой странице предусмотрен механизм авторизации.

Главным элементом всей системы является так называемая работа по оценке, далее – оценка. Она состоит из следующих элементов:

- оценщик
- дата оценки
- вид объекта оценки
- наименование объекта оценки
- вид стоимости
- валюта оценки
- итоговая стоимость
- характеристики объекта оценки
- общее описание оценки

Общее описание оценки представляет из себя описание произвольной формы хода проведения оценки и может включать в себя описание найденных аналогов и непосредственного расчета стоимости.

С функциональной точки зрения данный сервис будет поддерживать работу с различными видами объектов оценки транспорта, для каждого из которых будет существовать как набор собственных уникальных характеристик, так и некоторый набор общих характеристик. К общим характеристикам относятся: год выпуска, тип трансмиссии, объем двигателя, мощность двигателя.

Данные характеристики доступны для просмотра после перехода со страницы со списком оценок (как гостя, так и пользователя) на страницу подробного описания оценки. Эта страница включает в себя информацию по всем элементам оценки, в том числе для характеристик объекта оценки включает в себя общие и уникальные характеристики.

Для удобного проведения выборок по оценкам будет реализован механизм фильтрации. Для общего списка со страницы гостя будет реализована фильтрация по дате оценки, виду объекта оценки, виду определяемой стоимости, наименованию объекта оценки и общим характеристикам объектов оценки.

Фильтрации по уникальным характеристикам объектов оценки осуществляются после выбора из выпадающего списка для фильтрации по виду объекта оценки.

Для просмотра списка оценщиков на странице гостевого пользователя необходимо предусмотреть механизм перехода на данный список. С этого списка, в свою очередь, осуществляется переход на список оценок данного оценщика.

Для пользователя ранее были предусмотрены механизмы создания новой оценки и редактирования уже существующей оценки данного пользователя. В ходе использования этих механизмов осуществляется заполнение формы с элементами оценки описанными выше и последующим сохранением этих данных в базе.

Так ранее для администратора были предусмотрены механизмы добавления и изменения пользователей, необходимо определить данные механизмы. При добавлении нового или редактировании уже существующего пользователя администратор заполняет поля логина и пароля пользователя, после чего сохраняет изменения в базу.

Для всех механизмов, которые вносят в базу данных изменения, следует предусмотреть способ выхода из данного механизма без внесения изменений.

Для поддерживаемых сервисом видов оценки предусматривается существование уникальных характеристик, соответственно необходимо предусмотреть виды объектов оценки и их уникальные характеристики.

Виды объектов оценки, поддерживаемые сервисом:

- самоходные краны
- авторефрижераторы
- пожарные автомобили общего применения

Для самоходных кранов уникальными характеристиками являются:

- максимальная грузоподъемность
- максимальный вылет стрелы
- количество осей
- общий балласт

Для авторефрижераторов уникальными характеристиками являются:

- объем холодильной установки
- мощность холодильной установки
- грузоподъемность

Для пожарных автомобилей общего применения уникальными характеристиками являются:

- вместимостью цистерны
- напор насоса

Для дальнейшего продвижения сервиса, в случае удачного тестирования прототипа, можно разработать следующий функционал:

- механизм добавления новых видов объектов оценки
- механизм расширения прав доступа к инструментам сервиса
- API для сторонних сервисов
- Механизм интеграции со средами для работы с массивами данных
- Внедрение искусственного интеллекта для поиска аналогов

С точки зрения дизайна, необходимо использовать светлые цветовые решения и размещать инструменты таким образом, чтобы соответствовать паттерну вертикального проектирования пользовательских интерфейсов.

Базы данных необходимо проектировать в соответствии с второй нормальной формой. Это условие необходимо для построения базы данных таким образом, чтобы последующее расширение списка видов объектов оценки не привело к необходимости переработки архитектуры базы данных и упрощения работы с базой данных. Также подобная система упростит модель сущностей сервиса, что благотворно скажется на возможности масштабирования системы и расширения видов объектов оценки.

Разработка данного программного комплекса в соответствии с данными требованиями позволит создать легко масштабируемый прототип данного комплекса, что

позволит уменьшить риски по части не окупаемой разработки, и в то же время позволит использовать основу программного кода для разработки действующего сервиса.

Выбор инструментов разработки. Для разработки программного комплекса, описанного в предыдущей главе, необходимо описать основные модули, из которых будет состоять данный комплекс с точки зрения архитектуры программных компонентов.

На сегодняшний день одним из самых популярных шаблонов проектирования приложений является шаблон MVC (model-view-control).

Данный шаблон разбивает приложение на 3 слоя:

- Модель – это слой сущностей, которые использует программа во время своей работы. В случае данного программного комплекса это оценки и роли.
- Слой визуализации – это слой пользовательских интерфейсов.
- Слой контроллеров – этот слой связывает модель и слой визуализации. Также это слой программной логики.

Для взаимосвязи компонентов приложения воспользуемся шаблоном проектирования IoC (inversion of control). Это позволит избежать жестких связей между компонентами приложения, что позволит легко масштабировать систему и заменять отдельные слои в случае какой-либо необходимости. Для реализации этого шаблона воспользуемся фреймворком Spring MVC.

Выбор данного фреймворка также подводит к необходимости выбора Java как основного языка программирования. Данный язык отлично подходит для создания приложений, одним из основных требований к которым является возможность последующего использования программного кода и масштабируемость.

Так как данный программный комплекс использует базу данных для хранения информации о проведенных оценках, необходимо определиться с конкретной реализацией базы данных. В данном случае, для начала, необходимо определиться с выбором типа базы данных. Так как в основе приложения лежит работа с классами объектов и характеристиками данных классов, то наиболее подходящая система управления базами данных (далее СУБД) – объектно-реляционная СУБД. В качестве конкретной реализации используем PostgreSQL.

Для взаимосвязи Java классов модели и выбранной СУБД воспользуемся технологией ORM (object-relation mapping). Данная технология описывает как должны отображаться данные классов в базе данных. В качестве конкретной реализации используем Hibernate как самый популярный ORM фреймворк.

Для обеспечения разделения логики приложения по ролям и доступа к инструментам в соответствии с ролью необходимо реализовать механизм аутентификации пользователей. Для реализации этого функционала воспользуемся фреймворком Spring Security. Этот инструмент идеально интегрируется с Spring MVC, выбранным ранее как основной инструмент взаимосвязи компонентов приложения. Как уже отмечалось ранее, от использования механизмов шифрования аутентификационных данных пользователей отказываемся ради снижения трудозатрат на разработку. Данный функционал позднее будет добавлен достаточно просто благодаря использованию шаблона IoC.

Для отслеживания работы приложения воспользуемся логированием. Это механизм создания записей в так называемые лог-файлы. Эти записи описывают работу системных компонентов, например, регистрируют вызов функций или факт аутентификации пользователя в системе. В качестве конкретной реализации используем SLF4J. Это надстройка над самым популярным инструментом логирования – log4j.

Для взаимосвязи пользовательского интерфейса и ядра системы используем Servlet API. Это позволит не только связать эти слои, но и обеспечить вместе с Spring Security контролируемый доступ к данным из внешних систем. Так как вся система завязана на Spring Framework, то и данную реализацию стоит использовать из данного пакета.

Для реализации пользовательского интерфейса используем HTML, JSP, JSF и JavaScript. Данный список инструментов позволит создать полноценный пользовательский интерфейс, рассчитанный на работу в браузерах.

Чтобы гарантировать работоспособность системы, покроем ее тестами. Для этого используем фреймворк JUnit.

В качестве контейнера приложений используем Tomcat, как наиболее легковесный контейнер и наиболее простой в использовании, с учетом того, что приложение не требует сложных серверных настроек.

Для сборки нашего приложения в используемый для деплоя war-архив используем Maven как основной инструмент сборки.

Заключение. Разработанный программный комплекс является только прототипом для тестирования в рабочих условиях. Тестирование предполагает использование комплекса оценщиками и выявление ошибок в работе системы, а также определение иных недостатков.

Также, если этап тестирования будет пройден удачно, на этапе разработки рабочей версии стоит учесть и иные пожелания оценщиков. Позднее стоит задумываться о распространении программного комплекса за пределы Республики Беларусь и, возможно, внедрение его в иной, более крупный комплекс.

ЛИТЕРАТУРА

1. Постановление Госстандарта Республики Беларусь от 13 января 2017 г. № 4СТБ 52.6.01-2017 «Оценка стоимости объектов гражданских прав. Оценка стоимости транспортных средств»
2. Приказ Госкомимущества от 25.08.2015 № 184 ТКП 52.6.01-2015 «Оценка стоимости объектов гражданских прав. Оценка стоимости дорожных транспортных средств»
3. Приказ Белорусского бюро по транспортному страхованию от 11 августа 2004 г. № 29-од «Об утверждении Правил определения размера вреда, причиненного жизни или здоровью потерпевшего в результате дорожнотранспортного происшествия, для целей обязательного страхования гражданской ответственности владельцев транспортных средств».
4. Указ Президента Республики Беларусь от 13 октября 2006г. № 615 «Об оценочной деятельности в Республике Беларусь».