

УДК 004

ИНФОРМАЦИОННАЯ СИСТЕМА ВИЗУАЛИЗАЦИИ АССОРТИМЕНТА МАГАЗИНОВ ПО ПРОДАЖЕ СТРОИТЕЛЬНЫХ МАТЕРИАЛОВ И КОМПОНЕНТОВ

Хваталин Максим Михайлович

магистрант

Мордовский государственный университет им. Н.П. Огарёва, Саранск

author@apriori-journal.ru

Аннотация. В статье рассматриваются основные проблемы процесса демонстрации своего ассортимента магазинами по продаже строительных материалов и компоненты, проблемы проектирования АИС. Обзор инструментов для программной реализации.

Ключевые слова: автоматизация; Ruby; JavaScript; 3D моделирование; СУБД PostgreSQL, хранилище типа ключ-значение Redis.

INFORMATION VISUALIZATION STOCK OF BUILDING MATERIAL SHOPS SYSTEM

Khvatalin Maxim Mikhailovich

undergraduate

Mordovian state university of N.P. Ogaryov, Saransk

Abstract. The article considers problems of building material shop's stock demonstration, also problems of AIS engineering. Reviewing instruments for programming realization.

Key words: automation; Ruby; JavaScript; 3D modeling; DBMS PostgreSQL, key-value storage Redis.

Практически любой человек, затеявая ремонт в собственной квартире, офисе и других помещениях испытывает трудности с подбором строительных материалов и компонентов – мебели, обоев, плинтусов, потолков и др. Очень часто довольно трудно подобрать соответствующую комбинацию, учитывая размеры своего помещения, уровня освещенности, вкусовых предпочтений.

Рассмотрим существующее программное обеспечение, позволяющее решать указанную проблему. Их можно разбить на несколько категорий:

- 1) Моделирование строений, мебели, интерьера – Google SketchUp, Artlandis Studio.
- 2) Моделирование, анимация, рендеринг, постобработка. Blender, 3Ds Max, Autodesk Maya.
- 3) Проектирование архитектурно-строительных конструкций и решений, а также элементов ландшафта, мебели – ArchiCAD.

Практически все, указанные продукты, являются платными. И зачастую их цена больше нескольких десятков тысяч рублей, что сразу отмечает их использование вне дизайн-студий, кроме того они требуют профессиональных навыков работы.

Однако, даже если указанные выше недостатки могут показаться несущественными, есть недостаток и серьезнее – для того, чтобы смоделировать интерьер необходимы 3D модели, которые придется либо самостоятельно создавать, либо где-то искать в потоках информации всемирной паутины. И то, и другое потребует множество затрат времени, не говоря уже о том, что вероятность найти в магазинах желаемые строительные материалы и компоненты крайне низка.

Указанные инструменты не предполагают создание приложений для покупателей с простым интерфейсом и встроенными моделями ассортимента магазина.

Некоторые магазины имеют свои сайты с возможностью просмотра в 3D своего ассортимента. Однако, все они используют устаревшую по

сегодняшним меркам технология Adobe Flash и слишком узкий функционал – просмотр 3D модели без возможности их комбинации.

Для решения рассмотренной проблемы и ликвидации указанных недостатков было решено разработать онлайн сервис. Он будет представлять собой web-приложение, использующее современные web-технологии, предоставляющий доступ магазинам за небольшую постоянную плату.

Магазины смогут, перейдя в личный кабинет, загружать трехмерные модели своего ассортимента в любом из предложенных форматов трехмерных графических файлов, а также модели помещений при необходимости. Затем получив в личном кабинете постоянную ссылку, добавить её на свой сайт.

Покупатели, перейдя по ссылке, смогут выбрать помещение, либо сформировать своё по заданным характеристикам, а затем смоделировать интерьер из необходимых материалов и компонентов в интерактивном режиме с автоматическим соблюдением пропорций.

Рассмотрим преимущества.

Преимущества для магазинов:

- 1) Не нужно тратить деньги на разработку такого сайта с таким функционалом, не нужно тратить деньги на его поддержку.
- 2) Простота работы.

Преимущества для покупателей:

- 1) Бесплатность.
- 2) Минимальные затраты времени.
- 3) Требуются лишь базовые навыки работы с компьютером, не сложнее использования социальных сетей.
- 4) Доступ из любой точки Земли, где есть быстрый доступ к сети интернет.
- 5) Нет привязки к операционной системе, требуется лишь любой современный браузер – кроссплатформенность.

Для достижения поставленной цели необходимо решить следующие задачи:

- Сформировать функциональные требования к АИС;
- Спроектировать архитектуру;
- Подобрать необходимый инструментарий;
- Реализовать поставленные функциональные требования.

Функциональные требования удобно изобразить в виде диаграммы «Прецедентов» [1].

Архитектура системы представляет собой, практически ставшей на сегодняшний день классической, четырехслойную архитектуру. В качестве ключевых слоев можно выделить frontend и backend.

Frontend представлен Single Page Application, осуществляющий обмен с backend по протоколам HTTP и WebSocket, используя формат обмена JSON.

Backend представлен проксирующим сервером, API и WebSocket серверами, а также хранилищами данных.

API сервер написан на каркасе разработки Ruby on Rails, используя язык программирования Ruby и архитектурный паттерн MVC, что позволило добиться высокой степени декомпозиции, скорости разработки. WebSocket сервер написан также на языке Ruby, с использованием языка C путем связывания, т.к. к неоспоримым достоинствам языка C по сравнению с Ruby относится быстрота выполнения программ [2], что крайне важно для realtime.

В качестве хранилища данных выбрана реляционная БД [3] для хранения основных данных, а также хранилище типа ключ-значение для хранения сессий, кэширования, организации взаимодействия между различными процессами. PostgreSQL обеспечивает высокую производительность, поддерживает множество типов данных, включая массивы, json, координаты, а также зарекомендовал себя с хорошей стороны в крупных проектах.

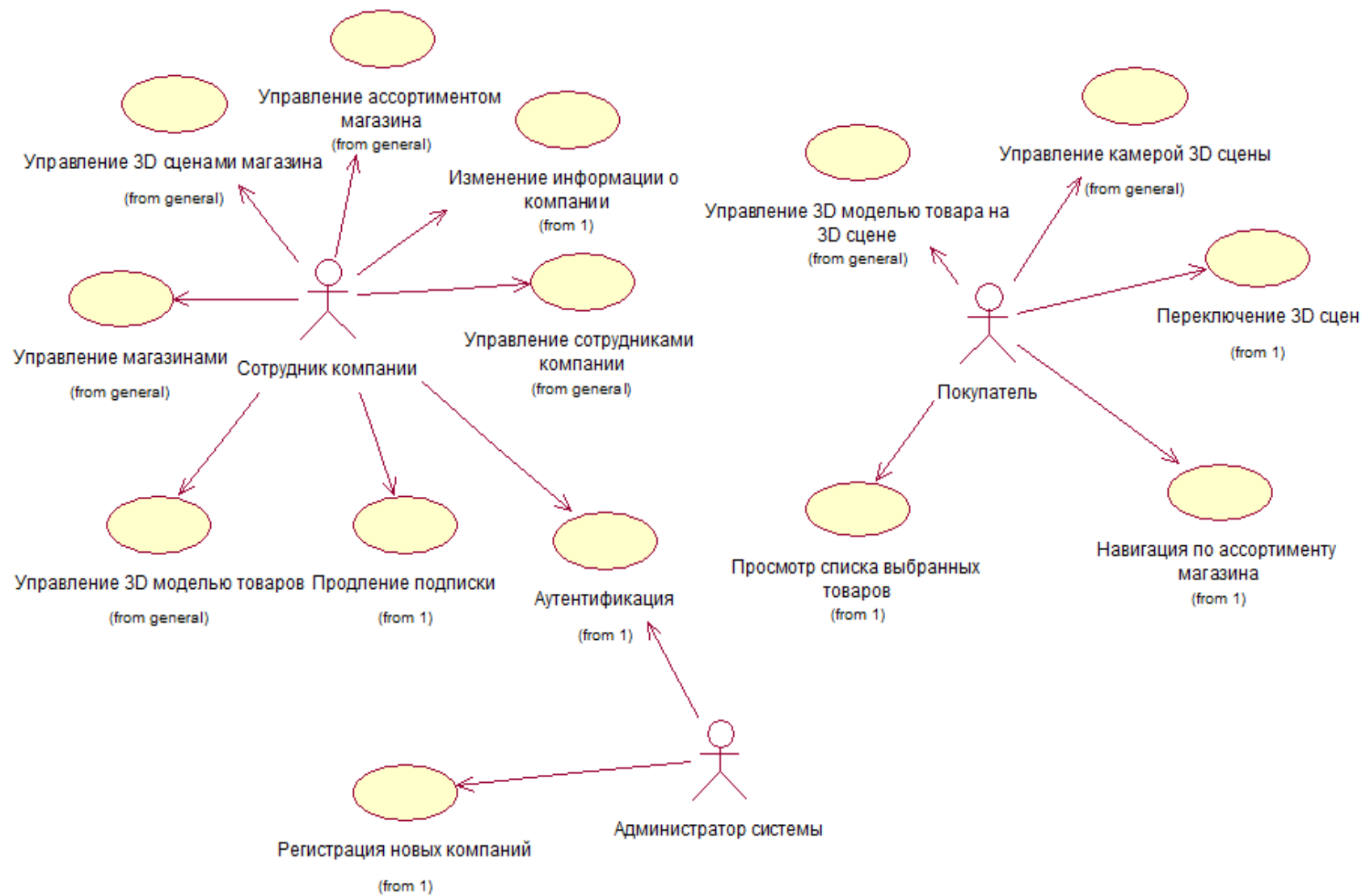


Рис. 1. Диаграмма прецедентов

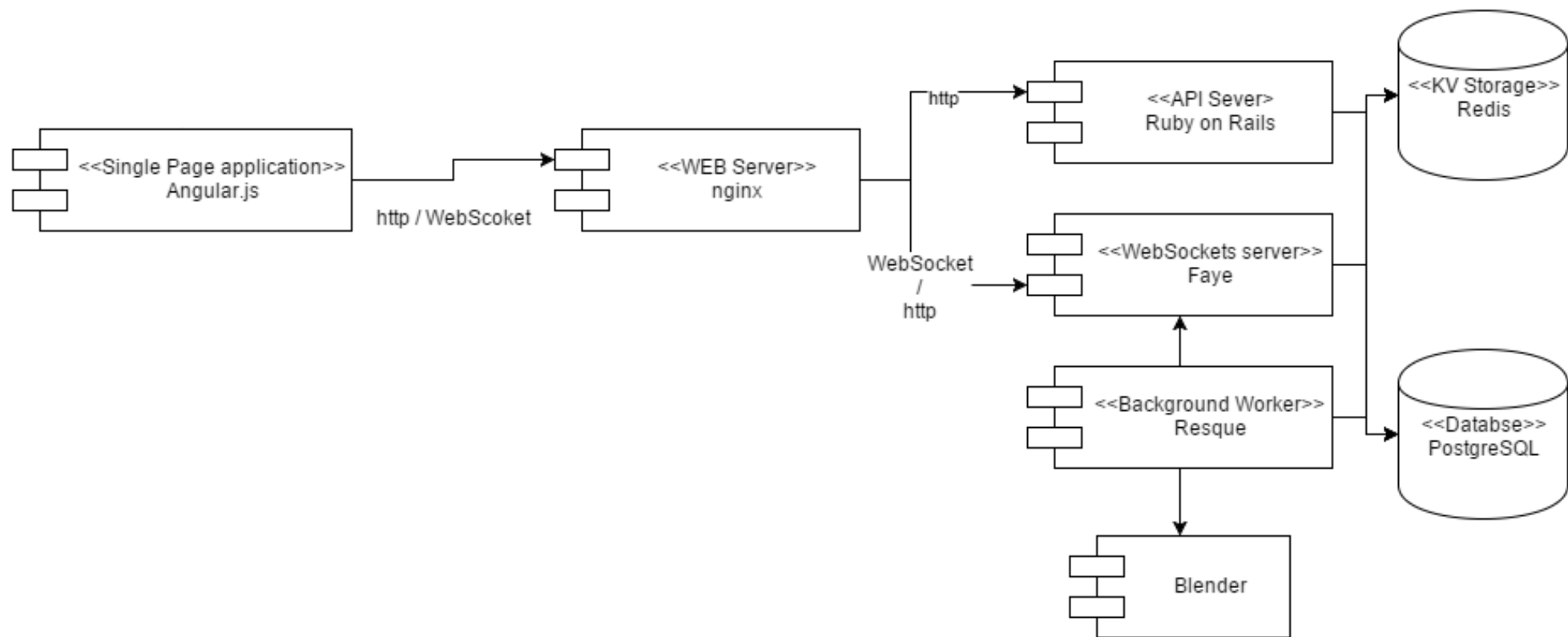


Рис. 2. Архитектура системы

Один из важных моментов при проектировании архитектуры АИС – вопрос каким образом будет организовано выполнения отложенных задач. Этот вопрос не обошел и данный проект, необходимо проводить конвертацию 3D объектов из одного формата в другой для дальнейшей обработки. Выполнение фоновых задач организовано с помощью библиотеки Resque. Все рабочие Resque выполняются в отдельных процессах, что позволяет не блокировать основной веб-сервер приложения. Взаимодействие между сервером приложения и рабочими Resque выполняется с помощью хранилища Redis используя механизм public-subscribe.

Список использованных источников

1. Маглинец Ю.А. Анализ требований к автоматизированным информационным системам. М.: Интернет-университет информационных технологий, 2008. 200 с.
2. Александров Э.Э. Программирование на языке C в Microsoft Visual Studio 2010 / Э.Э. Александров, В.В. Афонин. Саранск: Изд-во Мордовского ун-та, 2010. 424 с
3. Аббакумов А.А., Акимов В.Л., Егунова А.И., Лещанкин К.А., Таланов В.М. Базы данных (MS ACCESS, MYSQL). Саранск: Изд-во Средне-волжского математического общества, 2011.