

А. В. Марков, А. П. Магель

РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ВЗАИМОДЕЙСТВИЯ ЗАИНТЕРЕСОВАННЫХ СТОРОН В ЖИЗНЕННОМ ЦИКЛЕ ЖИЛОГО ДОМА

14

Наличие подходящего веб-приложения (полноценной программы, которая взаимодействует с пользователями (стейкхолдерами), выполняет их требования и возвращает ожидаемый результат) значительно упрощает процесс мониторинга строительства жилых домов. Рассмотрены ключевые заинтересованные стороны, участвующие в жизненном цикле жилого дома, разработан интерфейс веб-приложения и функционал для связи базы данных с модулями данного приложения.

The availability of a suitable web-application (a full-fledged program that interacts with users (stakeholders), meets their requirements and returns the expected results) greatly simplifies the process of monitoring the construction of residential building. The main stakeholders involved in the life cycle of the residential building are considered, the interface of the web-application and the functionality for linking the database with the modules of this application are developed.

Ключевые слова: заинтересованные стороны, интерфейс, клиент-сервер, база данных, интегрированная среда разработки.

Keywords: stakeholders, web-interface, client-server, database, integrated development environment.

Введение

Стейкхолдер (заинтересованная сторона) – понятие, которое описывает человека, группу лиц или отдельные организации, чьи действия, поведение или решения могут влиять на успешность системы. Каждый стейкхолдер может как улучшить результат проекта на отдельно взятом этапе, так и полностью разрушить его, что приведет к завершению проекта без достижения основной цели.

Благодаря концепции управления с позиции стейкхолдеров становится понятно, что если не уследить за выполнением какого-либо этапа строительства жилого дома или не обговорить все детали с внешними партнерами и работниками своей компании, то можно получить незапланированный результат – как положительный, так и отрицательный. В первом случае результат все же будет получен, хотя и другим путем, а во втором вся работа будет проделана напрасно.

Поэтому важно выстроить работу таким образом, чтобы точно знать, какой стейкхолдер и в какой степени участвует в процессах реализации проекта. Тогда можно будет спланировать его результат и заранее быть уверенным, что проект не войдет в число провальных.



Следовательно, целью работы является написание подходящего веб-приложения, которое позволило бы значительно упростить процесс взаимодействия стейкхолдеров друг с другом при реализации проектов строительной компании.

Для достижения поставленной цели необходимо следующее:

- 1) выделить ключевые заинтересованные стороны в осуществлении строительных проектов;
- 2) рассмотреть жизненный цикл жилых домов и выделить основные этапы;
- 3) построить структуру базы данных и настроить бизнес-логику;
- 4) разработать функционал для связи базы данных с модулями веб-приложения;
- 5) разработать интерфейс веб-приложения;
- 6) настроить связь между интерфейсом и модулями.

Описание разработанного веб-приложения

Для веб-приложения была выбрана стандартная структура разработки, когда перед пользователем в браузере загружается содержимое страницы, в котором посредством выполнений некоторых исполняемых триггеров осуществляются HTTP-запросы к серверу. В это время серверная часть приложения с помощью созданных модулей и функционала преобразует через систему управления базами данных загруженную информацию после получения новых запросов на изменение той или иной информации.

Сначала нужно проанализировать информацию, которая будет храниться в базе с данными, и выявить участников будущих объектов, которые каким-либо образом могут влиять на этапы строительства. Для этого мы воспользовались теорией Эдуарда Фримана: он описывает такие процессы, применимые к стейкхолдерам, как анализ и управление, и выделяет 6 основных этапов (рис. 1).



Рис. 1. Теория Фримана

После проведенного анализа были выделены следующие стейкхолдеры: администратор; учредитель; инвестор; поставщик.

Изучив проекты жилых домов, мы определили 13 ключевых этапов строительства: техническое задание; эскиз; проект; анализ; рабочая до-



кументация; подготовка участка под строительство; земляные работы; укладка фундамента; строительство внешних стен; подвод коммуникаций; обустройство крыши; завершающие этапы; обустройство придомовой территории.

Собрав и отсортировав все важные сведения, мы построили диаграмму вариантов использования нашего приложения основными группами стейкхолдеров, которая представлена на рисунке 2.

16

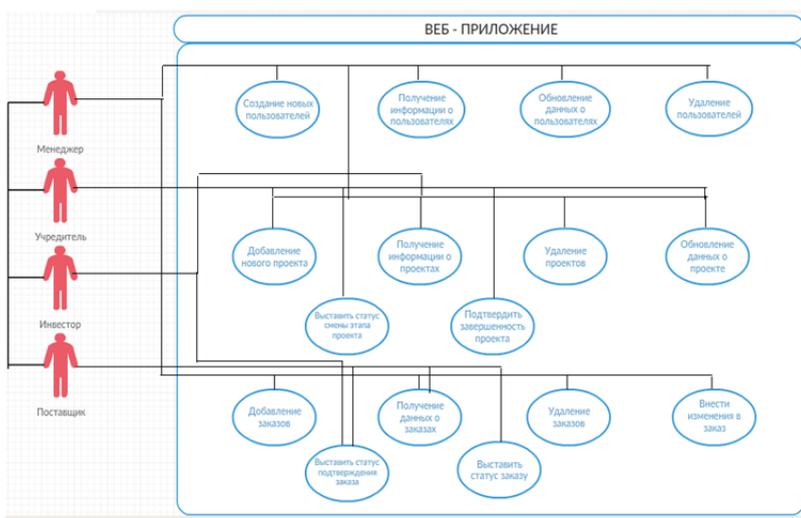


Рис. 2. Диаграмма вариантов использования веб-приложения

После определения первичных функций для всех стейкхолдеров, начинается строительство структуры БД. Для визуального проектирования БД использовался инструмент MySQL Workbench, интегрирующий проектирование, моделирование, создание и эксплуатацию БД в единое бесшовное окружение для системы баз данных.

Данный инструмент имеет следующие преимущества:

- графический режим представления БД;
- наглядный и функциональный механизм установки связей между таблицами, в том числе «многие ко многим» с созданием таблицы связей;
- Reverse Engineering – восстановление структуры таблиц из уже существующей на сервере БД (связи восстанавливаются в InnoDB, при использовании MyISAM связи необходимо устанавливать вручную);
- удобный редактор SQL-запросов, позволяющий сразу же отправлять их серверу и получать ответ в виде таблицы;
- возможность редактирования данных в таблице в визуальном режиме [15–18].

Благодаря MySQL Workbench реализованные таблицы взаимосвязаны в единую систему БД. Расписаны все основные аспекты: информация о проектах, статусы завершенности строительного этапа того или



Рис. 4. Компиляция кода

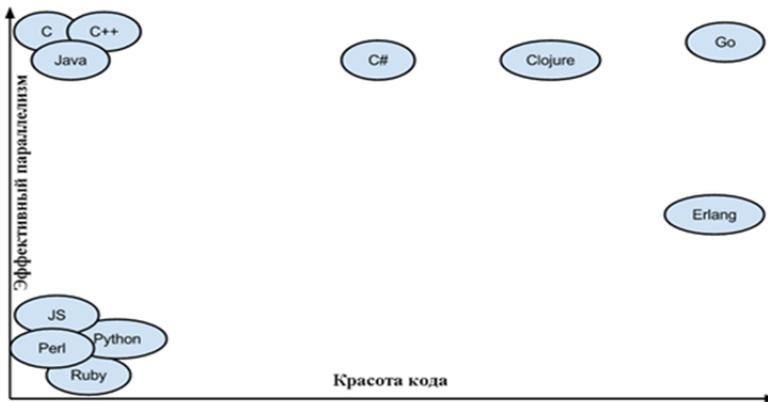


Рис. 5. Эффективность и «красота» языков программирования

После установки GoLang необходимо сделать следующее:

- настроить среду;
- установить и настроить специальный драйвер для связи с MySQL;
- заранее настроить подключение к базе данных на основе MySQL Workbench;
- после подключения создать несколько поступающих запросов (handlers), а также структуры для групп пользователей, проектов, заказов. Важно помнить, что у Go отсутствуют привычные классы. Используя запросы СУБД (вставка, удаление, выбор, обновление и т.д.) и технологии запросов POST, GET, PUT и DELETE, прописываем основной функционал приложения [2; 4–6; 12; 16; 17].

Чтобы разработать веб-интерфейс, воспользуемся HTML5 (стандартный язык разметки документов) и CSS3 (таблица стилей) с специальной библиотекой jQuery для описания сайта.

Макет страниц имеет фиксированную ширину по левому выравниванию с применением отступов и состоит из четырех основных частей: header, навигационная часть, контент и footer (рис. 6).

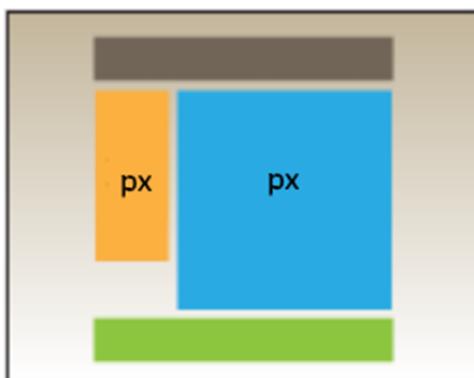


Рис. 6. Основной макет

В части header содержатся данные пользователя, роль и Ф.И.О. В навигационной части содержится список веб-страниц, доступных пользователю по его роли. В контенте располагается система запросов на тот или иной заказ, проект, пользователя, реализующие просмотры, обновления, изменения, удаление полученной информации с баз данных для взаимодействия между пользователями. Footer используется для информации о строительной компании.

С подключаемой библиотекой jQuery JavaScript получает JavaScript-плагины, а также ряд преимуществ: поддержку; кроссбраузерность; документацию; быструю скорость написания кода; легкость понимания.

Веб-интерфейс помогает осуществлять связь с сервером пользователя, который в дальнейшем изменит, удалит, создаст информацию, отображающуюся для всех пользователей или определенного круга лиц в обновленном виде (рис. 7).

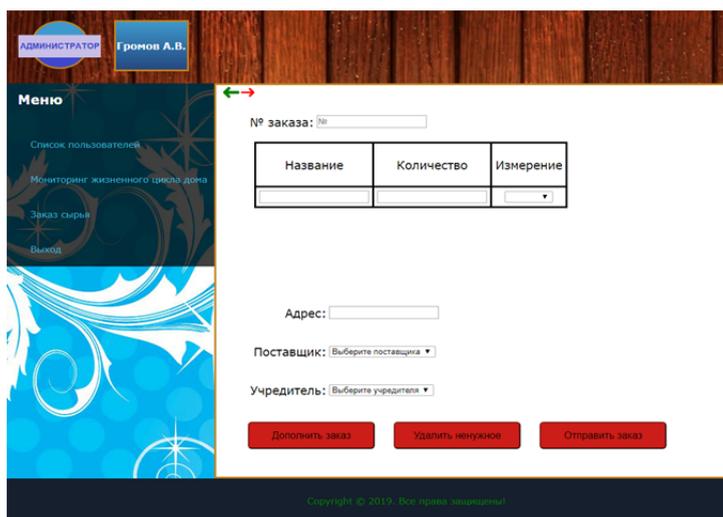


Рис. 7. Веб-сайт для создания заказов



Дополнительные функции в интерфейсе помогают сортировать отображаемую перед пользователем информацию. Сортировка происходит по принципу скрывания текста, который не удовлетворяет тем или иным признакам, хранящимся в базе данных [1; 3; 7–11; 13; 14].

Связываем функционал и интерфейс после написания. Для их тестирования и разработки использовались заглушки. Само веб-приложение использует архитектуру «клиент – сервер». В качестве клиента выступает браузер, в то время как сервер находится в строительной компании. Основной посыл такой архитектуры заключается в распределении обязанностей между сервером и клиентом: хранение, обработка данных являются преимущественно серверной обязанностью, в то время как клиент может удалять, добавлять, обновлять данные. Сам обмен данных происходит по сети. Ответ сервера на запрос изображен на рисунке 8.



Рис. 8. Представление результата выполнения запроса без интерфейса

На примере процесса работы над заказами рассмотрим связь и взаимодействие основных стейкхолдеров. Пусть на текущем этапе строительства требуются строительные материалы. Администратор создает новый заказ, которому присваивается порядковый номер. В заказе он прописывает все необходимые материалы, их количество, адрес, по которому нужно их доставить, обязательно указывает учредителя и потенциального поставщика.

Затем учредитель из своего личного кабинета просматривает заявки, чтобы внести в них коррективы, дать согласие на заказ или отправить заявку на доработку администратором со своим комментарием по заказу.

Если заявка была одобрена, потенциальному поставщику приходит оповещение о заказе. Поставщик также имеет возможность принять или отклонить заказ. В первом случае ему необходимо указать фактическое время доставки.

После этого заказ попадает во вкладку «Отправленные» и обзаводится новым параметром – статусом. Статус имеет право менять тот поставщик, который дал согласие на доставку, а администратор и учредитель оповещаются о состоянии статуса.

Когда заказ доставляется по адресу назначения, администратор отмечает, что заказ может быть закрыт. Только после этого поставщик имеет право присвоить статус заказу «Отправлено».

После этого администратору нужно отправить отчет учредителю и получить уведомление о том, что учредитель доволен доставкой заказа. На рисунке 9 представлена схема общего процесса работы над заказами.

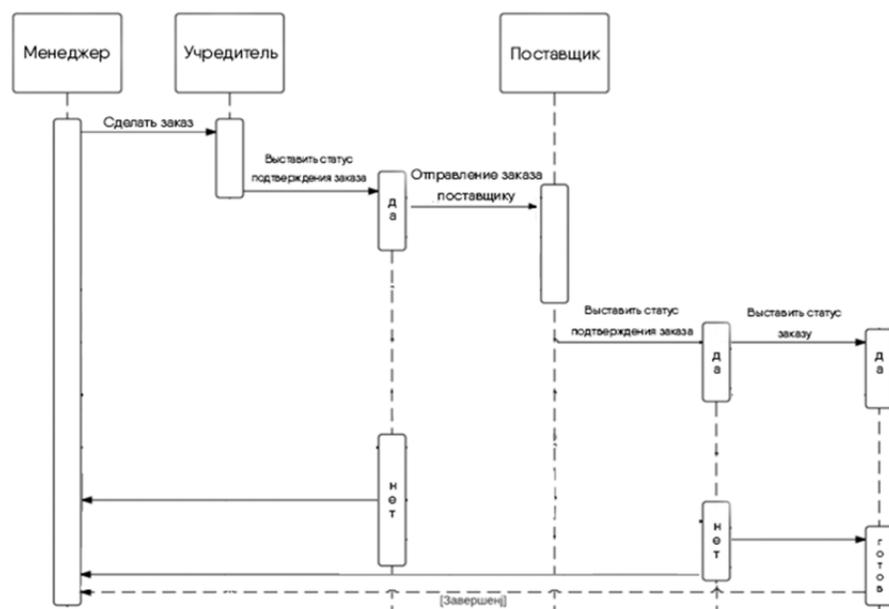


Рис. 9. Схема общего процесса работы над заказами

Выводы

Таким образом, для разработки веб-приложения для взаимодействия заинтересованных сторон в жизненном цикле жилого дома были выполнены следующие задачи:

- выделены ключевые стейкхолдеры, проведен их анализ, даны оценки влияния и важности их участия в строительстве жилых домов;
- выделены основные этапы жизненного цикла жилого дома;
- построена структура базы данных;
- разработан функционал для связи базы данных с модулями веб-приложения;
- разработан веб-интерфейс;
- настроена связь между интерфейсом и модулями;
- приложение внедрено и протестировано на базе строительной компании «ПМК-9».

На данный момент веб-приложение имеет неокончательный вид. Работа над ним продолжается с целью улучшения и наращивания функционала. Сейчас оно имеет схожую систему с PDM-системой из-за хранения всей необходимой информации о процессе строительства.

Список литературы

1. Дженнифур Р.Н. HTML5, CSS3 и JavaScript. Исчерпывающее руководство. М., 2014.
2. Докси К. Введение в программирование на Go. Creative Commons, 2014.
3. Макфарланд Д. Большая книга CSS3. СПб., 2014.



4. Марков А. В., Магель А. П. Разработка веб-приложения для взаимодействия заинтересованных сторон в жизненном цикле жилого дома // Студенческий. 2019. №11(55).
5. Программирование на Go с нуля: 9 полезных видеоуроков // Proglib : [сайт]. URL: <https://proglib.io/p/go-programming> (дата обращения: 10.03.2019).
6. Работаем с MySQL в Go // Medium : [сайт]. URL: <https://blog.maddevs.io/golang-mysql-9aa09cddb666> (дата обращения: 10.03.2019).
7. Руководство по языку Go // Metanit : [сайт]. URL: <https://metanit.com/go/tutorial> (дата обращения: 10.03.2019).
8. Справочник CSS // HTML Book : [сайт]. URL: <http://htmlbook.ru/css> (дата обращения: 19.02.2019).
9. Справочник по HTML // HTML Book : [сайт]. URL: <http://htmlbook.ru/html> (дата обращения: 18.02.2019).
10. Фельке-Моррис Т. Большая книга веб-дизайна. М., 2012.
11. Формат JSON, метод toJSON // JavaScript.ru : [сайт]. URL: <https://learn.javascript.ru/json> (дата обращения: 04.03.2019).
12. Хоган Б. HTML5 и CSS3. Веб-разработка по стандартам нового поколения. СПб., 2012.
13. Kurniawan A. Go Programming by Example. PE Press, 2015.
14. Ajax-запрос // jQuery: [сайт]. URL: <http://jquery.page2page.ru/index.php5/Ajax-запрос> (дата обращения: 02.03.2019).
15. jQuery. Write less, do more. URL: <https://jquery.com> (дата обращения: 25.02.2019).
16. Kozyra N. Mastering Go Web Services. Packt Publishing, 2015.
17. Youngman N., Peppé R. Get Programming with Go. Manning, 2018.
18. Sau Sheong Chang. Go Web Programming. Manning, 2015.
19. Kennedy W., Ketelsen B., St. Martin E. Go in Action. Manning, 2015.

Об авторах

Андрей Витальевич Марков — асп., Балтийский федеральный университет им. И. Канта, Россия.

E-mail: fantazydragon@mail.ru

Артур Павлович Магель — специалист, Балтийский федеральный университет им. И. Канта, Россия.

E-mail: magel_arthur@mail.ru

The authors

Andrey V. Markov, PhD Student, Immanuel Kant Baltic Federal University, Russia.

E-mail: fantazydragon@mail.ru

Arthur P. Magel, Expert, Immanuel Kant Baltic Federal University, Russia.

E-mail: magel_arthur@mail.ru