

СОДЕРЖАНИЕ

Введение

- 1 Анализ и моделирование предметной области программного средства
 - 1.1 Описание предметной области
 - 1.2 Разработка функциональной модели предметной области
 - 1.3 Анализ требований к разрабатываемому программному средству
 - 1.4 Разработка информационной модели предметной области
 - 1.5 UML-модели представления программного средства и их описание
 - 2 Проектирование и конструирование веб-приложения
 - 2.1 Постановка задачи
 - 2.2 Обоснование выбора компонентов и технологий для реализации программного средства
 - 2.3 Архитектурные решения
 - 2.4 Описание алгоритмов
 - 2.5 Проектирование пользовательского интерфейса
 - 2.6 Методы и средства для обеспечения безопасности данных
 - 3 Тестирование и проверка работоспособности программного средства
 - 3.1 Выбор и обоснование видов тестирования
 - 3.2 Результаты тестирования
 - 3.3 Вывод тестирования
 - 4 Руководство по использованию программного средства
- Заключение
- Список использованных источников
- Листинг программного кода
- Схема модели данных
- Схема алгоритма основного процесса предметной области
- Скриншоты рабочих окон программного средства

Введение

В современном мире высокая популярность использования людьми сети Интернет, большая часть коммуникации и взаимодействия людей между собой происходит именно там. Интернет позволяет коммуницировать с людьми по всему миру таким образом, что упрощает процесс функционирования онлайн-школ. Для подобных целей создаются различные веб-сервисы.

Целью курсового проекта является разработка веб-приложения для автоматизации работы онлайн школы иностранных языков на основе объектно-ориентированного языка программирования Java и фреймворка Spring Boot. Существует большое количество веб-сервисов для онлайн школ в сети Интернет, большинство данных сервисов являются иностранными, но имеют интуитивный пользовательский интерфейс. В некоторых случаях у данных веб-сервисов отсутствуют минимально необходимые функции.

Разрабатываемое программное средство должно обеспечивать выполнение следующих функций: авторизация и регистрация в системе, реализация механизма добавления новых курсов, формирование и редактирование кабинетов пользователей, а также обеспечение требований безопасности к сохранности пользовательских данных.

Для достижения поставленных целей курсового проектирования необходимо произвести анализ требований к разрабатываемому программному средству, произвести анализ существующих веб-сервисов, их функциональные возможности и недостатки в работе, выбрать языки программирования и средства разработки, выполнить постановку задач по разработке веб-сервиса.

Необходимо произвести проектирование архитектуры и выполнить формализацию предметной области, выполнить проектирование и реализацию способа хранения данных, проектирование и разработку пользовательского интерфейса, тестирование веб-сервиса.

В первом разделе «Анализ предметной области» рассматривается сущность предметной области, требования к разработке программного средства.

Во втором разделе «Проектирование и конструирование веб-приложения» описывается структура и дизайн программного средства, организация базы данных. Рассматриваются программно-технические средства необходимые для разработки веб-приложения.

В третьем разделе «Тестирование и проверка работоспособности программного средства» отражаются методы и процессы тестирования.

В разделе «Список использованных источников» приводится список использованной литературы.

Приложения содержат описание структуры программного средства, текст программных модулей и результаты работы приложения.

1 Анализ и моделирование предметной области программного средства

1.1 Описание предметной области

Предметная область для разрабатываемого программного средства относится к области разработки приложений, которые предназначены для интернет браузеров и не требуют установки на устройствах пользователя, позволяет пользователям взаимодействовать с данными, получать или вводить их и обрабатывать в динамическом режиме. Программы тесно связаны с сервером, поэтому работа их не статична и не требует отдельной загрузки страниц после каждого взаимодействия с сайтом. Как правило веб-приложение можно определить по обработке всех данных на лету. С помощью веб-приложений можно общаться, создавать контент на основе предпочтений, информация на страницах может обновляться динамично. Многие веб-программы могут использоваться, как дополнительный инструмент, например, переводчик.

Веб-сайты архитектурно реализуются на основе HTML-разметки. Состоят из одной или несколько веб-страниц, но может содержать и неограниченное их количество. На web-сайтах как правило публикуется статичная информация, которая не подразумевает активного взаимодействия пользователя с сайтом, помимо каких-либо базовых действий вроде авторизации и публикации информации.

На данный момент в интернете невозможно найти сайт, который не обладал бы интерактивными элементами. По-прежнему, веб-сайты остаются ресурсами с полезной информацией, а веб-приложения предлагают дополнительный пользовательский опыт. Есть задачи, которые невозможно реализовать без построения веб-приложения, но есть и задачи которые легко реализуются при помощи веб-сайтов.

Клиент-серверное приложение, в котором клиент взаимодействует с веб-сервером при помощи браузера. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется преимущественно на сервере, обмен информацией происходит по сети. Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются межплатформенными службами. Существенное преимущество построения веб-приложений для поддержки стандартных функций браузера заключается в том, что функции должны выполняться независимо от операционной системы данного клиента.

Другой подход заключается в использовании Adobe Flash, Silverlight или Java-апплетов для полной или частичной реализации пользовательского интерфейса. Поскольку большинство браузеров поддерживает эти технологии, Flash- или Java-приложения могут выполняться с легкостью. Так как они предоставляют программисту большой контроль над интерфейсом, они способны обходить многие несовместимости в конфигурациях браузеров.

Веб-приложение состоит из клиентской и серверной частей, тем самым реализуя технологию «клиент-сервер». Клиентская часть реализует пользовательский интерфейс, формирует запросы к серверу и обрабатывает ответы от него. Серверная часть получает запрос от клиента, выполняет вычисления, после этого формирует веб-страницу и отправляет её клиенту по сети с использованием протокола HTTP. Веб-приложение может выступать в качестве клиента других служб, например, базы данных или другого веб-приложения, расположенного на другом сервере.

Ярким примером веб-приложения является система управления содержимым статей Википедии: множество ее участников могут принимать участие в создании сетевой энциклопедии, используя для этого браузеры своих операционных систем и не загружая дополнительных исполняемых модулей для работы с базой данных статей.

В настоящее время набирает популярность новый подход к разработке веб-приложений, называемый Ajax. При использовании Ajax страницы веб-приложения не перезагружаются целиком, а лишь загружаются необходимые данные с сервера, что делает их более интерактивными и производительными.

Также в последнее время набирает большую популярность технология WebSocket, которая не требует постоянных запросов от клиента к серверу, а создает двунаправленное соединение, при котором сервер может отправлять данные клиенту без запроса от последнего.

Актуальность в рамках технической направленности проявляется в решении проблемы изучения языков в ходе реализации личностно-деятельностного и конвергентного подходов, масштабировании, возможности добавить новый текст в программу без разработки нового программного кода, самостоятельной разработки кода и прототипа программы без заданного шаблона.

Электронное приложение, оптимизирующее условия развития умения учиться в дополнительном образовании, позволяет наметить перспективные пути новых исследований и проектных решений в данном направлении и обладает научным потенциалом.

1.2 Разработка функциональной модели предметной области

Функциональная модель представляет собой модель инфологического уровня представления, в которой акцентируется функциональный аспект моделирования предметной области, в виде иерархии функций способствует пониманию поведения субъекта моделирования.

По методологии структурного анализа в первую очередь строится контекстная диаграмма - самое общее описание главной функции системы в целом и ее взаимодействия с внешней средой. Последующая функциональная декомпозиция сопровождается построением диаграмм декомпозиции, которые описывают каждый фрагмент декомпозиции и их взаимодействие.

Детализация функциональной модели продолжается до достижения необходимой степени подробности. После каждого шага декомпозиции выполняется сеанс экспертизы с участием экспертов предметной области.

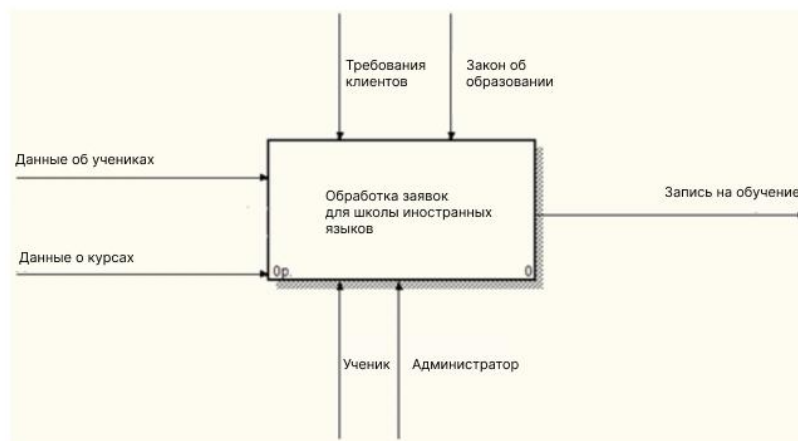


Рисунок 1.1 - Функциональная модель предметной области

На рисунке 1.1 представлена функциональная модель для программного средства для автоматизации работы школы иностранных языков.

1.3 Анализ требований к разрабатываемому программному средству

К разрабатываемому веб-сервису для реализации функционирования распорядка школы предъявляются следующие требования функциональности: авторизация и регистрация пользователей, возможность просмотра и изменения данных пользователя в личном кабинете; учет данных пользователей; просмотр, добавление и удаление данных о курсах; ведение графика занятий; поиск данных по параметрам, наименованию либо категории; просмотр и редактирование архива данных.

Функциональные требования описывают назначение и возможности разрабатываемого программного средства. Перед проектированием разработки веб-сервиса для автоматизация работы онлайн школы иностранных языков необходимо уделить достаточное внимание нефункциональным требованиям, которые влияют на пользовательский опыт использования разрабатываемого веб-сервиса, его корректную работу и отказоустойчивость.

К нефункциональным требованиям, предъявляемым к веб-сервису, относятся: интерфейс пользователя должен быть интуитивно понятным и соответствовать принципам инженерного дизайна; веб-сервис должен корректно отображаться в браузерах мобильных устройств и персональных компьютеров; устойчивость веб-сервиса к возникновению ошибок, их обработка и уведомление пользователей об возникновении.

1.4 Разработка информационной модели предметной области

Информационное моделирование — процесс описания или построения модели предметной области в том виде или формате, который, с одной стороны, легко воспринимается человеком, и, с другой стороны, легко может быть преобразован в набор элементов информационного хранилища, программных компонентов и других составляющих прикладного программного обеспечения. Чаще всего термин информационное моделирование можно видеть в контексте описания процесса построения ER диаграмм или UML диаграмм.

Программное обеспечение – часть организационно-экономической системы и процессов в ней, которая представляет интерес с точки зрения назначения ЭИС, отображается и используется в системе. Отражение в виде информации, представленной тем или иным образом называется информационной моделью.

Понятие включает в себя другие, более конкретные и частные виды моделей. Можно выделить следующие основные уровни информационного моделирования, проходя по которым осуществляется последовательное преобразование форм представления экономической информации: уровень концептуального моделирования; уровень логического и математического моделирования; уровень алгоритмического и программного моделирования.

1.5 UML-модели представления программного средства и их описание

Концептуальная модель (conceptual model) - на диаграммы такой модели будут смотреть, их будут обдумывать, но с самой моделью ничего делать не будут. Это не означает, что модель не нужна – это означает, что модель используется только для управления мыслительным процессом, для понимания. Поэтому мы называем такие модели концептуальными (также применяются термины модель анализа или аналитическая модель). Такой тип использования моделей один из самых важных, например, потому что так используются модели, которые получаются в результате анализа предметной области.

Модель проектирования (design model) - это модель проектирования представляет собой высокоуровневое (на уровне подсистем) и низкоуровневое (на уровне классов, если речь идет об использовании объектно-ориентированного подхода) описание программной системы. Модель проектирования предназначена для того, чтобы, руководствуясь ею, разработать программный код приложения. Как правило, архитектура (высокоуровневое описание) и код разрабатываются итеративно, и разработчикам в процессе разработки необходимо модифицировать модель проектирования, чтобы она соответствовала принимаемым решениям.

Главное требование к модели проектирования: понятность (usability). Действительно, разработчики должны полностью понимать модель, чтобы вести разработку.

Модель реализации (implementation model). Модель реализации предназначена для автоматического преобразования, возможно многократного, в существенно другой вид, например, в исполняемый код. Такое предназначение требует указания необходимых деталей реализации в модели, поскольку "от себя" компьютер их добавить не сможет. Главное требование к моделям реализации: полнота (completeness).

2 Проектирование и конструирование веб-приложения

2.1 Постановка задачи

Многопользовательское программное средство, задачей которого является автоматизация работы школы иностранных языков, позволяющее вести каталог данных, управлять статусами и редактировать списки записей, позволит упростить недостатки, имеющиеся у аналогов на данный момент.

Основные задачи, для которых создается программное средство для работы школы иностранных языков:

- авторизация и регистрация пользователей в системе;
- предоставление возможности ведения графика и учета занятий;
- предоставление возможности записи на курсы;
- предоставление возможности ведения отчетности учеников.

Основными преимуществами системы перед системами-аналогами являются:

- подробная структурированность каталога курсов;
- возможность хранения личной информации пользователей и архива данных;
- взаимодействие нескольких категорий пользователей между собой.

Назначение веб-приложения для онлайн школы иностранных языков заключается в том, чтобы систематизировать работу администраторов и преподавателей по ведению архива данных и статусов задач, а также автоматизировать работу с каталогом для пользователей.

2.2 Обоснование выбора компонентов и технологий для реализации программного средства

Java - объектно-ориентированный язык программирования строго типизированный общего назначения, разработанный компанией Sun Microsystems (сейчас принадлежит компании Oracle). Приложения обычно транслируются в специальный байт-код, поэтому могут работать на любой компьютерной архитектуре, для которой существует реализация виртуальной Java-машины. В проекте будет использоваться редакция языка Java – Java SE.

Java SE - это стандартная редакция Java, которая используется для разработки простых Java приложений. Например, можно создавать консольные приложения, апплеты, приложения с графическим интерфейсом. До 6 версии Java эта редакция имела название J2SE.

Spring Framework - фреймворк с исходным кодом для Java-платформы. Spring обеспечивает решения многих задач, с которыми сталкиваются Java-разработчики и организации, которые хотят создать информационную систему, основанную на платформе Java. Из-за широкой функциональности трудно определить наиболее значимые структурные элементы, из которых он состоит. Spring не всецело связан с платформой Java Enterprise, несмотря на свою масштабную интеграцию с ней, что является важной причиной его популярности. Spring, вероятно, наиболее известен как источник расширений, нужных для эффективной разработки сложных бизнес-приложений вне тяжеловесных программных моделей, которые исторически были доминирующими в промышленности. Еще одно его достоинство в том, что он ввел ранее неиспользуемые функциональные возможности в сегодняшние господствующие методы разработки, даже вне платформы Java. Этот фреймворк предлагает последовательную модель и делает её применимой к большинству типов приложений, которые уже созданы на основе платформы Java. Считается, что Spring реализует модель разработки, основанную на лучших стандартах индустрии, и делает её доступной во многих областях Java.

Spring обеспечивает решения многих задач, с которыми сталкиваются Java-разработчики и организации, которые хотят создать информационную систему, основанную на платформе Java. Из-за широкой функциональности трудно определить наиболее значимые структурные элементы, из которых он состоит. Spring не всецело связан с платформой Java Enterprise, несмотря на свою масштабную интеграцию с ней, что является важной причиной его популярности. Spring, вероятно, наиболее известен как источник расширений, нужных для эффективной разработки сложных бизнес-приложений вне тяжеловесных программных моделей, которые исторически были доминирующими в промышленности. Еще одно его достоинство в том, что он ввел ранее неиспользуемые функциональные возможности в сегодняшние господствующие методы разработки, даже вне платформы Java.

MySQL - свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощенной Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией.

Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации.

Apache Maven - фреймворк для автоматизации сборки проектов на основе описания их структуры в файлах на языке POM (англ. Project Object Model), являющемся подмножеством XML. Проект Maven издаётся сообществом Apache Software Foundation, где формально является частью Jakarta Project. Название системы является словом из языка идиш, смысл которого можно примерно выразить как «собиратель знания». Maven обеспечивает декларативную, а не императивную сборку проекта. В файлах описания проекта содержится его спецификация, а не отдельные команды выполнения. Все задачи по обработке файлов, описанные в спецификации, Maven выполняет посредством их обработки последовательностью встроенных и внешних плагинов. Maven используется для построения и управления проектами, написанными на Java, C#, Ruby, Scala, и других языках.

Список использованных источников

1. Пьюривал, С. Основы разработки веб-приложений / С. Пьюривал. – Санкт-Петербург : Питер, 2015. – 272 с.
2. Вигерс, К. Разработка требований к программному обеспечению / К. Вигерс. – Москва : Русская редакция, 2004. – 575 с.
3. Хоффман, Э. Безопасность веб-приложений. Разведка, защита, нападение / Э. Хоффман. – Санкт-Петербург : Питер, 2021. – 336 с.
4. Caselook [Электронный ресурс]. – Режим доступа: <https://www.caselook.ru/>. – Дата доступа: 15.03.2025.
5. Case.one [Электронный ресурс]. – Режим доступа: <https://www.case.one>. – Дата доступа: 15.03.2025.
6. Pravo(tech) [Электронный ресурс]. – Режим доступа: <http://pravo.tech.product/>. – Дата доступа: 15.03.2025.
7. Эккель, Б. Изучаем Философия Java. – Москва : ДМК Пресс, 2019. – 624 с.
8. Уоллис, К. Spring в действии. – Санкт-Петербург : Питер, 2018. – 800 с.
9. Тахагхогхи, С. Руководство по MySQL / С. Тахагхогхи, Х. Вильямс. – Москва : Русская редакция, 2007. – 544 с.
10. IntelliJ idea [Электронный ресурс]. – Режим доступа: <https://www.jetbrains.com/>. – Дата доступа: 06.03.2025.
11. Java [Электронный ресурс]. – Режим доступа: <https://www.java.com/ru/>. – Дата доступа: 07.03.2025.
12. Spring Boot [Электронный ресурс]. – Режим доступа: <https://spring.io/projects/spring-boot/>. – Дата доступа: 07.03.2025.
13. Spring Framework [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Spring_Framework/. – Дата доступа: 07.03.2025.
14. MySQL [Электронный ресурс]. – Режим доступа: <https://www.mysql.com/>. – Дата доступа: 11.03.2025.
15. Bootstrap [Электронный ресурс]. – Режим доступа: <https://getbootstrap.com/>. – Дата доступа: 11.03.2025.
16. DataGrip [Электронный ресурс]. – Режим доступа: <https://www.jetbrains.com/ru-ru/datagrip/>. – Дата доступа: 11.03.2025.
17. Postman [Электронный ресурс]. – Режим доступа: <https://www.postman.com/>. – Дата доступа: 1.03.2025.
18. Шкляр, Л. Архитектура веб-приложений / Л. Шкляр, Р. Розен. – Москва : Эксмо, 2011. – 640 с.
19. Мартин, Р. Чистая архитектура. Искусство разработки ПО / Р. Мартин. – Санкт-Петербург : Питер, 2021. – 550 с.
20. Shklar, L. Web Application Architecture. Principles, protocols and practices / L. Shklar, R. Rosen. – New Jersey : Wiley, 2009. – 448 с.
21. Арлоу, Д. UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование / Д. Арлоу, И. Нейштадт. – Санкт-Петербург : Символ-Плюс, 2007. – 624 с.

22. Кумсков, М. И. Системный Анализ. Предметная область. Модели на UML / М. И. Кумсков. – Москва : Издательские решения, 2020. – 100 с.
23. Нотация IDEF0 [Электронный ресурс]. – Режим доступа: <https://bpmn.pro/process/idef0/>. – Дата доступа: 18.03.2025.
24. Рудикова, Л. В. Базы данных. Разработка приложений / Л. В. Рудикова. – Санкт-Петербург : БХВ-Петербург, 2006. – 744 с.
25. Скотт, Б. Проектирование веб-интерфейсов / Б. Скотт, Т. Нейл. – Санкт-Петербург : Символ-Плюс, 2010. – 352 с.