

Министерство образования Республики Беларусь
Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Компьютерных технологий

Кафедра проектирования информационных компьютерных систем

Дисциплина "Объектно-ориентированное программирование"

К защите допустить:
Руководитель курсовой работы
старший преподаватель
кафедры
_____ А.В.Михалькевич
08.06.2025

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе
на тему

Автоматизация работы новостного портала

БГУИР КР 1-40 05 01-10 № 188 ПЗ

Студент (подпись студента)

Курсовая работа
представлена на проверку
08.06.2025

(подпись студента)

Реферат

БГУИР КР 1-40 05 01-10 № 188 ПЗ, гр. 784371

, Автоматизация работы новостного портала, Минск: БГУИР - 2025.

Пояснительная записка 32812 с., 8 рис., 0 табл.

Ключевые слова:

Предмет Объектно-ориентированное программирование, А.В.Михалькевич

Содержание

[Введение](#)

[1. Описание предметной области](#)

[2. Постановка задачи и обзор методов ее решения](#)

[3. Модели представления системы и их описание](#)

[4. Информационная модель системы и ее описание](#)

[5. Обоснование оригинальных решений по использованию технических и программных средств, не включенных в требования.](#)

[6. Описание алгоритмов реализации бизнес-логики серверной части проектируемой приложения](#)

[7. Руководство пользователя](#)

[8. Результаты тестирования разработанной системы и оценка выполнения задач](#)

[Заключение](#)

[Список использованных источников](#)

[Приложения](#)

Введение

ВВЕДЕНИЕ В данном курсовом проекте будет разработана система автоматизации работы новостного портала с поддержкой электронной доски объявлений (новостей). Для выполнения этой задачи необходимо будет выполнить следующие условия: 1. Веб-приложение должно быть выполнено на языке JAVA с применением технологий JSP и Servlets для построения GUI интерфейса (Java Grafical User Inerfase). 2. В рамках работы должны быть использованы следующие техники: разработка и использование компонентов JavaBeans, использование механизмов сокрытия данных, обработка исключительных ситуаций. 3. Бизнес-логика системы должна быть реализована только на серверной части приложения.

1 1. Описание предметной области

Электронная доска объявлений (новостей) функционально подобна обыкновенной: это сайт, где каждый желающий может вывесить своё объявление или новость, а все посетители сайта — прочитать его. Электронная доска объявлений (новостей), как правило, поделена на несколько тематических разделов, согласно содержанию объявлений (новостей).

Большинство электронных досок — бесплатные. Для размещения своего объявления пользователю нужно лишь ввести в специальной форме его тему, своё имя/псевдоним либо название организации, а также координаты: адрес электронной почты, почтовый адрес,

телефон, URL своего сайта и т. п. (набор данных зависит от конкретного ресурса). Как правило, отображаются только имена авторов и темы объявлений, а для просмотра полного текста объявления пользователь должен щёлкнуть по ссылке, ведущей к нему. В некоторых досках объявления могут подавать только зарегистрированные пользователи, в некоторых — все. Сейчас в интернете существует тысячи и даже десятки тысяч досок объявлений. Обычно каждая из них посвящается какому-либо отдельному виду объявлений. Существуют национальные доски объявлений, предназначенных для жителей конкретной местности.

Электронные доски объявлений бывают двух видов: модерлируемые (те, у которых есть так называемый модератор — человек, контролирующий работу этой доски) и немодерируемые — работающие автоматически.

Большинство досок объявлений в интернете предполагают размещение объявлений о продаже и покупке товаров и услуг. Но есть и такие сервисы, которые предлагают размещать объявления о передачи различных вещей в дар, а также о поиске новых хозяев для домашних животных, которые передаются в добрые руки безвозмездно. На страницах досок объявлений представлены тысячи бесплатных объявлений о недвижимости, авто, услугах, поиске работы, знакомствах, покупке и продаже оборудования, компьютеров, средств связи и прочему. Доски объявлений позволяет разместить объявление с фотографией и без дополнительной регистрации.

2 Постановка задачи и обзор методов ее решения

2.1 Постановка задачи

Разработаем упрощенную электронную доску объявлений. Исходя из материалов предметной области, центром системы будет пользователь, размещающий объявление. Для начала он должен внести в базу свои данные, а затем добавить новое объявление.

Каждый пользователь будет заноситься в базу данных, и по этим данным станет возможно отследить, какие объявления были им размещены.

Каждое объявление, как правило, относится к какой-либо категории, будь то автомобили, недвижимость, животные и др. При публикации объявления необходимо указать категорию, к которой оно относится, что позволит найти данное объявление при просмотре какой-либо конкретной категории.

2.2 Функциональная модель

В основе IDEF0 методологии лежит понятие блока, который отображает некоторую бизнес-функцию. Четыре стороны блока имеют разную роль: левая сторона имеет значение "входа", правая - "выхода", верхняя - "управления", нижняя - "механизма" (рисунок 1).



Рисунок 1 – Функциональная модель

Проанализировав предметную область и требования к задаче, выделим следующие объекты,

которые имеют отношение к системе:

1. Список пользователей
2. Список категорий
3. Список объявлений

Составим список всех функций, имеющих отношение к системе:

1. Добавление пользователя
2. Изменение данных пользователя
3. Просмотр профиля пользователя
4. Удаление пользователя
5. Добавление категории
6. Изменение категории
7. Удаление категории
8. Фильтрация по категориям
9. Публикация объявления
10. Редактирование объявления
11. Удаление объявления
12. Просмотр деталей объявления

Основной процесс (а также его декомпозиция) функциональной модели находится в приложении А.

3 Модели представления системы и их описание

Система состоит из двух связанных компонентов: клиента (браузер) и сервера. Вся логика приложения выполняется на сервере и осуществляется по вызову клиента, т.е. клиент создает запрос. Сервер выполняет его и возвращает результат клиенту. Диаграмма состояний клиентской части представлена в приложении В. Диаграммы вариантов использования описывают функциональное назначение системы или то, что система должна делать. Данная диаграмма представлена в приложении Б.

Центральное место в объектно-ориентированном программировании занимает разработка логической модели системы в виде диаграммы классов. Диаграмма классов (class diagram) служит для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования. Диаграмма классов может отражать, в частности, различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывать их внутреннюю структуру и типы отношений. Данная диаграмма представлена в приложении Д.

При рассмотрении диаграммы состояния, было отмечено, что хотя эта диаграмма и используются для спецификации динамики поведения систем, время в явном виде в них не присутствует. Временной же аспект поведения может иметь существенное значение при моделировании синхронных процессов, описывающих взаимодействия объектов. Для моделирования взаимодействия объектов во времени в языке UML используются диаграммы последовательности (приложение Г). Полный проект программной системы представляет собой совокупность моделей логического и физического уровней, которые должны быть

согласованы между собой. В языке UML для физического представления моделей систем используются диаграммы реализации (implementation diagrams), которые включают в себя диаграмму компонентов (приложение Е) и диаграмму развертывания (приложение Ж).

4 Информационная модель системы и ее описание

Все данные располагаются в базе данных, которая в свою очередь состоит из таблиц. В данном проекте использовались 3 таблицы, которые связаны между собой:

1. Таблица «user» (состоит из полей ФИО, электронной почты, телефона)
2. Таблица «notice» (состоит из полей заголовков, текст объявления, дата размещения)
3. Таблица «category» (состоит из полей номер категории, наименования категории)

Итоговая модель представлена в приложении 3.

Сущность находится в первой нормальной форме, если все атрибуты являются простыми или атомарными (их нельзя разделить на составные части без потери смысла) и среди атрибутов отсутствуют повторяющиеся группы. Также, не допускается хранить в одном атрибуте разные по смыслу значения.

Для приведения сущности к первой нормальной форме необходимо:

1. разделить сложные атрибуты на атомарные;
2. для групп повторяющихся атрибутов создать новые сущности;
3. установить с новыми сущностями связи типа «один ко многим»;
4. атрибуты, хранящие разносмысловую информацию, разделить на односмысловые.

Наша информационная модель соответствует вышеперечисленным требованиям, следовательно, находится в первой нормальной форме.

Сущность находится во второй нормальной форме, если она находится в первой нормальной форме и в ней отсутствуют неключевые атрибуты, функционально зависящие от части первичного ключа. Сущность, имеющая простой первичный ключ (т.е. состоящий из одного атрибута) и находящаяся в первой нормальной форме, автоматически находится и во второй нормальной форме.

Для приведения сущности ко второй нормальной форме необходимо:

1. выделить атрибуты, которые функционально зависят только от части первичного ключа;
2. поместить их в новую сущность;
3. установить с новой сущностью связь типа «один ко многим»;
4. повторить указанные выше действия, если это необходимо.

Наша информационная модель соответствует вышеперечисленным требованиям, следовательно, находится во второй нормальной форме.

Сущность находится в третьей нормальной форме, если она находится во второй нормальной форме и отсутствуют функциональные зависимости между ее неключевыми атрибутами.

Другая формулировка. Сущность находится в третьей нормальной форме, если она находится во второй нормальной форме и каждый ее неключевой атрибут нетранзитивно

зависит от первичного ключа.

Таким образом, можно сказать, что в третьей нормальной форме каждый неключевой атрибут сущности функционально зависит только от всего первичного ключа и ни от чего другого.

Для приведения сущности к третьей нормальной форме необходимо:

1. выделить атрибуты, которые функционально зависят от одного и того же неключевого атрибута;
2. поместить их в новую сущность;
3. установить с новой сущностью связь типа «один ко многим»;
4. повторить указанные выше операции, если это необходимо.

Наша информационная модель соответствует вышеперечисленным требованиям, следовательно, находится в третьей нормальной форме.

5 Обоснование оригинальных решений по использованию технических и программных средств, не включенных в требования.

Для написания курсового проекта использовались 2 фреймворка – Spring и Hibernate.

The Spring Framework — 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 с помощью рефлексии. Контейнер отвечает за управление жизненным циклом объекта: создание объектов, вызов методов инициализации и конфигурирование объектов путем связывания их между собой.

Объекты, создаваемые контейнером, называются управляемыми объектами или бинами (beans). Обычно конфигурирование контейнера осуществляется путем загрузки XML файлов, содержащих определение бинов и предоставляющих информацию необходимую для создания бинов.

Объекты могут быть получены либо с помощью Поиска зависимости, либо Внедрения зависимости. Поиск зависимости — шаблон проектирования, когда вызывающий объект запрашивает у объекта-контейнера экземпляр объекта с определенным именем или определенного типа. Внедрение зависимости — шаблон проектирования, когда контейнер передает экземпляры объектов по их имени другим объектам либо с помощью конструктора, либо свойства, либо фабричного метода.

Spring Framework обладает следующими отличительными особенностями:

1. Простота и мощь: в то время как простота использования является одной из основных особенностей Spring Framework, достигнута была она без малейшей потери предоставляемых возможностей. Хорошим примером является Spring JDBC: в то время как предоставлена возможность выполнять запросы к базе данных вообще без использования JDBC API,

существует возможность обратиться к любой функции этого интерфейса без потери преимущества в виде автоматической обработки ошибок. Существует возможность даже использовать без потерь расширения, присущие только одной конкретной базе данных — особенность, которую тяжело выполнить без Spring Framework.

2. Гибкость: вместо использования монолитного подхода, Spring содержит несколько модулей, которые независимы друг от друга. Благодаря этому возможно использование этого фреймворка только в тех частях реализуемой системы, где это принесет наибольшую пользу, и он может быть внедрен шаг за шагом, а не разом, как это бывает в случае серверов приложений.
3. Выбор: эта особенность подчёркивает, что нет ничего в мире универсального, что может быть использовано во всех случаях жизни. Spring Framework не требует от разработчика использовать какой-то специфический продукт или технологию. Например, поддерживается работа с большинством существующих фреймворков работающих с базами данных (JDBC, JPA, Hibernate, JDO, Apache OJB и других). Аналогично в области веб возможно как использование продукта, предложенного самим Spring, так и других широко используемых продуктов: JSF, Struts, Tapestry, WebWork. Spring также предлагает различные подходы к конфигурированию приложений: от использования конфигурационных XML файлов до использования Java аннотаций.

Эти особенности привели в результате к продукту, который не ограничен какой-то одной областью или технологией и может быть легко адаптирован и изменен для различных целей, оставаясь при этом простым для использования.

Hibernate — библиотека для языка программирования Java, предназначенная для решения задач объектно-реляционного проецирования (object-relational mapping — ORM). Она представляет собой свободное программное обеспечение с открытым исходным кодом, распространяемое на условиях GNU Lesser General Public License. Данная библиотека предоставляет лёгкий в использовании каркас (фреймворк) для отображения объектно-ориентированной модели данных в виде традиционных реляционных баз данных.

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.

6 Описание алгоритмов реализации бизнес-логики

серверной части проектируемой приложения

Для публикации объявления необходимо предварительно внести контактную информацию пользователя. Для этого через форму регистрации указываются ФИО пользователя, его электронная почта, а также телефонный номер. Информация добавляется в базу данных.

Далее заполняется форма публикации объявления. Указывается тема объявления, подробности, категория, к которой следует его отнести. При выполнении данного условия объявление добавляется в базу данных и становится доступным для просмотра.

Для каждого объявления возможен просмотр пользователя, разместившего его, категории. При выборе категории все объявления, относящиеся к ней, автоматически отображаются на экране. Также при выборе конкретного пользователя на экран выводятся все объявления, которые были им опубликованы.

Диаграмма работы бизнес-логики серверной части представлена в приложении И.

7 Руководство пользователя

Для запуска проекта его необходимо развернуть на сервере Tomcat. Для этого поставляемый на диске файл `bulletinBoard.war` скопировать в папку `%TOMCAT_HOME%\webapps` и запустить сервер командным файлом `startup.bat`.

После этих манипуляций сервер развернет проект из архива и зарегистрирует его.

Для корректной программы также необходимо на сервере MySQL создать базу данных `bulletinBoard` посредством выполнения скрипта `bulletinBoardDB.sql`.

Для запуска WEB проекта необходимо в адресной строке браузера ввести `http://localhost:8080/bulletinBoard`, после чего в браузере откроется основной вид программы (рисунок 2).



Рисунок 2 – Окно после запуска программы

8 Результаты тестирования разработанной системы и оценка выполнения задач

Проведем тестирование разработанной системы. Произведем регистрацию пользователя. Для этого открываем список пользователей нажимаем кнопку «Новый Пользователь» (рисунок 3).



Рисунок 3 – Регистрация нового пользователя

Как видно на рисунке 4, пользователь добавлен в базу данных и отображён в списке пользователей.



Рисунок 4 – Обновлённый список пользователей

Просмотрим профиль созданного пользователя (рисунок 5).



Рисунок 5 – Профиль созданного пользователя

Как видно на рисунке, данный пользователь не создал ещё ни одного объявления. Опубликуем объявление, нажав кнопку «Новое Объявление».



Рисунок 6 – Ввод данных нового объявления

Нажимаем кнопку «Сохранить». Объявление добавлено в базу данных, о чём свидетельствует обновившийся профиль пользователя. На рисунке 7 показан результат.



Рисунок 7 – Результат добавления объявления

Просмотрим список всех объявлений. Как видим, новое объявление благополучно отобразилось и в общем списке (рисунок 8)



Рисунок 8 – Обновлённый список объявлений

Заключение

В ходе выполнения курсового проекта было разработано web-приложение автоматизации работы новостного портала с модулем «Электронная доска объявлений». Так как данный проект не задумывался как коммерческое приложение, богатое функционалом, способное привлечь потенциальных клиентов, оно не способно серьёзно конкурировать с коммерческими электронными досками объявлений. Однако, при некоторых доработках, улучшении дизайна, данная доска объявлений может достойно стать в ряд подобных бесплатных приложений.

Список использованных источников

1. [печатное издание] 1. **Большой экономический словарь.** — М., 2002. — С. 298-299.
2. [печатное издание] 2. **UML. Классика CS. 2-у изд./Пер. с англ.; Под общей редакцией проф. С.Орлова** - СПб.: Питер, 2006. - 736 с.: ил.
3. [печатное издание] 3. 2. **Фаулер М. Архитектура корпоративных программных приложений.: Пер. с англ.** - М.: Издательский дом "Вильямс", 2006. - 544с.: ил. - Парал. тит. англ.
4. [печатное издание] 4. 3. **Басс Л., Клементс П., Кацман Р. Архитектура программного обеспечения на практике. 2-е издание.** - СПб.: Питер, 2006. - 575 с.: ил.
5. [печатное издание] 5. **Кей С. Хорстманн, Гари Корнелл. Java 2. Библиотека профессионала. Том 2. Тонкости программирования.** - Вильямс, 2006 г.
6. [печатное издание] 6. **Брюс У. Пери. Java сервлеты и JSP. Сборник рецептов** - КУДИЦ-Образ, 2006 г.
7. [печатное издание] 7. **Р. Мюллер. Базы данных и UML: Проектирование.**- Лори, 2002г.

Приложения

1. [электронный документ] [5ed4b3730cbf0_Blank_zadania.doc](#)
2. [электронный документ] [5ed4b3fd8596a_Приложение.docx](#)