**Содержание**

Введение…………………………………………………………………………….3

Глава 1. Основные сведения о социальных сетях .……………………………....4

1.1. Понятие социальной сети ……………………………………………...4

1.2. История развития социальных сетей …………………………….…….4

1.3. Причины популярности социальных сетей …………………………....6

Глава 2. Реализация проекта ……………………………………………………..…7

2.1. Основная идея и возможности системы ……………………………..…7

2.2. Дизайн проекта ……………………………………………………….….7

2.3. Используемые технологии ……………………………………………....8

2.3.1. Перечень используемых технологий ……………………….….8

2.3.2. Java ………………………………………………………….……9

2.3.3. Struts 2 ……………………………………………..……………10

2.3.4. JSP ………………………………………………………………11

2.3.5. Tomcat …………………………………………………………..12

2.3.6. MySql …………………………………………………………...12

2.4. Шаблоны проектрирования …………..………………………………..12

2.4.1. MVC ………………………………………………………………..13

2.4.2. DAO ………………………………………………………………..13

Глава 3. Перспективы развития …………………………………………………...14

Заключение …………………………………………………………………………15

Список литературы ………………………………………………………………...16

Приложение 1. Схема базы данных ………….…………………………………...17

Приложение 2. Краткое описание вариантов использования …………………..18

**Введение**

В современном мире информационные технологии играют важную роль. Многие люди проводят большое количество времени в интернете, при этом каждый находит для себя занятие по вкусу: кто-то просматривает информационные и новостные сайты, кто-то смотрит фильмы и различные видео онлайн, слушает музыку, кто-то совершает покупки в интернет-магазинах и, конечно же, практически каждый использует интернет пространство для общения с другими людьми. Именно для общения и существуют социальные сети, которые в наше время приобрели огромную популярность среди пользователей сети. И эта популярность не случайна. Социальные сети позваляют приобретать новых знакомых, находить людей со схожими интересами и вкусами, восстанавливать контакты с давними знакомыми, и также налаживать общение с людьми со всех уголков планеты. С уверенностью можно сказать, что социальные сети объединяют людей всего мира.

В связи с несомненной значимостью социальных сетей в жизни человека, мы решили посвятить свой проект разработке тематической социальной сети с оригинальной идеей, каждый участник которой будет иметь возможность поделиться частичкой доброты и сделать кого-то в этом мире счастливее.

**Цели и задачи работы**

Нашей целью было изучение принципов построения веб-приложения и применения их на практике для создания тематической социальной сети.

Задача нашей работы заключалась в реализации социальной сети на JavaEE с использованием фреймворка Struts 2.

**Глава 1.**

**Основные сведения о социальных сетях**

**1.1. Понятие социальной сети**

Социальная сеть – это платформа для построения социальных связей и отношений между людьми, которые разделяют интересы, виды деятельности, опыт или связи в реальной жизни. Сервис социальной сети состоит в представлении каждого пользователя (часто в профиле), его или её социальных связей и множества разнообразных дополнительных сервисов. Сайты социальных сетей – это веб-сервисы, которые предоставляют людям возможность создавать публичный профиль, составлять список пользователей, с которыми он пожелает контактировать, просматривать профили и связи других пользователей и делиться своими. Социальные сети – это основной инструмент для общения наравне с электронной почтой. Сайты социальных сетей очень разнообразны и объединяют новые информационные инструменты и средства общения такие, как мобильная связь, обмен фото и видео, блоггинг. Онлайн сервисы для общения иногда рассматриваются как сервисы социальных сетей, хотя в более широком смысле сервисы социальных сетей обычно подразумевают индивидуальные сервисы, тогда как онлайн сервисы для общения являются групповыми сервисами. Сайты социальных сетей позволяют пользователям делиться идеями, изображениями, постами, занятиями, событиями, интересами с людьми в своей сети.

**1.2. История развития социальных сетей**

Термин «социальная сеть» был введён в 1954 году социологом из «Манчестерской школы» Джеймсом Барнсом в работе «Классы и собрания в норвежском островном приходе», которая вошла в сборник «Человеческие отношения. Однако идею социальной сети как структуры, элементами которой являются действующие лица (индивидуумы или организации), ученые предвосхитили еще в конце восьмидесятых – французский социолог Давид Эмиль Дюркгейм и его немецкий коллега Фердинанд Тённис практически описали природу этого явления в своих теориях и исследованиях социальных групп.

Понимание социальной сети с того времени, безусловно, изменилось, немалую роль в этом сыграло развитие технологий, благодаря которым появились онлайн сети.

Первой социальной сетью в современном понимании этого термина принято считать сайт Classmates.com, cозданный в 1995 г Рэнди Конрадом, инженером компании «Боинг», который очень хотел найти своего одноклассника. Как и современный российский аналог(“Одноклассники”), сайт помогал зарегистрированным пользователям находить и поддерживать связь с друзьями и знакомыми, с теми, с кем человек имел дело в течении всей своей жизни - в детском саду, школе, вузе, на работе или находясь на военной службе. Он существует до сих пор. Однако американский сервис для одноклассников достаточно долго не поддерживал создание личных страниц (профилей) и добавление друзей, ограничивался возможностью соединения человека с его учебным заведением и поиском по списку людей, учившихся там же, поэтому существует другое мнение на тему, какую социальную сеть считать первой, и оно имеет право на существование.

Исследователи выделяют среди других сайт знакомств SixDegrees.com, созданный в 1997 г. Правда, в отличие от «Одноклассников», данный ресурс просуществовал только до 2000 года. Особенным этот сайт сделал ряд инструментов, позволяющий пользователям создавать личные страницы, добавлять друзей, а с 1998 года и совершать поиск по страницам друзей. Хотя на тот момент многие известные сайты знакомств и сайты, объединявшие представителей одного сообщества, предлагали подобные сервисы по отдельности: личные страницы можно было создавать практически на всех сервисах, а мессенджеры ICQ и AIM поддерживали списки друзей – только соцсеть SixDegrees.com соединила все эти возможности.

Иногда также первой соцсетью называют международную любительскую компьютерную сеть «Фидонет» (FidoNet). Она была создана американским программистом Томом Дженнингсом в 1984 году и служила для передачи сообщений с электронных досок объявлений друг на друга. Изначально он сделал ее для передачи сообщений со своей электронной доски на электронную доску своего друга, и структура сети была линейной: каждый узел связывался с другим напрямую. С ростом числа узлов структуру пришлось изменить на древовидную. Однако ее функционал оставался достаточно специфичен и ограничен: например, обмениваться почтой с другим узлом сети можно было только в специально отведенный для этого час ночью, когда стоимость телефонных звонков была ниже.

С того момента до сегодняшнего дня было создано множество социальных сетей, некоторые из них ознаменовали этапы изменения социальной сети как явления.

По мере увеличения числа соцсетей в веб-пространстве постепенно стали появляться тематические социальные сети, например, LinkedIn стала одной из первых и самой популярной в настоящий момент сетью деловых контактов. В марте 2013 года сайт стал 22-м в списке самых посещаемых сайтов в мире. Нельзя не отметить социальную сеть MySpace, которая ознаменовала переход к эпохе предоставления пользователям больших, чем ранее, возможностей для творчества. Долгое время MySpace оставалась самой популярной социальной сетью в США и во всём мире, в июле 2006 года сайт становился самым посещаемым ресурсом в США, обогнав почтовую службу Yahoo!, но последние несколько лет подряд посещаемость ресурса неуклонно снижается. Facebook, запущенный в 2004 году, стал самой крупной социальной сетью в мире в начале 2009. Facebook был впервые представлен как социальная сеть Гарварда, расширившаяся до других университетов и в итоге до всего мира.

**1.3. Причины популярности социальных сетей**

Существует множество причин, по которым владельцев аккаунтов на разных сайтах социальных сетей становится всё больше. Среди них можно выделить следующие:

* Безграничность общения. Общаться можно когда угодно и с кем угодно. Нет ни пространственных, ни временных ограничений. Это позволяет не только поддерживать связь с друзьями и родственниками, находящимися далеко, но и находить новые знакомства.
* Возможность приобщаться к группам по интересам. Различные объединения в социальных сетях позволяют искать единомышленников, интересную и нужную информацию, обмениваться разными навыками и опытом. Это очень удобно, когда в реальной жизни в той или иной области найти единомышленников сложно.
* Для пользования социальными сетями не нужны какие-то особые навыки. Если же вопросы все-таки возникают, то всегда поможет служба поддержки или специализированные сайты. Но обычно интерфейс порталов максимально понятен, поэтому разбираться в нем долго не приходится.
* Возможность получить оценку себя и своих занятий другими пользователями. Большинство людей в своей жизни ориентируются на оценку окружающих, а в интернете ее получить достаточно легко: система «лайков» существует во многих социальных сетях.

**Глава 2.**

**Реализация проекта**

**2.1. Основная идея и возможности системы**

Идея проекта заключается в обмене желаниями между пользователями. Смысл этой идеи состоит в том, чтобы предоставить возможность людям не только делиться своими мечтами, но и помогать другим в их осуществлении.

Стать участником нашего сайта сможет любой пользователь, пройдя простой процесс регистрации. Зарегистрированные пользователи будут иметь возможность заполнять свой профиль и составлять список своих желаний, который может включать не более 5 пунктов. Данный список, также как и профиль, смогут просматривать другие пользователи. Заинтересовавшиеся пользователи будут иметь возможность подавать заявки на выполнение любого желания и при приёме заявки данное желание осуществлять, также при необходимости заявку можно и отменять. Хозяин желания сможет выбрать любого из подавших заявку в качестве исполнителя. При возникновении проблем при выполнении желания, как хозяин, так и исполнитель, сможет отменить выполнение. Когда желание будет выполнено, его хозяин должен будет подтвердить выполнение. Каждый пользователь сможет увидеть те желания, в которых принимал участие он или любой другой пользователь.

Как и в любой социальной сети, в нашей сети существует возможность обмена сообщениями и поиска пользователей. Кроме того существует поиск по желаниям и поиск именинников.

Существует русская и английская версии сайта.

Подробнее со всеми возможностями системы можно ознакомиться в Приложении 2. «Краткое описание вариантов использования».

**2.2. Дизайн проекта**

Наша социальная сеть имеет тематическое название «Shooting stars», напрямую связанное с тематикой проекта и объясняющее его предназначение. В переводе с английского название означает «Падающие звёзды». Считается, что если загадать желание, увидев падающую звезду, то оно непременно исполнится. Так и на нашем сайте каждый участник сможет стать своеобразной падающей звездой для другого, исполнив его желание.

Логотип нашей сети был разработан специально для неё и соответствует названию.

Девиз сети звучит следующим образом: «Мечты сбываются… Загадай желание!». В английской версии – «Dreams come true… Make a wish!». Девиз олицетворяет идею сети.

Цветовое решение сайта исполнено в оранжевом, кирпичном, различных светлых тонах, с использованием классических белого и чёрного цветов.

Оранжевый – основной цвет – цвет пламени свечи на торте, очень тёплый и позитивный цвет. Оранжевый цвет означает активность и хорошо развитое творческое начало. По своей сути, этот цвет – отличный антидепрессант, именно его оттенки способны восстанавливать после сильного нервного потрясения. Многие, наверное, замечали, что люди, предпочитающие этот цвет, дружелюбны и всегда полны энергией, активны, открыты ко всему новому. Оранжевый цвет в основном символизирует тепло солнца, а так же накал внутренней энергии. Оранжевый цвет в психологии – это почти синоним непрерывного движения. Свободолюбие, терпимость, многосторонняя развитость, талант – вот все то, чем может одарить этот необыкновенный солнечный цвет каждого при мудром его использовании. В оформлении используются различные оттенки этого цвета.

В общем и целом, мы постарались оформить наш сайт таким образом, чтобы он был максимально удобен в использовании и располагал к общению, творчеству и взаимопомощи.

**2.3. Используемые технологии**

**2.3.1. Перечень используемых технологий**

Серверная часть нашего проекта была написана на языке Java (версии JDK 1.8.0 и JavaEE v7) с использованием фреймворка Struts 2. Использовался веб-сервер Tomcat 8.0.5 и в сервер баз данных MySql 5.6.21. Работа с базами данных осуществлялась на языке SQL. Для конфигурации сервера использовался язык XML. Для регистрации событий и ошибок использовался Log4j.

При разработке клиентской части применялись JavaScript и Ajax, CSS3 и HTML5, технологии JSP и JSTL, кастомные теги, написанные на Java.

В качестве среды разработки мы пользовались Intellij IDEA 14.1.1 (Java EE IDE).

**2.3.2. Java**

Java является объектно-ориентированным языком программирования.

ООП — методология программирования, основанная на представлении программного продукта в виде совокупности объектов, каждый из которых является экземпляром конкретного класса. ООП использует в качестве базовых элементов взаимодействие объектов.

Основные принципы ООП:

* Инкапсуляция (encapsulation) — принцип, объединяющий данные и код, манипулирующий этими данными, а также защищающий данные от прямого внешнего доступа и неправильного использования. Другими словами, доступ к данным класса возможен только посредством методов этого же класса.
* Наследование (inheritance) — процесс, посредством которого один класс может наследовать свойства другого класса и добавлять к ним свойства и методы, характерные только для него.
* Полиморфизм (polymorphism) — механизм, использующий одно и то же имя метода для решения похожих, но несколько отличающихся задач в различных объектах при наследовании из одного суперкласса. Целью полиморфизма является использование одного имени при выполнении общих для суперкласса и подклассов действий.

Системная библиотека классов языка Java содержит классы и пакеты, реализующие и расширяющие базовые возможности языка, а также сетевые средства, взаимодействие с базами данных, графические интерфейсы и многое другое.

Программы на Java транслируются в байт-код, выполняемый виртуальной машиной Java (JVM) — программой, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор. Достоинством подобного способа выполнения программ является полная независимость байт-кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве, для которого существует соответствующая виртуальная машина. Другой важной особенностью технологии Java является гибкая система безопасности, в рамках которой исполнение программы полностью контролируется виртуальной машиной. Любые операции, которые превышают установленные полномочия программы (например, попытка несанкционированного доступа к данным или соединения с другим компьютером), вызывают немедленное прерывание.

Java EE (Enterprise Edition) представляет собой широко используемую платформу, содержащую набор взаимосвязанных технологий, которые существенно сокращают стоимость и сложность разработки, развертывания многоуровневых серверных приложений, а также управления ими. Платформа Java EE основана на платформе Java SE и предоставляет набор интерфейсов API (интерфейсов разработки приложений) для разработки и запуска портируемых, надежных, масштабируемых и безопасных серверных приложений.

Актуальная версия Java EE имеет номер 7.0.

Java EE API, включает в себя несколько технологий, которые расширяют функциональность базовых Java SE API-интерфейсов. Среди них мы использовали:

javax.servlet. \*

Спецификация сервлетов определяет набор программных интерфейсов для обслуживания HTTP-запросов. Она включает в себя спецификации JavaServer Pages.

javax.el. \*

Этот пакет определяет классы и интерфейсы для языка выражений в Java EE. Expression Language (EL) — это простой язык, изначально предназначенный для удовлетворения конкретных потребностей разработчиков веб-приложений.

**2.3.3. Struts 2**

Apache Struts 2 — фреймворк с открытым исходным кодом для создания Java EE веб-приложений. Основывается на Java Servlet API и расширяет его, в архитектурном плане реализует шаблон проектирования MVC.

Model-view-controller (MVC, «Модель-представление-контроллер») — схема использования нескольких шаблонов проектирования, с помощью которых модель данных приложения, пользовательский интерфейс и взаимодействие с пользователем разделены на три отдельных компонента так, что модификация одного из компонентов оказывает минимальное воздействие на остальные.

Основная цель применения этой концепции состоит в разделении бизнес-логики (модели) от её визуализации (представления, вида). За счет такого разделения повышается возможность повторного использования.

В обычном Java EE веб-приложении клиент, как правило, отправляет информацию на сервер из формы. Введённые данные передаются сервлету, который обрабатывает её, при необходимости взаимодействуя с базой данных, и формирует HTML ответ.

 Struts был создан для того, чтобы чётко разделить модель (бизнес-логику), представление (HTML-страницы) и контроллер (отвечающий за передачу данных от модели к представлению и обратно).

Конфигурация задается в файле struts.xml

**2.3.4. JSP**

JSP (JavaServer Pages) — технология, позволяющая веб-разработчикам создавать содержимое, которое имеет как статические, так и динамические компоненты. Страница JSP содержит текст двух типов: статические исходные данные, которые могут быть оформлены в одном из текстовых форматов HTML,SVG, WML, или XML, и JSP- элементы, которые конструируют динамическое содержимое. Кроме этого могут использоваться библиотеки JSP-тегов, а также EL (Expression Language), для внедрения Java-кода в статичное содержимое JSP-страниц.

Смысл разделения динамического и статического содержания в том, что статические ресурсы могут находиться под управлением HTTP-сервера в то время, как динамические нуждаются в движке (JSP Engine) и в большинстве случаев в доступе к уровню данных. Рекомендуется разделить и разрабатывать параллельно две части приложения: часть, состоящяя только из динамических ресурсов, и часть, состоящяя только из статических ресурсов.

JSP является платформонезависимой, переносимой и легко расширяемой технологией для разработки веб-приложений.

Спецификация JSP содержит библиотеку стандартных тегов JSTL, по- зволяет разработчику создавать собственные теги, кроме того, теги обеспе- чивают возможность использования JavaBean и обращение к классам бизнес-логики. Содержимое Java Server Pages (теги HTML, теги JSP и скрипты) переводится в сервлет код-сервером. Этот процесс ответствен за трансляцию как динамических, так и статических элементов, объявленных внутри файла JSP. Об архитектуре сайтов, использующих JSP/Servlet-технологии, часто говорят, как о thin-client (использование ресурсов клиента незначительно), потому что большая часть логики выполняется на сервере.

**2.3.5. Tomcat**

Tomcat (в старых версиях — Catalina) — контейнер сервлетов с открытым исходным кодом, разрабатываемый Apache Software Foundation. Реализует спецификацию сервлетов и спецификацию JavaServer Pages (JSP) и JavaServer Faces (JSF). Написан на языке Java.

Tomcat позволяет запускать веб-приложения, содержит ряд программ для самоконфигурирования.

Tomcat используется в качестве самостоятельного веб-сервера, в качестве сервера контента в сочетании с веб-сервером Apache HTTP Server, а также в качестве контейнера сервлетов в серверах приложений JBoss и GlassFish.

Контейнер сервлетов  — программа, представляющая собой сервер, который занимается системной поддержкой сервлетов и обеспечивает их жизненный цикл в соответствии с правилами, определёнными в спецификациях. Может работать как полноценный самостоятельный веб-сервер, быть поставщиком страниц для другого веб-сервера, например Apache, или интегрироваться в Java EE сервер приложений. Обеспечивает обмен данными между сервлетом и клиентами, берёт на себя выполнение таких функций, как создание программной среды для функционирующего сервлета, идентификацию и авторизацию клиентов, организацию сессии для каждого из них.

**2.3.6. MySql**

MySQL  — свободная реляционная система управления базами данных. MySQL является решением для малых и средних приложений. Входит в состав серверов WAMP, AppServ, LAMP и в портативные сборки серверов Денвер, XAMPP, VertrigoServ. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы.

**2.4. Шаблоны проектрирования**

Шаблон проектирования или паттерн (design pattern) в разработке программного обеспечения — повторимая архитектурная конструкция, представляющая собой решение проблемы проектирования в рамках некоторого часто возникающего контекста.

Обычно шаблон не является законченным образцом, который может быть прямо преобразован в код; это лишь пример решения задачи, который можно использовать в различных ситуациях. Объектно-ориентированные шаблоны показывают отношения и взаимодействия между классами или объектами, без определения того, какие конечные классы или объекты приложения будут использоваться.

**2.4.1. MVC**

Model-view-controller (MVC, «модель-представление-контроллер», «модель-вид-контроллер») — схема использования нескольких шаблонов проектирования, с помощью которых модель приложения, пользовательский интерфейс и взаимодействие с пользователем разделены на три отдельных компонента таким образом, чтобы модификация одного из компонентов оказывала минимальное воздействие на остальные. Данная схема проектирования часто используется для построения архитектурного каркаса, когда переходят от теории к реализации в конкретной предметной области.

Концепция MVC позволяет разделить данные, представление и обработку действий пользователя на три отдельных компонента:

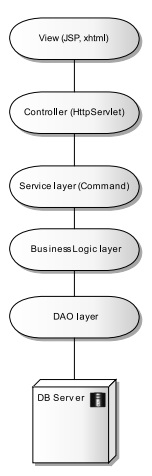
* Модель (Model). Модель предоставляет знания: данные и методы работы с этими данными, реагирует на запросы, изменяя своё состояние. Не содержит информации, как эти знания можно визуализировать.
* Представление, вид (View). Отвечает за отображение информации (визуализацию).
* Контроллер (Controller). Обеспечивает связь между пользователем и системой: контролирует ввод данных пользователем и использует модель и представление для реализации необходимой реакции.

**2.4.2. DAO**

В программном обеспечении data access object (DAO) — это объект, который предоставляет абстрактный интерфейс к какому-либо типу базы данных или механизму хранения. Определённые возможности предоставляются независимо от того, какой механизм хранения используется и без необходимости специальным образом соответствовать этому механизму хранения. Этот шаблон проектирования применим ко множеству языков программирования, большинству программного обеспечения, нуждающемуся в хранении информации и к большей части баз данных, но традиционно этот шаблон связывают с приложениями на платформе Java Enterprise Edition, взаимодействующими с реляционными базами данных через интерфейс JDBC.

При создании информационной системы выявляются некоторые слои, которые отвечают за взаимодействие различных частей приложения. Связь с базой данных является важной частью любой системы, поэтому всегда выделяется часть кода, ответственная за передачу запросов в БД и обработку полученных от нее ответов. Общее определение шаблона Data Access Object трактует его как прослойку между приложением и СУБД. DAO абстрагирует бизнес-сущности системы и отражает их на записи в БД. DAO определяет общие способы использования соединения с БД, моменты его открытия и закрытия или извлечения и возвращения в пул.

Вершина иерархии DAO представляет собой класс или интерфейс с описанием общих методов, которые будут использоваться при взаимодействии с таблицей или группой таблиц. Как правило, это методы выбора, поиска сущности по признаку, добавление, удаление и замена информации.



Архитектурное проектирование. Взаимодействие слоев.

**Глава 3.**

**Перспективы развития**

В будущем планируется расширение и совершенствование системы, исследование потенциального круга пользователей, запуск на реальном домене и seo-продвижение.

Среди дополнительных функций, которые планируется добавить:

* Рекомендации пользователей
* Поиск по возможностям
* Возможность восстановления пароля
* Подтверждение адреса электронной почты
* Рассылки
* Возможность удаления сообщений, чатов
* Запрос на подтверждение выполненного желания
* Добавление пользователя с правами администратора
* При регистрации и поиске пользователей – выбор стран из предложенного списка
* Поздравление именинников (изменение дизайна страницы – добавление дополнительных элементов)
* Ввод ключевых слов при описании своих возможностей.
* Адаптивный дизайн
* Поддержка различных браузеров

**Заключение**

Таким образом, в ходе работы нами были изучены принципы проектирования веб-приложения, исследован и реализован процесс разработки веб-приложения, изучены средства и технологии, которые используются при разработке веб-приложения. Кроме того, мы ознакомились с основными составляющими и возможностями социальных сетей.

На практике были применены все знания, полученные при теоритическом исследовании, использованы различные технологии, которые совершенствуют и упрощают процесс разработки веб-приложения, и в результате был создан реальный многофункциональный проект, представляющий из себя реализацию нашей идеи – тематическая социальная сеть по обмену желаниями.

Однако существуют ещё некоторые усовершенствования, которые мы хотим добавить в нашу систему. Поэтому мы не собираемся останавливаться на достигнутом, а будем продолжать развивать и расширять наш проект, пока в конечном итоге он не будет запущен в реальном интернет пространстве.

**Список литературы**

1. Блинов, И.Н., Романчик, В. С.

Java. Методы программирования : уч.-мет. Пособие — Минск : издательство «Четыре четверти», 2013. — 896 с. ISBN 978-985-7058-30-3

1. http://studopedia.org/3-143183.html

“Понятие и история возникновения социальных сетей.”

**Приложение 1. Схема базы данных**

