Университетский политехнический колледж

**Утверждено** Специальность

предметной комиссией 09.02.03

«Программное обеспечение»

«28» декабря 2017 г.

Председатель ПЦК

\_\_\_\_\_\_\_\_\_\_\_\_ Н. Р. Гафуров

# ЗАДАНИЕ

на курсовое проектирование по модулю ПМ.02 «Разработка и администрирование баз данных»

|  |  |  |  |
| --- | --- | --- | --- |
| учащемуся группы | 32928/3 |  | Горюшин Андрей Максимович |
| Тема курсового проекта: | «**Управления социальными взаимодействиями**» | | |
|  | | | | |

1. ТЕХНИКО-МАТЕМАТИЧЕСКАЯ ФОРМУЛИРОВКА ЗАДАЧИ

Разработать веб-приложению, позволяющее автоматизировать поздравление людей, создавать напоминания о будущем праздничном событии с возможностью прикрепления подарка к этому событию, возможность анонимного или нет одноразового заказа без регистрации

Организовать многопользовательский доступ к разработанному веб-приложению.

2. ОПИСАНИЕ ИСХОДНЫХ ДАННЫХ

Входная информация поступает в БД в виде данных о пользователе, данных для оформления заказа, данных о событиях и данных о подарках, так же об изменении и удалении дынных карточки событий и данных о подарках.

1. ТРЕБОВАНИЯ К ФУНКЦИОНАЛЬНЫМ ХАРАКТЕРИСТИКАМ

* реализовать многопользовательский доступ к веб-приложению;
* реализовать возможность анонимного или нет одноразового заказа без регистрации;
* реализовать возможность редактировать личные данные в профиле;
* реализовать возможность после регистрации добавления карточки событий и добавления подарка к этому событию, так же изменения и удаление карточки события и подарка, прикрепленного к этому событию;
* реализовать возможность поиска карточки события по ее заголовку;
* обеспечить удобный пользовательский интерфейс.

1. ТРЕБОВАНИЯ К СОСТАВУ И ФОРМЕ ВЫДАЧИ РЕЗУЛЬТАТОВ ПРОГРАММЫ

На выходе пользователи системы получают информацию в виде таблиц из БД (количество зарегистрированных пользователей, количество одноразовых заказов, количество событий данного пользователя)

5. ПЕРЕЧЕНЬ МАТЕРИАЛОВ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ

Введение

1. Теоретические основы разработки
   1. Описание предметной области
   2. Анализ методов решения
   3. Обзор средств программирования
   4. Обоснование выбора языка программирования
   5. Обзор систем управления базами данных
   6. Обоснование выбора системы управления базами данных
2. Практическая часть
   1. Постановка задачи
   2. Проектирование базы данных
   3. Проектирование приложения
   4. Описание приложения
   5. Руководство оператора
   6. Программа и методика испытаний
   7. Протокол испытаний

Заключение

Список литературы

Приложения

1. ПЕРЕЧЕНЬ МАТЕРИАЛОВ ГРАФИЧЕСКОЙ ЧАСТИ

В графической части представлены ER-диаграмма и схема базы данных. Графическая часть включена в пояснительную записку.

1. СПИСОК РЕКОМЕНДОВАННОЙ ЛИТЕРАТУРЫ

* ГОСТ 19.105-78 Общие требования к программным документам
* ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом
* ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению
* ГОСТ 19.301-78 Программа и методика испытаний. Требования к содержанию и оформлению
* ГОСТ 19.401-78 Текст программы. Требования к содержанию и оформлению
* ГОСТ 19.402-78 Описание программы. Требования к содержанию и оформлению
* ГОСТ 19.505-79 Руководство оператора. Требования к содержанию и оформлению
* ГОСТ 19.701-90 Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Дата выдачи задания | 15.01.2018 |  | Зав. отделением |  | Сухорукова О. А. |
| Срок окончания | 15.05.2018 |  | Преподаватель |  | Гафуров Н. Р. |
|  |  |  | Студент |  | Горюшин А.М |

**Теоретические основы разработки**

Была цель создать веб-приложение целью которого являлось **управление социальными взаимодействиями**. Задача заключалась в том, чтобы создать такой сервис, который помогал бы занятым людям создавать напоминания о праздниках и поздравлять своих родственников, друзей, коллег по рабате.

Наш сервис предоставляет возможные решения для разных типов людей, так как у нас реализовано две функциональные возможности. «Одноразовый заказ», он подойдет для тех, кто не хочет регистрироваться или не нужен весь функционал сервиса. Пользователь заполняет форму для оформления заказа, которая состоит из 4-х простых шагов. Также «одноразовый заказ» по желанию предоставляет функция анонимности, чтобы получатель не узнал, кем был сделан подарок.

Для тех, кто хочет, чтобы перед ними раскрылся полный функционал сервиса, то можно зарегистрироваться. После регистрации будет предоставлен личный кабинет, где будут записаны персональные данные и выведены в карточку пользователя. Данные можно редактировать. Основной функционал, который будет доступен после регистрации, это доска для создания событий и добавления подарков к определенному событию. После заполнения формы на создание карточки события, которая содержит: название, дату и время, повод, кому дарить, периодичность напоминания и за сколько стоит напомнить до начала события. К карточке можно добавить один или несколько подарков, выбрав категорию, предмет из этой категории и количество. Данные занесенные в карточку можно изменять, как и данные о подарке, если пользователь захочет удалить карточку или подарок, то у него появится предупредительное окно о том, что карточку или подарок нельзя будет восстановить, после удаления. Также в доске событий реализован список на тот случай, если у пользователя будет создано много карточке, и он захочет найти какую-то конкретную из этого множества.

1. Описание предметной области

В наше время у всех людей есть родственники, множество друзей и знакомых, коллег по работе. И всем им нужны подарки на различные праздники: День рождение, Новый год, 8 марта, 23 февраля, годовщина свадьбы, но в силу своих личных дел, семьи, работы люди не могут запомнить праздничные события всех, а тем более им не хватает времени ездить по магазинам и выбирать подарки. Из-за того, что люди забывают о праздниках родных, близких и друзей, они начинают ругаться, перестают общаться и разрывают связь друг с другом. Так же есть люди с ограниченными возможностями, которые в силу своего здоровья не могу ездить по магазинам и дарить их свои близким и родным, так как это будет доставлять им неудобства.

1. Анализ методов решения

Данную проблему можно решить несколькими способами. Первый – это сделать такой сервис, с помощью которого люди могли бы создавать напоминания, с некоторыми настройками: периодичность напоминаний, за сколько напомнить, у кого праздник и что это за праздник, но это решает только одну проблему. Второе – это создать такой сервис, где люди могли бы заранее выбирать подарки и не волноваться, что они будут думать о них в самый последний момент перед праздником, а когда подходит даты события им приходило на почту уведомление или был обратный звонок о подтверждении покупки.  
Третий способ, это объединить первое и второе решение, чтобы можно было создавать событие, с такими же настройками, что в первом пункте, но также можно было прикрепить подарок к этой же карточке. Если пользователь зайдет через полгода, он понял, какой подарок прикреплен к какому празднику, кому это дарить, а также посмотреть дату начала события.

3. Обзор средств программирования

Изначально стоял между 3-я редакторами кода. WebStorm, Sublime Text 3, Visual Studio Code. Расскажу немного о каждом.

1. WebStorm от JetBrains прекрасен в обоих своих проявлениях: как IDE он поддерживает работу с системами контроля версий, позволяет удалённо развернуть код, как редактор — стандартные удобства, вроде подсветки синтаксиса, автодополнений, навигации.

Преимущества:

* LiveEdit — просмотр внесённых в код изменений без необходимости его сохранять;
* взаимодействие с фреймворками, например, React, Angular, Meteor;
* больше сотни встроенных тестов для обнаружения ошибок;
* интегрирование с Mocha, Protractor, Jest, Karma для юнит-тестов;
* полномасштабный дебаггер для отладки кода на серверной и клиентской сторонах;
* навигация для одновременной работы с несколькими файлами;
* автодополнение кода, подсветка синтаксиса.

Недостатки:

* стоит 129 $ за первый год работы;
* для начинающих функционал избыточен.

Это был неплохой вариант, но, по-моему, мнению он занимает 3-е место, так как достаточно тяжелое IDE предназначенная для масштабных проектов. И весь функционал не был бы даже затронут в нашем случае.

1. Sublime Text 3 - удобный и проверенный временем кроссплатформенный редактор, с настраиваемым интерфейсом и возможностью совершать тривиальные действия при помощи горячих клавиш.

Преимущества:

* горячие клавиши;
* навигация по коду в виде мини-карты;
* возможность изменить визуальную тему;
* поддержка сниппетов;
* подсветка, автодополнение переменных и синтаксиса;
* множественная правка благодаря использованию указателей;
* поддержка систем сборки;
* проверка синтаксиса прямо во время ввода;
* очень много плагинов;
* автосохранение.

Недостатки:

* полная версия стоит 70 $;
* отсутствие анализатора кода для расстановки ссылок.

Это отличный редактор, очень быстрый, легкий, функциональный, для создания сайтов подходит идеально, но, к сожалению, не для веб-приложений. Даже несмотря на то, что Sublime предоставляет большое кол-во плагинов и должен был стать идеальным вариантом, но функционала этих плагинов было недостаточно, чтобы создавать веб-приложение на основе фреймворка React JS. Большинство плагинов оказались не очень качественными. Именно поэтому он занимает 2-е место.

1. Visual Studio Code он прост для освоения, удобен в использовании, и при этом функционален.

Преимущества:

* контекстное автодополнение, как синтаксиса, так и используемых переменных, модулей, функций и т. д.;
* дебаггер с точками останова, стеком вызовов, интерактивной консолью;
* поддержка сниппетов и шаблонов;
* интеграция с Git;
* удобный и простой интерфейс;
* бесплатный редактор.

Недостатки

* не очень удобен для верстки.

VS Сode стал идеальным решением для создания нашего приложения. В нем было все, что нужно.

1. Обоснование выбора языка программирования
2. Возможность изучить и испробовать что-то новое
3. Веб-сайты в основном служат для информационных целей (Wikipedia), а веб-приложения — это интерактивные сервисы, то есть это означает, что пользователь является не пассивным потребителем, а активным участником (заполняет формы, совершает покупки, отправляет данные)

У нас веб-приложение потому что пользователь должен сам что-то делать, а не читать информацию.  
Мы выбрали веб-приложение, так как у нас не стояла цель создать информационный или новостной портал, где публиковались бы статьи на определенную или разные тематики. Нам нужно было создать такой ресурс, который бы решал проблемы пользователей и одновременно был интерактивным для взаимодействия с ними.

Для начала стоит определиться: SPA или нет?

Есть два типа веб-приложений

1. SPA (single page application) – одностраничное приложение.
2. MPA (multi page application) – многостраничное приложение.

В первом случае весь контент будет динамически генерироваться на одной странице. Это означает, что страница не будет обновляться, а лишь раз загрузится и будет получать данные, обрабатывать и отображать благодаря JavaScript (далее JS). Второй вариант это традиционный. Когда для каждой страницы создается отдельный файл.

Использовать мы будем React, который поддерживает routing, поэтому SPA.

Как уже говорилось, для интерактивности использовали JS, а точнее его библиотеку React в связке с Redux.

Почему именно React?

React — это JS-библиотека для создания пользовательских интерфейсов разработанная Facebook. React является составляющей V в аббревиатуре MVC.

* Простой. React позволяет просто выразить, как ваше приложение должно выглядеть в любой момент времени и автоматически управляет всеми обновлениями интерфейса при изменении данных приложения.
* Декларативный. При изменении данных React нажмет воображаемую кнопку «Обновить» и обновит только измененные части.
* Создание комбинируемых компонентов. В основе React — создание переиспользуемых компонентов. Фактически, в React вам всего лишь нужно создать необходимые компоненты. Компоненты настолько инкапсулированы, что позволяют легко переиспользовать и тестировать код.
* Virtual DOM. Повышает производительность, что может снизить вероятность возникновения возможных неудобств и улучшает user experience (UX).
* Использование изоморфного подхода. Помогает производить рендеринг страниц быстрее, тем самым позволяя пользователям чувствовать себя более комфортно во время работы с приложением. Поисковые системы индексируют такие страницы лучше. Поскольку один и тот же код может быть использован как в клиентской, так и в серверной части приложения, нет необходимости в дублировании одного и того же функционала. В результате время разработки и затраты снижаются.

Redux — это одна из реализаций Flux, инструмент управления как состоянием данных, так и состоянием интерфейса в JS-приложениях. Он подходит для одностраничных приложений, в которых управление состоянием может со временем становиться сложным. Redux не связан с каким-то определенным фреймворком, и, хотя разрабатывался для React, может использоваться с Angular, Vue или jQuery.