

каждому говорить о том, что ему требуется. С одной стороны, от этого может пострадать качество исполнения работ, потому что более узкие специалисты смогут сделать это лучше, но это одновременно и очень затратное дело, не каждый малый или средний бизнес сможет пойти на такие расходы и поэтому готовы работать с одним разработчиком.

Возьмем на примере стандартный необходимыми большинству торговых компаний для удобного распространения услуг или товаров инструмент – интернет-магазин (сайт), он так же может выполнять задачу как корпоративный сайт, который представляет бренд, и чаще всего это лицо компании. В большинстве случаев, как говорится “встречаем по одежке”, но во фразе заменяем вторую часть “проводим по уму” на “и по уму”, вот к такому результату должен прийти разработчик по окончании разработки, чтобы все абстрактное – материализовать, что означает и дизайн хорош и функционал недалеко отстал.

Разработчику необходимо узнать лицо конечного клиента, он должен присутствовать с ним рядом или, ещё лучше, вживиться в его роль, чтобы точно знать, что ему надо и как было бы ему удобнее помочь найти то, что ему необходимо. Если мы говорим о дизайне, то сайт должен иметь приятные на глаз оттенки и сочетания цветов, а также размеры и цвет шрифтов. Потом, после уже дизайна, начинается этап исследования удобного для пользователя взаимодействия интерфейса системы. И если, это все смешать, мы получим очень неплохую “одежку”, но к этому надо ещё и добавить психологию: игра цветов компонентов управления системы. Тут все как в жизни, красный – нельзя, стоять, зеленый – можно ехать. Вот так же с интерфейсом системы, надо помогать клиенту ориентироваться что сейчас он это может сделать, а в другом случае остановить его, что если кнопка зеленая, значит она приведет его в нужное место. Но человеческий фактор никто не отменял, люди ошибаются и надо как можно больше сузить набор возможных ошибок пользователя. Если человек хочет сделать заказ с сайта, то предоставь ему несколько вариантов связаться с тобой, будь это номер телефона чтобы он мог позвонить, или избавьте его от этой процедуры, позвоните ему сами! Но никогда не обещайте ему позвонить за определенное время, лучше скажите правду, что перезвоните в кратчайшие сроки тем самым не обманите его, он будет держать в голове мысль о том, что процесс движется и он нем позаботятся. Все люди работают и все они устают, поэтому чаще всего они что-то ищут уже после заката солнца, когда уже большинство магазинов по закрывались и уже только на следующее утро только откроются, поэтому им необходимо тоже дать возможность, что-то найти и забрать. Такую долю клиентов тоже не стоит забыть, сделайте им удобную форму заказа, которая будет круглосуточно открыта для новых заявок. Упростите её настолько, насколько это возможно, только номер телефона и комментарий, но, когда вы делаете очень подробную форму заполнения информации, что ему надо обязательно вписать свою дату рождения, он устанет и уйдет в другую систему.

Помимо “одежки”, необходим и “ум” – функционал и логичность всей системы в целом. Сюда входят, как и все возможные фильтры при выборе какого-нибудь продукта, так и строка с “умным поиском” сверху, для максимального сокращения пути клиента к цели. Не стоит забывать и про “хлебные крошки” на каждой странице сайта, чтобы он всегда мог без лишних движений вернуться на шаг назад. И, как уже писалось выше про поколение сервисов, что клиенту надо все за него найти, сделайте ему блок, на основе его предыдущего опыта поиска в каталоге, рекомендуемые именно для него товары, но и ни стоит недооценивать и ту возможность, что ему в придачу к этому, можно и ещё что-нибудь порекомендовать, или то, что очень хорошо подходит к данному типу товара, что вместе обычно с ним ещё смотрят и другие товары.

Ну если с дизайном и функционалом уже определились, то необходима и смесь навыков продавца и маркетолога – это продвижение и вручение продукта клиенту. Никогда не стоит забыть про рекламу, любой разработчик, который соглашается конкурировать со студиями или компаниями по разработке ПО, должен знать эти основы рекламы для начального удачного старта проекта. Он должен так иметь навыки в копирайтинге, подготовке рекламных шаблонов, и анализе рынка спроса по всем торговым площадкам и площадка для размещения рекламы. Выявление и анализ для подсчета среднесуточного бюджета и умению разумно оптимизировать рекламу для товаров, потому что никому не хочется платить за воздух, и поэтому отслеживать это, и при отсутствии товара, выключать рекламу.

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ «СИСТЕМА ИНФОРМИРОВАНИЯ В ЭКСТРЕННЫХ СИТУАЦИЯХ»

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Гончаров Я.И.

*Пачинин В. И. – зав. кафедрой ИСuТ, к.т.н., доцент
Савенко А.Г. – ассистент каф. ПЭ, м.т.н.*

Основой для реализации задачи по информированию в экстренных ситуациях служит необходимость поддержки людей, попавших в экстренную и опасную для их жизни обстановку, а также обработка их данных о местоположении для помощи им в кратчайшие сроки. Создание сервиса экстренного оповещения на мобильные платформы, который представляет собой разнонаправленное решение, предоставляющее пользователям гибкую систему по организации собственной безопасности и личной неприкосновенности.

Главная задача системы – оповещение пользователей данной системы о находящихся рядом с ними зонах с потенциально опасными для их жизни и безопасности условиями, а также оповещение выбранных пользователем контактов о его местоположении в случае возникновения опасности для жизни.

Для удобства использования мобильного приложения разработан интуитивно понятный пользовательский интерфейс, что непременно улучшит отзывчивость пользователей и уменьшит время на освоение навыков работы с данной системой.

Все требующиеся для работы системы данные хранятся на уровне настроек большого приложения, что позволило отказаться от привычной структуры с использованием СУБД.

Для разработки мобильного приложения был использован фреймворк Xamarin. Xamarin – фреймворк для кроссплатформенной разработки мобильных приложений с использованием языка C#. Он представляет возможность применения любого языка программирования с полным доступом ко всем возможностям SDK платформы и механизму создания UI, получая на выходе приложение, которое имеет высокую производительность.

Разработанное мобильное приложение выполняет следующие функции:

- добавление/изменение/удаление экстренного оповещения;
- создание и редактирование времени отправки экстренного оповещения;
- оповещение о находящимся поблизости зонах с потенциально опасными условиями для жизни и личной безопасности;
- отправка экстренных оповещений выбранным контактам;
- создание и отправка экстренных оповещений на основании маршрута следования.

Список использованных источников:

1. Шарп, Д. Microsoft Visual C#. Подробное руководство / Д. Шарп – СПб.: Питер, 2017. – 848 с.
2. Албахари, Б. C# 7.0. Карманный справочник / Б. Албахари, Дж. Албахари – М.: Вильямс, 2017. – 224 с.
3. Xamarin [Электронный ресурс]. – Режим доступа: <https://developer.xamarin.com/>. – Дата доступа: 18.12.2017.

ПАТТЕРН УПРАВЛЕНИЯ СОСТОЯНИЕМ НА ПРИМЕРЕ VUEX

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Гулис Н.В.

Бакунова О.М., старший преподаватель каф. ИСиТ, м.т.н.

Состояние (от английского state) является поведенческим шаблоном проектирования. Такой шаблон применяется в тех ситуациях, когда для функционирования программы или приложения объекту необходимо изменять свое поведение в зависимости от состояния.

Паттерн (от английского pattern – диаграмма, схема, шаблон) – это схема или принцип решения определенной задачи при проектировании программ. В данной работе будет описан паттерн управления состоянием на примере его реализации на языке программирования JavaScript – Vuex.

Vuex – это паттерн для управления состоянием, а также библиотека для JavaScript-фреймворка «Vue.js». В большинстве случаев он используется при создании одностраничных web-приложений. Он представляет собой централизованное хранилище данных, доступ к которому имеют все компоненты приложения, или так называемый «единственный источник истины». Данные в хранилище изменяются согласно строго определенным правилам. Такой подход позволяет обеспечить предсказуемость изменений.

Хранилище во Vuex включает в себя четыре главных понятия – состояние, мутации, действия и геттеры.

Состояние во Vuex представляет собой JavaScript-объект, содержащий в себе информацию, используемую хранилищем. Изменять состояние напрямую строго запрещается. Все изменения состояния во Vuex производятся через специальные функции – мутации.

Мутации – функции предназначенные для изменения состояния приложения. Каждая мутация включает в себя строковый тип, или имя, и функцию-обработчик, в котором и совершаются изменения состояния, переданного в мутацию аргументом. В мутацию также можно передавать и другой аргумент, так называемую «нагрузку», представляющую собой произвольный объект. Пример объявления мутаций представлен на рисунке 1.

```
mutations: {
  [SET_PROFILE] (state, userInfo) {
    state.user.profile = userInfo;
  },
  [SET_ERROR] (state, error) {
    state.error = error;
    state.pending = false;
  },
  [LOGIN] (state) {
    state.pending = true;
  },
  [LOGIN_SUCCESS] (state) {
    state.user.authenticated = true;
    state.error = null;
    state.pending = false;
  }
}
```

Рисунок 1 – Пример объявления мутаций