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 больного 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 находящихся 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.

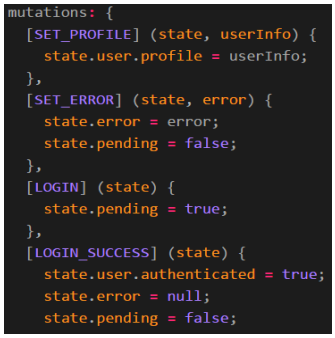

Рисунок 1 – Пример объявления мутаций

Типы, или имена, мутаций можно задать заранее константами либо переменными и вынести в отдельный документ. Такой подход улучшает читаемость кода, облегчает работу других разработчиков и дает представление о всех возможных мутациях для приложения.

Все мутации обязательно должны являться синхронными. Такое требование обусловлено тем, что приложение обязательно должно запоминать свое состояние до и после выполнения мутации. Данное ограничение позволяет отследить и заранее предугадать порядок выполнения мутаций. Это существенно облегчает задачу отладки приложения и позволяет отслеживать все изменения состояния. Для асинхронных операций же предусмотрены действия.

Вызов мутаций в компонентах приложения осуществляется с помощью функции «commit» либо при помощи специальных мапперов, привязывающих мутации к компоненту.

Действия в отличие от мутаций не вносят изменений напрямую в состояние приложения и могут содержать в себе асинхронные функции. Также действия могут использоваться для поочередного вызова мутаций. Асинхронность действий позволяет выполнять в них такие операции как HTTP-запросы или вызов API. Таким образом с помощью действий можно выстраивать цепочки асинхронных запросов, сохраняя полученную информацию в состоянии с помощью промежуточных мутаций. Действия также позволяют вызывать внутри себя другие действия. Вызов действий в компонентах выполняется с помощью функции «dispatch». Пример объявления действия представлен на рисунке 2.

Для получения информации, основанной на состоянии хранилища, во Vuex используются геттеры. Такой функционал необходим, если одна и та же информация из хранилища должна быть отображена в нескольких компонентах. Использование геттеров значительно уменьшает повторение кода и улучшает его читаемость. Примеры геттеров приведены на рисунке 3.

```
actions: {
  logIn({commit}, credentials) {
    commit(LOGIN);
    return new Promise((resolve) => {
      axios.post('api/login', credentials)
      .then((res) => {
        let token = res.data.token;
        localStorage.setItem('token', token);
        commit(LOGIN_SUCCESS);
        resolve();
      })
      .catch((err) => {
        commit(SET_ERROR, err.response.data.error);
      })
    })
  },
```

Рисунок 2 – Пример объявления действия

Хотя хранилище Vuex и представляет собой один большой объект, для удобства разработчика предусмотрена возможность разделять его на модули. Каждый модуль имеет свои собственные мутации, геттеры и состояние. Однако модули могут также получить доступ и к глобальному состоянию хранилища.

Любой модуль можно разделить на подмодули, таким образом структура хранилища являет собой фрактал.

```
getters: {
  authenticated: state => {
    return state.user.authenticated;
  },

  profile: state => {
    return state.user.profile;
  },

  error: state => {
    return state.error;
  }
}
```

Рисунок 3 – Примеры объявления геттеров

Vuex предоставляет собственное расширение для консоли разработчика в браузерах, которое позволяет отслеживать все изменения состояния в режиме реального времени. Помимо отслеживания, оно позволяет откатить любое изменение, а также проверить состояние приложения до и после любого изменения. Данное расширение изображено на рисунке 4.

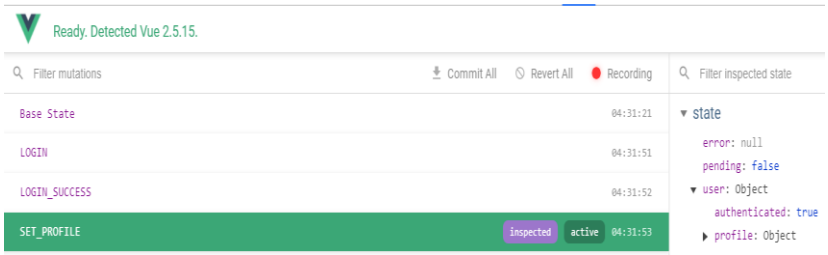

Рисунок 4 – Расширение для браузера «Vue»

Такое расширение значительно ускоряет процесс разработки, облегчает процесс отладки и поиска багов в приложении.

Vuex следует использовать в средних или больших по объему приложениях с большим количеством компонентов, в том числе использующих одинаковую информацию. Использование в небольших приложениях лишь усложнит их логику.

VUE.JS – АНАЛИЗ ФРЕЙМВОРКА

Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь

Гулис Н.В.

Образцова О.Н. – доцент каф. ИСиТ, , к.т.н., доцент
Бакунова О.М. – ст. преподаватель каф. ИСиТ
Бакунов А.М. - ст. преподаватель каф. ИСиТ
Калитеня И.Л., – ассистент каф. ИСиТ

В настоящее время трудно представить себе разработку актуального web-приложения без использования какого-либо фреймворка. Фреймворк (от английского framework – структура, каркас) – программная платформа, определяющая структуру программной системы, и облегчающая разработку больших программных проектов и их компонентов. В последние годы было написано множество различных фреймворков имеющих разную степень востребованности и качества. В работе будет рассмотрен один из фреймворков для языка JavaScript – Vue.js.

Vue.js был создан бывшим работником Google Эваном Ю и предназначен для разработки пользовательских интерфейсов. Впервые фреймворк был представлен в 2014 году. Он активно набирает популярность, о чем свидетельствует растущее количество его скачиваний, с января по декабрь 2017 года выросшее с 250 тысяч до 1 миллиона. График количества скачиваний Vue.jsпредставлен на рисунке 1. Также о популярности данного фреймворка свидетельствует большое количество крупных компаний, совершивших переход на его использование, таких как Nintendo, GitLab, Baidu.

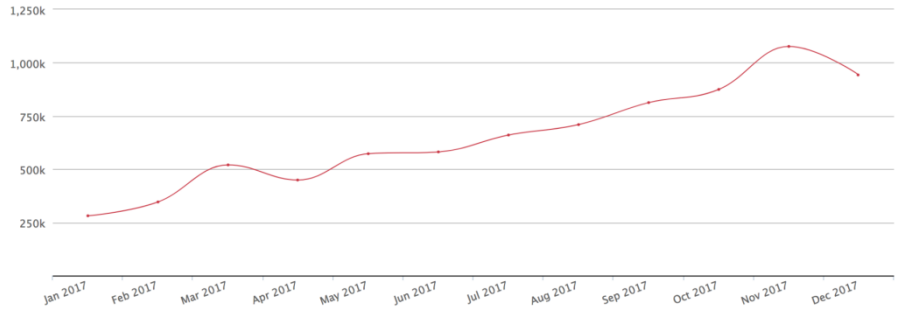

Рисунок 1 – График количества скачиваний Vue.js

Vue.jsпоставляется по лицензии MITдля открытого программного обеспечения (другое название «Лицензия X11»). Данная лицензия предоставляет полное право использования, модифицирования, копирования, дистрибуции и публикации программных продуктов, поставленных по этой лицензии[1].

Vue.js обладает высокой производительностью благодаря малому размеру (23кб в архивированном состоянии),а также благодаря использованию виртуального DOM.DOM(от английского Document Object Model– объектная модель документа)–это программный интерфейс, позволяющий скриптам и программам получать доступ к элементам HTML, XMLи XHTML-документов. Виртуальный DOMявляется более легковесной копией оригинального. Все изменения первоначально вносятся в копию, затем копия и оригинал сравниваются, и происходит перерисовка только измененных компонентов.

Данный фреймворк зачастую используется для создания одностраничных веб-приложений, однако может быть внедрен и в уже существующий проект на любом этапе разработки или сопровождения. Он