

После подбора данного усиления игроком, его огневая мощь будет увеличена на короткий промежуток времени, а именно вместо стандартного одного лазера, выпускаемого при нажатии клавиши огня, будет выпущено три. Эффект от данного усиления представлен на рисунке 6.

В ходе работы создано игровое приложение, которое взаимодействует с пользователем, принимая от последнего различные инструкции. Приложение «Galaxy Shooter», позволяющее игроку поучаствовать в космической баталии в качестве пилота космического корабля.



Рисунок 6 – Игрок с бонусом усиления огневой мощи

Игровое приложение было построено с использованием высокоуровневого объектно-ориентированного языка С# [1], игрового движка Unity3D [2] и межплатформенной среды разработки компьютерных игр Unity [3]. Приложение работает под операционной системой Windows без установки специализированных дистрибутивов.

Литература

1 Официальная документация С# – С# Guide : [Электронный ресурс] // URL: <https://docs.microsoft.com/en-us/dotnet/csharp/>. – Дата доступа: 30.01.2019.

2 Официальная документация Unity – Unity Manual: [Электронный ресурс] //URL: <https://docs.unity3d.com/ru/current/Manual/index.html>. – Дата доступа: 05.02.2019.

3 Официальная документация Unity – Unity Courses: [Электронный ресурс] // URL: <https://unity3d.com/ru/learn/courses>. – Дата доступа: 29.02.2019.

УДК 004.7

Н. С. Буйновец

СОЗДАНИЕ ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ УПРАВЛЕНИЯ ЗАДАНИЯМИ С ИСПОЛЬЗОВАНИЕМ ФРЕЙМВОРКА OPENUI5

Статья посвящена разработке приложения «Менеджер заданий», который предоставляет полный функционал для отображения, добавления, редактирования и удаления записей. Приложение хранит следующую информацию: название, статус, приоритет, заказчика, дату получения и окончания задания и статус выполнения задания в процентах. Для хранения и вызова данных приложения разработан Mock Server. При создании и редактировании записей осуществляется проверка всех вводимых данных на корректность. Приложение адаптировано под маленькие экраны. Web-приложение создано с использованием технологий HTML, CSS, JavaScript, OpenUI5.

Архитектура веб-приложения соответствует шаблону проектирования Model-View-Controller. Модели предоставляют данные и реагируют на команды контроллера, изменяя свое состояние. Контроллер отвечает за обработку действий пользователя, оповещая модель о необходимости изменений. Контроллером в данном приложении является разработанный файл `MasterPage.controller.js` и `DetailPage.controller`. Представление отвечает за получение необходимых данных и визуализацию данных модели, реагируя на изменения модели. Представлением в приложении является файл `MasterPage.view.xml` и `DetailPage.view.xml`. Для хранения и вызова данных приложения разработан и использован Mock Server. При создании / редактировании записей осуществляется валидация всех вводимых данных.

Для работы приложения необходимо запустить проект в среде разработки Visual Studio Code, как показано на рисунке 1.

```
getDialog : function () {
  if (!this.dialog) {
    this.dialog = sap.ui.xmlfragment("fragmentForAdd", "listoftasknew.fragments.addDialog", this);
    this.getView().addDependent(this.dialog);
  }
  return this.dialog;
},

addPress: function() {
  var dialog = this.getDialog();
  var i18nModel = this.getResourceBundle();
  dialog.setTitle(i18nModel.getText("DIALOG_ADD"));
  var oTask = {
    key: "",
    title: "",
    status: "",
    percent: "",
    priority: "",
    customer: "",
    startdate: ""
  };
}
```

Рисунок 1 – Запуск приложения в Visual Studio Code

После запуска в браузере приложения пользователь попадает на главную страницу приложения, как показано на рисунке 2.

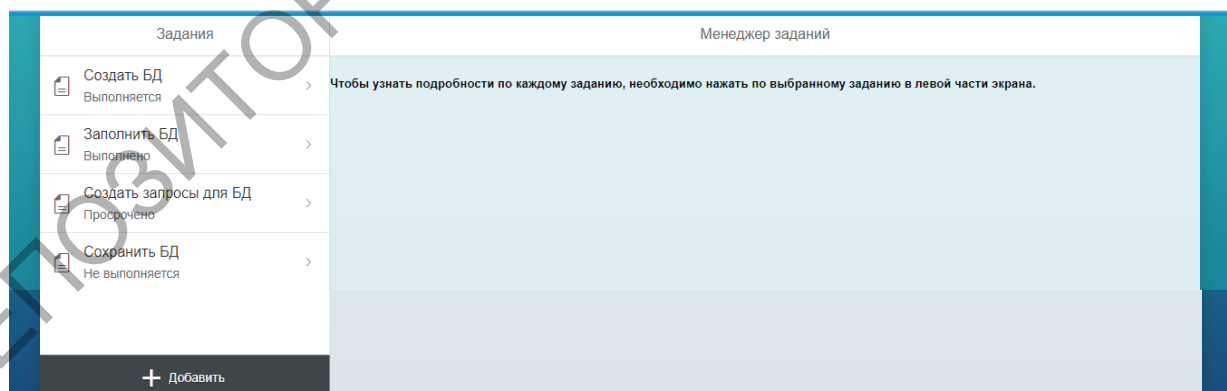


Рисунок 2 – Главная страница приложения

Для того, чтобы добавить новую запись, необходимо выбрать кнопку «Добавить». После нажатия откроется диалог с полями для ввода данных, как показано на рисунке 3.

Добавить запись	
*Название	Протестировать
*Статус	Выполняется
*Приоритет	Высокий
*Заказчик	сотрапу
*Дата получения задания	13.02.2019
*Срок выполнения	28.03.2019
Сохранить Закрыть	

Рисунок 3 – Страница добавления задания

Для создания диалога использовался компонент `sap.m.Dialog`. При попытке добавления записи с пустым обязательным для заполнения полем, появится уведомление о том, что обязательные поля не были заполнены. Результат добавления записи показан на рисунке 4.

При просмотре дополнительных сведений пользователю выводятся: название задание, статус выполнения, приоритет, заказчик, дата получения и срок выполнения. Также отображается статус задачи в процентах, который подсвечивается разными цветами в зависимости от сроков и текущей готовности задания.

Задания	Параметры
Создать БД Выполняется	Протестировать Название: Протестировать Статус: Выполняется 32.6 % Приоритет: Высокий Заказчик: сотрапу Дата получения задания: 13.02.19 Срок выполнения: 28.03.19
Заполнить БД Выполнено	
Создать запросы для БД Просрочено	
Сохранить БД Не выполняется	
Протестировать Выполняется	
+	

Рисунок 4 – Запись после добавления

Для изменения записи требуется перейти на необходимую запись и нажать на кнопку «Изменить». После чего откроется диалоговое окно, в котором необходимо произвести соответствующие действия и подтвердить результат изменений.

Для удаления записи требуется перейти на необходимую запись и нажать на кнопку «Удалить». Результат удаления записи показан на рисунке 5.

Задания	Менеджер заданий
Создать БД Выполняется	Чтобы узнать подробности по каждому заданию, необходимо нажать по выбранному заданию в левой части экрана.
Заполнить БД Выполнено	
Создать запросы для БД Просрочено	
Сохранить БД Не выполняется	

Рисунок 5 – Список заданий после удаления записи

На главной странице был разработан кастомный компонент, основывающийся на компоненте Text. При нажатии по нему вызывается событие, которое меняет цвет текста на случайный цвет при клике по компоненту. Результат работы компонента представлен на рисунке 6.

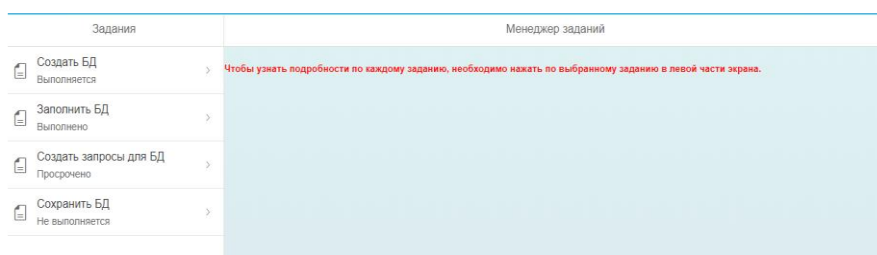


Рисунок 6 – Главная страница после нажатия по компоненту

В ходе работы написано приложение «Менеджер заданий» для управления заданиями с использованием технологий: HTML, CSS, JavaScript, OpenUI5 [1–2]. Приложение хранит следующую информацию: название, статус, приоритет, заказчика, дату получения и окончания задания и статус выполнения задания в процентах. Также приложение выполняет все основные действия с записями: добавление, удаление, изменение. Приложение способно выполнять валидацию вводимых данных, интернационализацию i18n и адаптировано под маленькие экраны. Разработанное приложение протестировано.

Приложение создано при помощи среды разработки Visual Studio Code.

Литература

1 Официальная документация OpenUI5 – OpenUI5 Documentation : [Электронный ресурс] // URL: <https://openui5.hana.ondemand.com/>. – Дата доступа: 28.01.2019.

2 SAP UI5 Tutorial ; [Электронный ресурс] // URL: https://www.tutorialspoint.com/sap_ui5 – Дата доступа: 25.02.2019.

УДК 004.67

М. Н. Гавриленко

РАЗРАБОТКА СИСТЕМ ОБРАБОТКИ И АНАЛИЗА ДАННЫХ С ИСПОЛЬЗОВАНИЕМ ИНСТРУМЕНТОВ BIG DATA

В статье рассматриваются вопросы актуальности использования новых технологий BigData для анализа и обработки информации. Функциональные возможности использования подобных технологий и языков программирования рассматриваются на примере разработки проекта по обработке различных данных в реальном времени, а также отображения информации в Angular приложении. Рассматриваются перспективы использования брокера-сообщений Kafka, а также Spark фреймворк для обработки и хранения информации.

Актуальность создания новых проектов и приложений в настоящее время трудно переоценить, ведь последнее время все данные, которые нас окружают, хранятся и записываются куда-либо, будь то облачные хранилища или обычные данные на