Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

Югорский государственный университет

Институт цифровой экономики

КУРСОВАЯ РАБОТА

По дисциплине «Языки программирования высокого уровня»

«Разработка игры в жанре TowerDefense в Unity3D»

Студенты гр. 1191б\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/А.Н.Свита/

(подпись)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/С.Н.Шергин/

(подпись)

Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

г. Ханты-Мансийск

1. г.

Оглавление

[Ведение 2](#_Toc75717517)

[2. NativeScript Vue 4](#_Toc75717518)

[Определение 4](#_Toc75717519)

[Достоинства и недостатки Vue 4](#_Toc75717520)

[3. Разработка мобильного приложения 5](#_Toc75717521)

[Создание окна регистрации 5](#_Toc75717522)

[Опции 5](#_Toc75717523)

[Подбор меню 6](#_Toc75717524)

[4. Заключение 7](#_Toc75717525)

[Список используемых источников 8](#_Toc75717526)

[Приложение А 9](#_Toc75717527)

[Приложение Б 11](#_Toc75717528)

# Ведение

Актуальность данной темы состоит в том что в нашей жизни составление правильного и грамотного рациона питания может улучшить здоровье и возможно найти что-то новое. И потому был разработана мобильное приложение с ограничителем в питании.

Цель работы: разработать мобильное приложение по питанию

Задачи:

1. Разобраться что такое NativeScript Vue.
2. Оформления вида самого приложения и функионала.
3. Тестирование приложения на ее работу.

# NativeScript Vue

## Определение

С начало узнаем, что такое NativeScript и Vue. NativeScript – это фреймворк с открытым исходным кодом для создания по-настоящему нативных мобильных приложений с использованием JavaScript. Vue – это прогрессивный фреймворк для создания пользовательских интерфейсов. Его ядро в первую очередь решает задачи уровня представления (view), что упрощает интеграцию с другими библиотеками и существующими проектами. Тогда NativeScript-Vue – это плагин для NativeScript, который дает вам возможность ипспользовать Vue.js для создания мобильного приложения.

## Достоинства и недостатки Vue

В достоинства Vue можно включить: довольно простую и функциональную библиотеку, фреймворк мало весит, требования к стеку отсутствуют, поэтому Vue.JS можно использовать на любом проекте, Возможность найти и подключить к проекту практически любого разработчика, кто хоть немного знаком с фронтенд разработкой. Низкий порог вхождения позволяет работать с фреймворком, как фронтендщикам, так и бэкэндщикам. Vue позволяет создавать функциональные приложения, соответствующие всем современным стандартам, при минимальном подключении новых ресурсов и, собственно, дешевле.

Также кроме достоинств есть недостатки и это: работа над состоянием приложения происходит “под капотом”. В масштабных проектах это может вылиться в неочевидность работы компонентов, что зачастую создает проблемы с отладкой и эффективной разработкой, шаблонизация React значительно гибче (JS vs DSL).

# Разработка мобильного приложения

## Создание окна регистрации

Создаем файл по имени App.vue после создаем два компонента TextField и Button и задаем в компоненте Button @tap = «gg» и в самом методе пишем [Листинг кода Б. 1] который переходит на другую страницу.

Листинг кода 1 – App.vue

<script >

import MenuVue from './Menu.vue';

export default {

data() {

return {

isLoggingIn: true,

user: {

email: "",

password: "",

confirmPassword: ""

},

};

},

methods: {

gg(){

this.$navigateTo(MenuVue, { clearHistory: true });

}

}

};

</script>

## Опции

В опции мы создаем компонент TabView в катором будут показаны два столбца «Основное» и «Предпочтения». В основном будут использоваться обычные компоненты, которые можно посмотреть в [Листинг кода Б. 2]. Но есть компонета которого нет в Vue, но есть в NativeScript TimePickerField, чтобы этот компонет появился нужно скачть сам плагин и написать в файле main.js [Листинг кода Б. 3].

Листинг кода 3 – Main.js

import DateTimePicker from '@nativescript/datetimepicker/vue'

Vue.registerElement('TimePickerField',()=>require('@nativescript/datetimepicker').TimePickerField)

## Подбор меню

В подбор меню мы опять возьмем новы компонент Carousel который активируется также в main.js [Листинг кода Б. 3]. В нем мы представляем наше меню из трех блюд, день недели, и изображение, которое при нажатии добавляет его в текущее меню [Листинг кода Б. 4] (Рис.4). После нажатия на меню нас перекинет нас на список блюд по неделям с названиями этих блюд и изображениями [Листинг кода Б. 5] (Рис. 5).

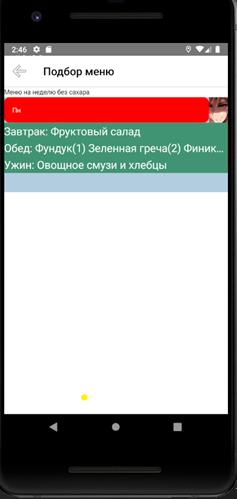


Рисунок 4 – Подбор меню

# Заключение

На данном этапе было созданы основные функции, которые нужны в первую очередь, в дальнейшем будут созданы отображение списка продуктов меню, которое выбрал сам пользователь. Кроме того, мы познакомились с NativeScript Vue и в дальнейшем буду его использовать в новых проектах.

# Список используемых источников

1. NativeScript Vue [Электронный ресурс] Режим доступа: nativescript-vue.org/ru/docs/introduction/#:~:text=NativeScriptVue%20%20это%20плагин%20для,как%20дома%20с%20NativeScript-Vue.
2. NativeScript [Электронный ресурс] Режим доступа: nativescript.org

# Приложение А

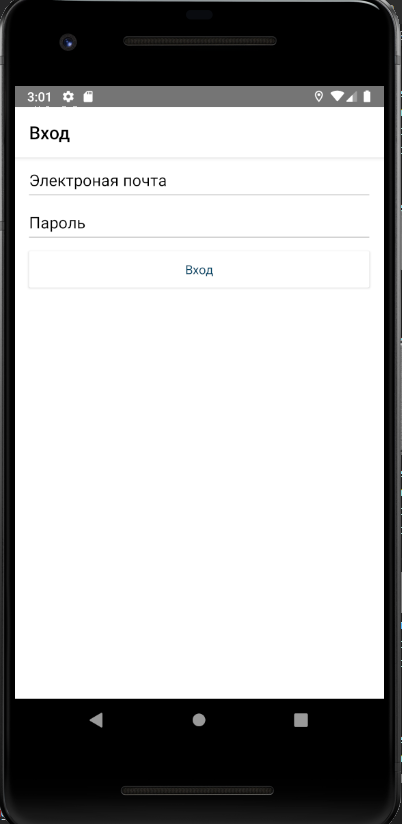


Рисунок А.1. – Вход

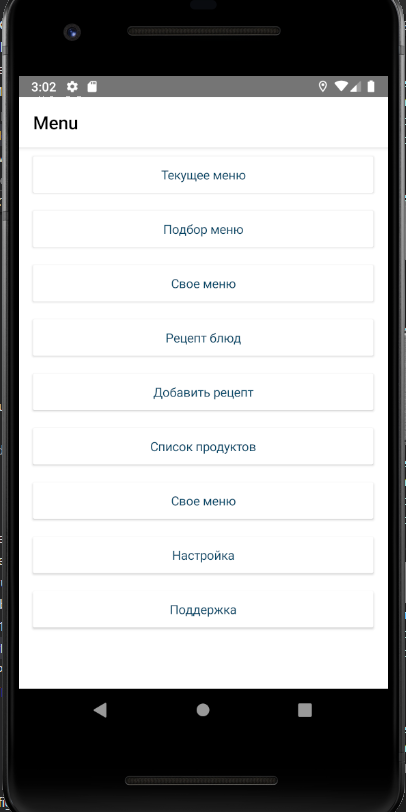


Рисунок А.2. – Меню

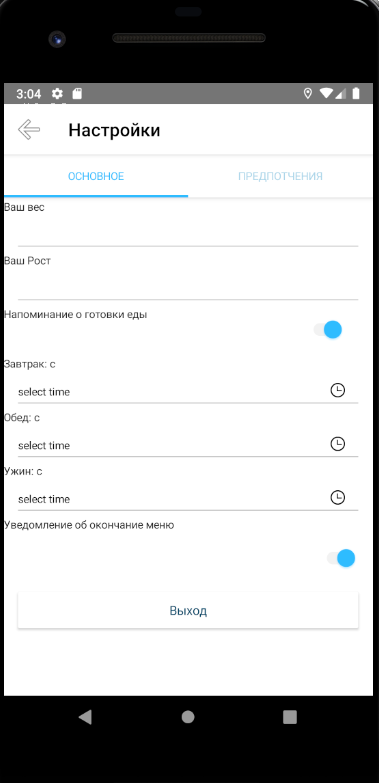


Рисунок А.3. – Опции

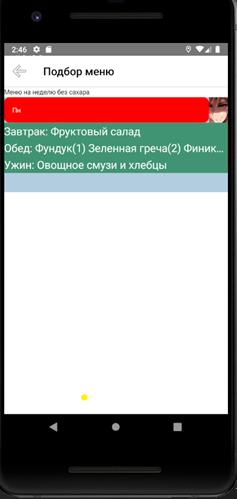


Рисунок А.4. – Подбор меню

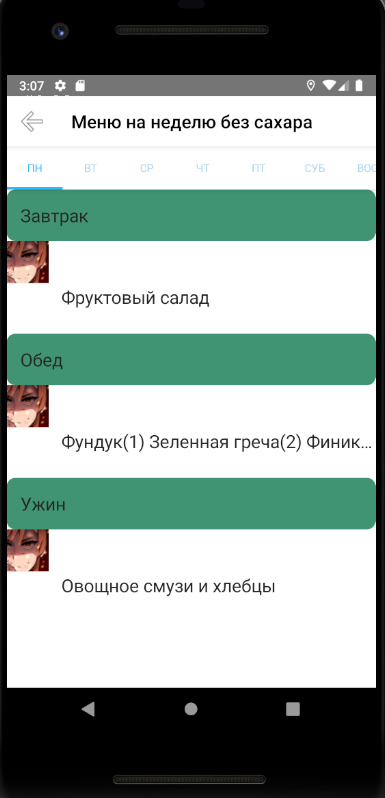


Рисунок А.5. – Меню продуктов

# Приложение Б

Листинг кода Б.1 – App

<template>

<Page class="app">

<ActionBar title="Вход"/>

<StackLayout class="page" >

<StackLayout class="form" >

<StackLayout class="input-field" >

<TextField class="input" hint="Электроная почта" keyboardType="email" autocorrect="false" autocapitalizationType="none" v-model="user.email"

returnKeyType="next" @returnPress="focusPassword" fontSize="18" />

<StackLayout class="hr-light" />

</StackLayout>

<StackLayout class="input-field">

<TextField ref="password" class="input" hint="Пароль" secure="true" v-model="user.password" :returnKeyType="isLoggingIn ? 'done' : 'next'"

@returnPress="focusConfirmPassword" fontSize="18" />

<StackLayout class="hr-light" />

</StackLayout>

<Button text="Вход" @tap="gg"/>

</StackLayout>

</StackLayout>

</Page>

</template>

<script >

import MenuVue from './Menu.vue';

export default {

data() {

return {

isLoggingIn: true,

user: {

email: "",

password: "",

confirmPassword: ""

},

};

},

methods: {

gg(){

this.$navigateTo(MenuVue, { clearHistory: true });

}

}

};

</script>

<style scoped>

</style>

Листинг кода Б.3 – Option1

<template>

<Page class="app">

<ActionBar title="Настройки">

<NavigationButton text="Go back" android.systemIcon="ic\_menu\_back" @tap="goBack"/>

</ActionBar>

<StackLayout >

<TabView>

<TabViewItem title="Основное">

<StackLayout>

<Label text="Ваш вес"/>

<TextField/>

<Label text="Ваш Рост"/>

<TextField/>

<GridLayout columns="\*, 15" rows="auto,auto">

<Label text="Напоминание о готовки еды"/><Switch checked="true"/>

</GridLayout>

<Label text="Завтрак: с "/> <TimePickerField hint="select time"/>

<Label text="Обед: с "/> <TimePickerField hint="select time"/>

<Label text="Ужин: с "/> <TimePickerField hint="select time"/>

<Label text="Уведомление об окончание меню"/><Switch checked="true"/>

<Button text="Выход" @tap="gg"/>

</StackLayout>

</TabViewItem>

<TabViewItem title="Предпотчения">

<StackLayout >

<Label text="Без лактозы"/><Switch checked="true"/>

<Label text="Без сахара"/><Switch checked="true"/>

<Label text="Белки"/><Switch checked="true"/>

<Label text="Веганство"/><Switch checked="true"/>

<Label text="Полезssные жиры"/><Switch checked="true"/>

<Label text="Сыроединие"/><Switch checked="true"/>

<Label text="Без глютена"/><Switch checked="true"/>

</StackLayout>

</TabViewItem>

</TabView>

</StackLayout>

</Page>

</template>

<script >

import AppVue from './App.vue';

import MenuVue from './Menu.vue';

export default {

data() {

return {

selectedTime:null

};

},

methods: {

goBack(){

this.$navigateTo(MenuVue);

},

gg(){

this.$navigateTo(AppVue, { clearHistory: true });

}

}

};

</script>

Листинг кода Б.3 – Main.js

import Vue from 'nativescript-vue'

import App from './components/App'

import VueDevtools from 'nativescript-vue-devtools'

import DateTimePicker from '@nativescript/datetimepicker/vue'

Vue.use(DateTimePicker)

if(TNS\_ENV !== 'production') {

Vue.use(VueDevtools)

}

// Prints Vue logs when --env.production is \*NOT\* set while building

Vue.config.silent = (TNS\_ENV === 'production')

Vue.registerElement('Carousel', () => require('nativescript-carousel').Carousel)

Vue.registerElement('CarouselItem', () => require('nativescript-carousel').CarouselItem)

Vue.registerElement('TimePickerField',()=>require('@nativescript/datetimepicker').TimePickerField)

new Vue({

render: h => h('frame', [h(App)])

}).$start()

Листиниг кода Б.4 – PodborMenu

<template>

<Page class="app">

<ActionBar title="Подбор меню">

<NavigationButton text="Go back" android.systemIcon="ic\_menu\_back" @tap="goBack"/>

</ActionBar>

<StackLayout>

<Label text="Меню на неделю без сахара"/>

<GridLayout>

<Carousel height="30%" width="100%" pageChanged="myChangeEvent" pageTapped="mySelectedEvent" indicatorColor="#fff000" finite="true" bounce="false" showIndicator="true" verticalAlignment="top" android:indicatorAnimation="swap" color="white">

<CarouselItem id="slide1" backgroundColor="#b3cde0" verticalAlignment="middle">

<GridLayout columns="\* ,35" rows="auto,auto">

<Label row="0" col="0" class="nedel" text="Пн" backgroundColor="red"/>

<Image row="0" col="1" horizontalAlignment="right" @tap="save" src="res://ssss" stretch="none"/>

</GridLayout>

<StackLayout @tap="eda">

<Label class="menu" text="Завтрак: Фруктовый салад"/>

<Label class="menu" text="Обед: Фундук(1) Зеленная греча(2) Финики(3)"/>

<Label class="menu" text="Ужин: Овощное смузи и хлебцы" />

</StackLayout>

</Carousel>

</GridLayout>

</StackLayout>

</Page>

</template>

<script >

import MenuVue from './Menu.vue';

import MenuVblborVue from './MenuVblbor.vue';

import TekuchiMenuVue from './TekuchiMenu.vue';

export default {

data() {

return {

selectedTime:null

};

},

methods: {

eda(){

this.$navigateTo(MenuVblborVue);

},

goBack(){

this.$navigateTo(MenuVue);

},

save(){

this.$navigateTo(TekuchiMenuVue);

}

}

};

</script>

<style scoped>

.nedel{

border-radius: 10%;

padding: 40px;

position: absolute;

bottom: 45px ;

background-color: #3f9473;

}

.menu{

position: absolute;

bottom: 45px ;

left: 15px;

background-color: #3f9473;

font-size:20%;

}

</style>

Листинг кода Б.5 – MenuVblbor

<template>

<Page class="app">

<ActionBar title="Меню на неделю без сахара">

<NavigationButton text="Go back" android.systemIcon="ic\_menu\_back" @tap="goBack"/>

</ActionBar>

<StackLayout>

<TabView>

<TabViewItem title="Пн">

<StackLayout class="ych">

<Label class="tabl" text="Завтрак"/>

<Image src="res://ssss" stretch="none"/>

<Label class="mesto" text="Фруктовый салад"/>

<Label class="tabl" text="Обед"/>

<Image src="res://ssss" stretch="none"/>

<Label class="mesto" text="Фундук(1) Зеленная греча(2) Финики(3)"/>

<Label class="tabl" text="Ужин"/>

<Image src="res://ssss" stretch="none"/>

<Label class="mesto" text="Овощное смузи и хлебцы"/>

</StackLayout>

</TabViewItem>

<TabViewItem title="Вт">

<StackLayout>

</StackLayout>

</TabViewItem>

<TabViewItem title="Ср">

<StackLayout>

</StackLayout>

</TabViewItem>

<TabViewItem title="Чт">

<StackLayout>

</StackLayout>

</TabViewItem>

<TabViewItem title="Пт">

<StackLayout>

</StackLayout>

</TabViewItem>

<TabViewItem title="Суб">

<StackLayout>

</StackLayout>

</TabViewItem>

<TabViewItem title="Воск">

<StackLayout>

</StackLayout>

</TabViewItem>

</TabView>

</StackLayout>

</Page>

</template>

<script >

import PodborMenuVue from './PodborMenu.vue';

export default {

data() {

return {

};

},

methods: {

goBack(){

this.$navigateTo(PodborMenuVue)

},

}

};

</script>

<style scoped>

.ych{

position: absolute;

left: 15px;

font-size:20%;

}

.mesto{

margin-left: 160px;

margin-bottom: 67px;

}

.tabl{

border-radius: 10%;

padding: 40px;

position: absolute;

bottom: 45px ;

background-color: #3f9473;

}

</style>