**4.3 Тестування програмних модулів**

**4.3.1 Тестування класу Guest**

**4.3.1.1 Тестування методу Authorization()**

Guest.Authorization() представляє собою метод, що перевіряє введені користувачем дані під час авторизації та проводить їх валідацію. Реалізація цього методу:

import android.app.Dialog

import android.content.Intent

import android.databinding.DataBindingUtil

import android.os.Bundle

import android.support.v7.app.AppCompatActivity

import android.widget.ListView

import android.widget.Toast

import com.android.volley.toolbox.Volley

import com.auth0.android.Auth0

import com.auth0.android.authentication.AuthenticationException

import com.auth0.android.provider.AuthCallback

import com.auth0.android.provider.WebAuthProvider

import com.auth0.android.result.Credentials

import com.auth0.samples.kotlinapp.databinding.ActivityMainBinding

class Authorization : Guest() {

var binding: ActivityMainBinding? = null

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

val queue = Volley.newRequestQueue(this)

binding = DataBindingUtil.setContentView(this, R.layout.activity\_main)

binding?.loggedIn = false

val listToDo = findViewById(R.id.list\_todo) as ListView

val loginButton = findViewById(R.id.login\_button)

loginButton.setOnClickListener { Authorization() }

}

override fun onNewIntent(intent: Intent) {

if (WebAuthProvider.resume(intent)) {

return

}

super.onNewIntent(intent)

}

private fun Authorization() {

val account = Auth0(getString(R.string.auth0\_client\_id), getString(R.string.auth0\_domain))

account.isOIDCConformant = true

WebAuthProvider.init(account)

.withScheme("demo")

.withAudience("StudyPosters")

.start(this, object : AuthCallback {

override fun onFailure(dialog: Dialog) {

runOnUiThread { dialog.show() }

}

override fun onFailure(exception: AuthenticationException) {

runOnUiThread {

Toast.AuthorizationPassword(

regexCheck = ^(?=.{1,50}

if (!regexCheck && password.length > 50)

this@MainActivity, "Пароль не совпадает!"

Toast.LENGTH\_SHORT).show()

Toast.AuthorizationEmail(

if (!requireNotNull(nickname).exist()){

this@MainActivity, "Пользователя с таким ником не существует!"

}

}

}

override fun onSuccess(credentials: Credentials) {

CredentialsManager.saveCredentials(this@MainActivity, credentials)

binding?.loggedIn = true

}

})

}

}

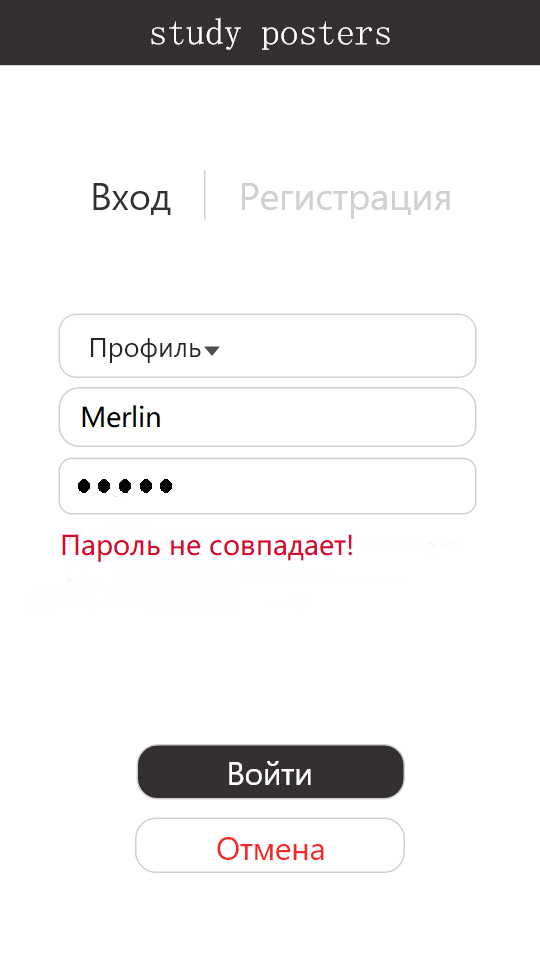


Рис. 1 – Перевірка паролю

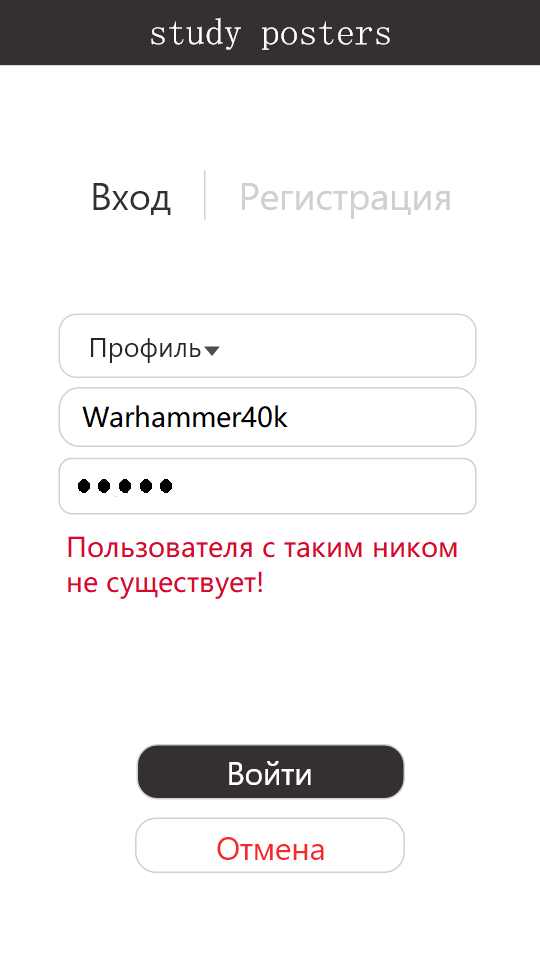


Рис. 2 – Перевірка існування користувача

|  |  |  |
| --- | --- | --- |
| № | Тестові набори | Очікувані результати |
| 1 | nickname = Merlin  password = admin | Пароль не совпадает! |
| 2 | nickname = Warhammer40k | Пользователя с таким ником не существует! |

**4.3.1.2 Тестування методу Registration()**

Guest.Registration() представляє собою метод, що перевіряє введені користувачем дані під час реєстрації та проводить їх валідацію. Реалізація цього методу:

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.text.TextUtils;

import android.util.Patterns;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

public class Registration : Guest() {

EditText firstLastName;

EditText email;

EditText password;

EditText confirmPassword;

EditText phone;

EditText nickname;

Button register;

Button cancel;

val fields = listOf(EditText);

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

firstLastName = findViewById(R.id.firstLastName);

nickname = findViewById(R.id.address);

email = findViewById(R.id.email);

password = findViewById(R.id.password);

confirmPassword = findViewById(R.id.confirmPassword);

phoneNumber = findViewById(R.id.phoneNumber);

register.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

checkDataEntered();

}

});

}

boolean isEmail(EditText text) {

CharSequence email = text.getText().toString();

return (!TextUtils.isEmpty(email) && Patterns.EMAIL\_ADDRESS.matcher(email).matches());

}

boolean isEmpty(EditText text) {

CharSequence str = text.getText().toString();

return TextUtils.isEmpty(str);

}

void checkDataEntered() {

if (firstName > 80) {

Toast t = Toast.makeText(this, "ФИО слишком длинное!", Toast.LENGTH\_SHORT);

t.show();

}

if (nickname.exist()) {

nickname.setError("Пользователь с таким никнеймом уже существует!");

}

if (requireNotNull(fields).forEach{it.keys}) {

fields.setError("Одно или несколько полей не заполнено!");

}

if (!confirmPassword.equals(password)) {

fields.setError("Пароли не совпадают!");

}

}

}

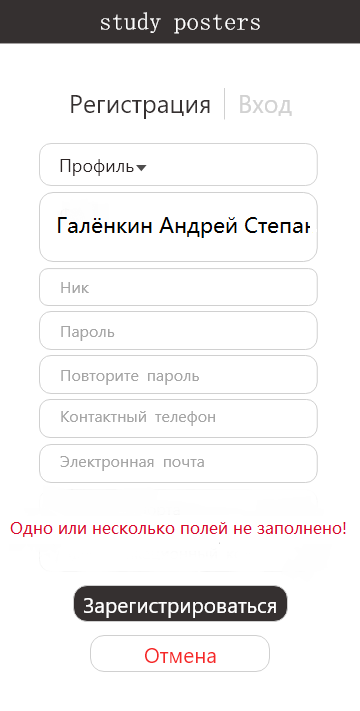


Рис. 3 – Перевірка на заповнення всіх полів

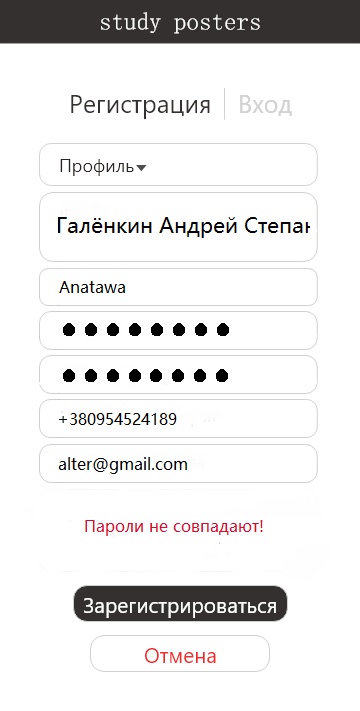


Рис. 4 – Перевірка збігу паролів

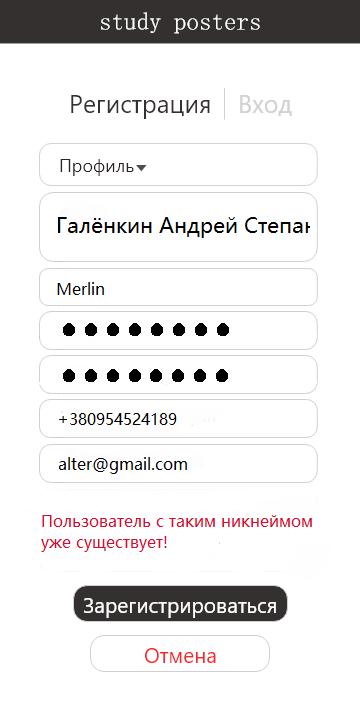


Рис. 5 – Перевірка на існування користувача

|  |  |  |
| --- | --- | --- |
| № | Тестові набори | Очікувані результати |
| 1 | firstLastName = Галёнкин Андрей Степанович  nickname = null  password = null  confirmPassword = null  email = null  phone = null | Одно или несколько полей не заполнено! |
| 2 | firstLastName = Галёнкин Андрей Степанович  nickname = Anatawa  password = 12345678  confirmPassword = 12345679  email = alter@gmail.com  phone = +380954524189 | Пароли не совпадают! |
| 3 | firstLastName = Галёнкин Андрей Степанович  nickname = Merlin  password = 12345678  confirmPassword = 12345678  email = alter@gmail.com  phone = +380954524189 | Пользователь с таким никнеймом уже существует! |