Online pénztárcára mobilalkalmazása

**A** szoftver célja, hogy a felhasználónak meglegyen az adott kényelme a szoftverben. Ezáltal mi is olyan fejlesztésen dolgoztunk/dolgozunk, amivel meg is tudjuk valósítani ezt. Könnyed feltöltést és adat tárolást tudunk biztosítani az szoftverrel. Biztonságos adat tárolást nem tudunk még biztosítani, fejlesztés alatt áll.

****A komponens megnevezése: mobilalkalmazás

**Az** Android Studio egy kifejezetten androidos rendszerrel rendelkező telefonra való szoftver fejlesztői program, amibe található beépített telefon(emulator) is, amivel könnyedén tudtuk tesztelni a programot kódolás során.

**Ennek** segítségével tudtuk megvalósítani a szoftverünket(mobilosalkalmazást). A kódolás kétfajta nyelven íródhatott az egyik java nyelven másik a kotlin nyelv lehetett volna. A két nyelv közül java-t választottuk mert ahhoz találtunk több segítséget és a tanórákon is ezt gyakoroltuk. PHP kódban írtuk meg a feltöltéshez szükséges kódot, ez a külső segítség egyszerűbb volt és átláthatóbb ezért is választottuk. Főként ezzel töltöttünk fel a tanórákon is adatokat az adatbázisba. így egyszerűbbnek láttuk a kód megírását a project munkánkhoz is.

**A** regisztráció oldalon találhatók azok az adatok, amikkel később a bejelentkezési oldalhoz szükségesek lesznek. Fontos, hogy nem itt vesszük fel a kártyák adatai (személyigazolvány vagy bankkártya), itt csak is a regisztrációhoz szükséges adatokat tároljuk majd el és visszük fel az adatbázisba.

**A** bejelentkezés oldalon található meg az a két adat, amit a regisztrációnál adunk meg. Ha sikeres a regisztráció vagy bejelentkezés akkor a telefon jelezni fog egy üzenettel, hogy sikerült-e a belépés/regisztráció: „Login Success” „Register Success”.

Miután bejelentkeztünk a fő oldalon találjuk magunkat, ahol két feltöltési lehetőség közül tudunk választani, egyik a bankkártya adatok feltöltése a másik az igazolványok feltöltésére szolgáló gombok.

**Mind** a bankkártya és az igazolványon is megtalálható a validáció. A validációk figyelik meg a helyes kitöltést az adott mezőben, mint például: ki van-e töltve a mező, helyes formátumban van-e megadva az adat, szám korlátozás minden mezőre (ahol szükséges). Ha minden adat jól lett megadva a kitöltési mezőkbe akkor a program feltölti az adatot az adatbázisba, ahol tároljuk az adatokat.  
**Fontos! Ebben a szoftverben nem lehet fizetni csak is adatok tárolására szolgál.**

**A** biztonság fejlesztése lesz a későbbiekben a fő szempontunk, mivel nem szeretnék a felhasználókat semmilyen veszélynek kitenni az interneten. Például kikerül bárhova bármilyen adatuk a szoftverünk által, amit mások fel tudnak használni a saját javukra, ezáltal károsítva a felhasználóinkat. Emiatt szeretnénk egy biztonságot nyújtó alkalmazást fejleszteni, amit bárki szívesen használna kötöttségek nélkül.

**További** fejlesztésünk majd a pontgyűjtő kártya felvétele és a vezetői engedély lesz. Egyéb kártyák hozzáadása a pénztárcához. Ez úgy működne, hogy az adott felhasználónak van egy különleges vagy klubtagsági kártyája és arra nincs megfelelő feltöltési mód, ez esetbe jönne be ez a lehetőség, hogy az adott felhasználó készíti el a feltöltést a szoftverben külön mezőt tudna felvenni és azt elnevezni ezáltal ő tudná le generálni az adott kártya adatainak a helyét és így feltölteni a központi adatbázisba.

|  |  |
| --- | --- |
| A képen szöveg, képernyőkép, elektronika látható  Automatikusan generált leírás  A mobilos szoftverünk fútatása néhány képben és magyarázatban (emulátoros megjelenítés) | A képen szöveg, elektronika, monitor, képernyőkép látható  Automatikusan generált leírás |
| Regisztrációs felület. Itt adatható meg az adat, amit később az adatbázisban látunk, ha a „register” gombra kattintunk. A „login\_here” gombbal tudunk a bejelentkezés felületre menni. | Regisztrációs oldal adatokkal való feltöltése. |
| A képen szöveg, elektronika, megjelenítés, képernyőkép látható  Automatikusan generált leírás | A képen szöveg, elektronika, képernyőkép látható  Automatikusan generált leírás |
| Bejelentkezési felület. Itt a regisztrációs adatokkal tudunk belépni amit megadtunk az imént. A „register\_here” gombra kattintva ugrunk át a registrációs oldara. | Bejelentkező oldal adatokkal való feltöltése. |
| A képen szöveg, monitor, elektronika, képernyőkép látható  Automatikusan generált leírás  Főmenü. Bejelentkezés után a főmenübe találjuk magunkat, ahol két feltöltés között lehet választani(egyenlőre). „Add credit card” a bankártya adatokat tölti fel. „Add personal card” pedig a személyigazolvány adatokat tölti fel. | |
|  |  |
| Bankkártya feltöltés oldal. Itt tudjuk megadni a szükséges adatokat a bankkártya feltöltéshez.  „Back to main” gombbal pedig vissza tudunk ugrani a főmenübe. | Személyigazolvány feltöltés oldal. Itt tudjuk megadni a szükséges adatokat a személyigazolvány feltöltéshez.  „Back to main” gombbal pedig vissza tudunk ugrani a főmenübe. |
| Bankkártyára példa: | Személyigazolványra példa: |
| Hibás adat esetén vagy kihagyott mező esetén szól a szoftver, hogy rosszul/üresen hagyta a mezőt a felhasználó. | Hibás adat esetén vagy kihagyott mező esetén szól a szoftver, hogy rosszul/üresen hagyta a mezőt a felhasználó. |

Szoftver néhány eligazító kódja

A változó nevek eltárolása a TextInputEditText és a TextInputLayout-ba vannak eltárolva a következő képpen

Változó nevek tárolása bankkártya adatokhoz és validáciohoz:

//Database layout  
TextInputEditText textInputLayoutcardNumber, textInputLayoutDate, textInputLayoutSecurityCode, textInputLayoutFirstName, textInputLayoutLastName;  
//Validation text input  
TextInputLayout textInputSecurityCode, textInputCardNumber, textInputExpirationDate, textInputFirstName, textInputLastName;

Változó nevek tárolása személyigazolvány adatokhoz és validáciohoz:

//Database layout  
TextInputEditText textInputLayoutdocumentId, textInputLayoutfullname, textInputLayoutGender, textInputLayoutYear, textInputLayoutMonth, textInputLayoutDay;  
//Validation text input  
TextInputLayout textInputfullname, textInputDocumentId, textInputGender, textInputYear, textInputMonth, textInputDay;

A bejelentkezés változónevek tárolása:

TextInputEditText textInputEditTextUsername,textInputEditTextPassword;

A regisztráció változónevek tárolása:

TextInputEditText textInputEditTextFullname, textInputEditTextUsername,textInputEditTextPassword,textInputEditTextEmail;

A validáció függvénye, vizsgálja ha üresen marad a kitöltendő mező:

private Boolean validationX() {  
 String val = textInputX.getEditText().getText().toString();  
 if (val.isEmpty()) {  
 textInputX.setError("Missing documentum id");  
 return false;  
 } else {  
 textInputX.setError(null);  
 return true;  
 }  
}

A kitöltendő mező, itt megadjuk az adatot a bejelentkezés/regisztráció/bankkártya/személyigazolvány oldalon.

<com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/til\_personal\_card\_X"  
 style="@style/XInputLayoutBoxes"  
 android:layout\_marginEnd="@dimen/inner\_vertical\_input\_margin"  
 android:hint="@string/X"  
 app:layout\_constraintEnd\_toStartOf="@id/til\_X"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/til\_X">  
  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/et\_personal\_X"  
 style="@style/InputEditTexts"  
 android:fontFamily="@font/sinmediasans"  
 android:maxLength="@integer/maximum\_personal\_card\_document\_iD\_length"  
 android:textColorHint="#757575" />  
  
</com.google.android.material.textfield.TextInputLayout>

Sztálj formázás külön változóban:

<style name="XInputLayoutBoxes"  
 parent="Widget.MaterialComponents.TextInputLayout.OutlinedBox.Dense">  
 <item name="android:layout\_width">0dp</item>  
 <item name="android:layout\_height">wrap\_content</item>  
 <item name="errorEnabled">true</item>  
</style>

Név validációja, kis vagy nagybetű-e, változóban eltárolva:

<style name="FirstOrLastNameInput" parent="InputEditTexts">  
 <item name="android:digits">abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ</item>  
 <item name="android:inputType">textCapWords</item>  
</style>

Szín keverés a gombokra:

<gradient android:startColor="@color/purple\_200"  
 android:endColor="@color/purple\_500"  
 android:type="linear"/>  
  
<corners android:radius="15dp"/>

Háttérszín formázás

Első háttér szín:

<gradient  
 android:type="linear"  
 android:angle="315"  
 android:startColor="@color/colorGradientEnd"  
 android:centerColor="@color/colorGradientStart"  
 android:endColor="@color/colorGradientCenter"/>

Második háttér szín:

<gradient  
 android:type="linear"  
 android:angle="315"  
 android:startColor="@color/colorGradientStart"  
 android:centerColor="@color/colorGradientCenter"  
 android:endColor="@color/colorGradientEnd"/>

A bankkártya és a szemlyigazolvány megjelenítése:

<include  
 android:id="@+id/input\_panel"  
 layout="@layout/input"  
 android:layout\_width="365dp"  
 android:layout\_height="331dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/image"  
 app:layout\_constraintWidth\_percent="@dimen/width\_percentage" />

Adatbázis és java kód összekapcsolása feltöltéshez(külső kód használata adat feltöltéshez)

**Adatbázis konfiguráció php-ban:**

public $servername;

public $username;

public $password;

public $databasename;

public function \_\_construct(){

$this->servername = 'localhost';

$this->username = 'root';

$this->password = '';

$this->databasename = 'onlinewallet';

}

**Adatbázis adat összekapcsolás php-ban:**

function dbConnect()

{

$this->connect = mysqli\_connect($this->servername, $this->username, $this->password, $this->databasename);

return $this->connect;

}

function prepareData($data)

{

return mysqli\_real\_escape\_string($this->connect, stripslashes(htmlspecialchars($data)));

}

**Adat feltöltési kódja:**

function X($table, $A, $B, $C)

{

$A = $this->prepareData($A);

$B = $this->prepareData($B);

$C = $this->prepareData($C);

$this->sql =

"INSERT INTO " . $table . " (a, b, c) VALUES ('" . $a . "','" . $b . "','" . $c . "')";

if (mysqli\_query($this->connect, $this->sql)) {

return true;

} else return false;

}

Adatbázis táblához való csatlakozás:

require "DataBase.php";

$db = new DataBase();

if (isset($\_POST['a']) && isset($\_POST[b]) && isset($\_POST['c']) && isset($\_POST['day'])) {

if ($db->dbConnect()) {

if ($db->feltoltes("a", $\_POST['b'], $\_POST['c']) {

echo "Success";

} else echo "Failed";

} else echo "Error: Database connection";

} else echo "All fields are required";

**Tapasztalatok és együttmüködés:**

A csapat munka mindig is egy válaszpont az informatika szempontjából is. Csapatban mindig jobb megoldani a problémákat. Nálunk a csapatmunka nem mindig volt teljes vagy zökkenőmentes. Voltak kihívások, amiket nem mindig tudtunk átugrani ketten. Volt olyan része is a munkánknak amit összhangban meg tudtunk valósítani. A tapasztalat pedig az, hogyha többet lettünk volna csapatban lehet máshogy állunk a feladathoz ezáltal másabb is lett volna a munka morál. Összeségben nem voltunk egy rossz csapat, de van hova fejlődni.