Dokumentáció

# Mini közösségi app

## **Készítette: Kiss Bence Máté és Rasch Vincent Richard**

## A program célja

## A Python program célja egy közösségi alkalmazás létrehozása Tkinter segítségével, amely lehetővé teszi, hogy alapvető információkat megtudjunk a felhasználókról akár képek formájában, valamint támogassa az üzenetküldést, lehetővé téve a felhasználók közötti interakciót. Továbbá tartalmazzon egy falat, ahol minden felhasználó szabadon hozzáteheti saját tartalmát.

## Megvalósított technológia

## A közösségi alkalmazásunkat Python környezetben a Tkinter csomag segítségével hoztuk létre, amely lehetőséget biztosít grafikus felhasználói felületű alkalmazások fejlesztésére. A megvalósítás során több fő widget használatával (PhotoImage, Button, Label, Entry, Text, Scrollbar és Frame) érjük el a kívánt eredményt.

## Felhasználói interakciók

## A program lehetővé teszi a felhasználók bejelentkezését, az alapvető információk megjelenítését, valamint képgalériák megtekintését. Az üzenetküldési funkcióval a csapattagok aszinkron módon kommunikálhatnak, üzeneteket küldhetnek és olvashatnak. Végül, a fal funkció lehetővé teszi, hogy minden felhasználó szabadon hozzátehessen tartalmakat.

## Kód

A kód egy fájlból áll, maincode.py-ból. Mivel Vincent és Máté egy héttel előbb kész volt az előző projekttel, így nem kellett foglalkoznunk már az azonosító, galéria, chat ablakkal(a tartalmukkal igen, de az ablakokkal nem).



A maincode.py az egész ablak forráskódját tartalmazza.

## Rasch Vincent Richard része

Először létrehoztam a bejelentkező ablakot, beállítottam a méretét fix(nem változtatható) méretre, címét, színét illetve a betűtípust. A felhasználói adatokat elmentése eredetileg külön-külön változókba került elmentésre, majd egy dictionary-be. A jelszó helyett \*-ot jelenítettem meg a biztonság miatt. A login() függvényben rávizsgálok, hogy helyes adatokat adtak-e meg. Az openuser() függvényekben beállítom a felhasználói ablakok tulajdonságait. Elhelyezem a bal oldali Frame-ben a felhasználó nevét, galériát, info gombot, a magán üzenethez hozzáférő gombot, falba való üzenet küldést, illetve a fal frissítésre egy gombot. Eredetileg csak kettő felhasználó volt (Bence, Rasch), de majd tesztelés céljából egy harmadikat is létrehoztam(Test).

Fő ablak illetve a felhasználók adatait tároló dictionary létrehozása.

Ein Bild, das Text, Screenshot enthält.

Automatisch generierte Beschreibung

A login() függvény.

Ein Bild, das Text, Screenshot, Software, Display enthält.

Automatisch generierte Beschreibung

Emellett még én vállaltam be a privát üzenetek küldésének lebonyolítását

Ein Bild, das Text, Screenshot enthält.

Automatisch generierte Beschreibung

## Felül látható az open\_private\_message(sender) függvény, amelynek elsődleges feladata megnyitni a privát üzenetek ablakot. Mint látható van paraméter (sender): az üzenetet küldő személy, emiatt nem kellet minden egyes felhasználónak külön függvényt létrehozni. A függvényen belül található még kettő függvény, a load\_messages() függvény (amely betölti a már írt üzeneteket a kiválasztott személlyel) és a send\_message() függvény amely segítségével el tudjuk küldeni az üzenetünket.

Részek amelyekről szerintem érdemes külön beszélni (mert szenvedtem rajta), az elején akadt egy olyan kis gondunk, hogy ha x akart küldeni üzenetet y-nak akkor létrehozott egy x\_y.txt fájlt, viszont ha y akart írni x-nek akkor létrehozott egy y\_x.txt nevű fájlt, ezt úgy oldottam meg, hogy bekértem az üzenetet fogadó személynek a nevét.



(a üzenetet küldő nevét külön nem kell bekérni, mert az open\_private\_messages() függvényben meg van adva paraméterként) és ezeket a sorted() segítségével abc sorrendbe rakom, ez alapján hozom létre a fájl nevét.Ein Bild, das Screenshot, Text, Schrift, Reihe enthält.

Automatisch generierte Beschreibung

Ez megakadályozza az előbb említett problémát hisz, ha x küld y-nak akkor is x\_y.txt és fordítva is.

Emelett külön még felhívnám a figyelmet erre a sorra. 

Itt a .trace segítségével, mindig ha megváltozik a recipient\_var változó akkor lefuttatja a load\_messages() függvényt, tehát ha x épp ír y-nak akkor a xy szöveg látható és ha x átvvált z-re akkor ugye megváltozik a recipient\_var és emiatt újratölti a szöveget és már a xz szöveg lesz látható.

## Kiss Bence Máté része

Elsődleges feladatom két függvény kibővítése volt: az openuser1() és az openuser2(),tehát a fal létrehozása. Elhelyeztem egy ChatSend gombot, egy Entry mezőt az üzeneteknek a rightsidechat Frame-en belül. Létrehoztam egy Text szövegmezőt, ahol megjelennek a fal tartalmai. Ennek a sima szerkesztését letiltottam, illetve ha beírtunk egy üzenetet, akkor automatikusan törlődik a mezőből. Ugyanezen a Frame-en belül létrehozom a Scrollbar funkciót, hogy a korábbi üzeneteket vissza lehessen olvasni. A tartalmak pozíciója a személytől függ. Ha az éppen bejelentkezett személy írta az üzenetet akkor az jobbra kerül, ha más személy akkor balra. Margókkal oldottam meg, hogy ne lógjanak be semmibe az üzenetek. Az üzeneteket egy messages.txt fájlba mentem fájlkezeléssel, a korábbiakat is így olvasom be. Beállítottam hogy az üzenetek mindig úgy jelenjenek meg, hogy amikor a felhasználó belép, rögtön a legújabb üzeneteket lássa.



A képen szöveg, képernyőkép, szoftver, Operációs rendszer látható

Automatikusan generált leírás

Az én feladatom volt még a galériák létrehozása. Ehhez segítségnek a Classroomba feltöltött példát használtam. A galery() és a galery2() függvényben a képeket PhotoImage segítségével olvasom be és ezeket eltárolom egy listában. Egy Label segítségével megjelenítem őket, alatta helyeztem el egy státuszt jelző sávot, a kép mellé pedig léptető nyilakat. Egy kilépés gombbal az ablak bezárható.

A képen szöveg, képernyőkép, szoftver látható

Automatikusan generált leírás

Az alapvető információk ablakot rólunk egy openinfo() és egy openinfo2() függvényben hozom létre. Ide néhány adat rólunk bekerült pár Label segítségével. Illetve a dokumentáció egy nagyobb része is az én feladatkörömbe kapcsolódik.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

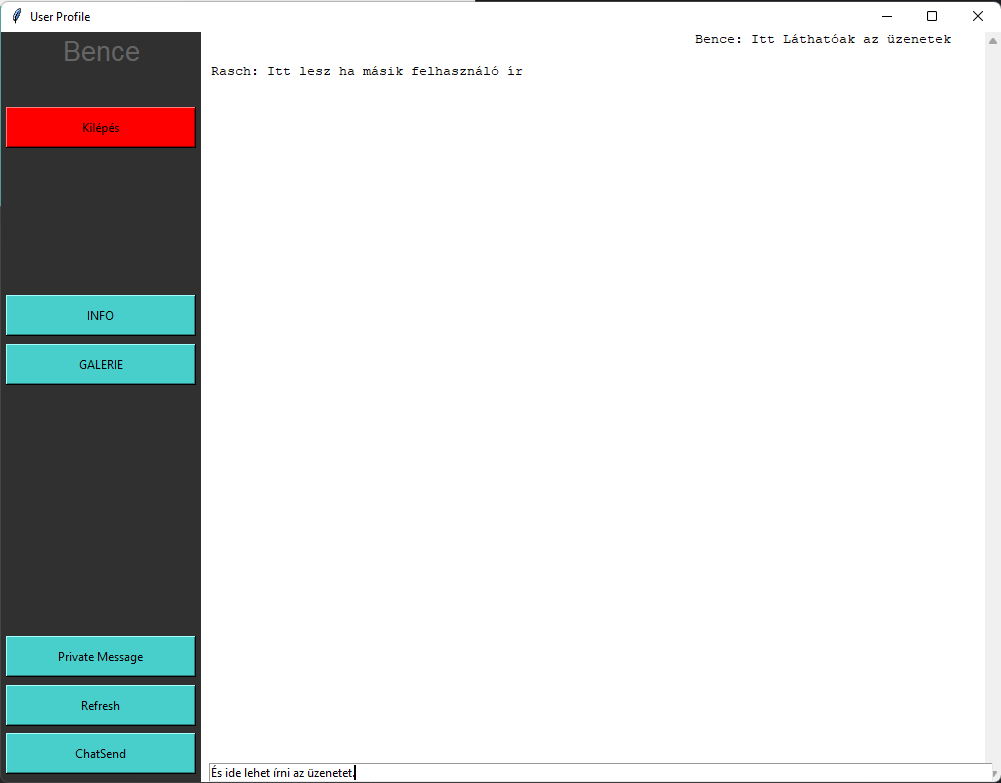
## Cél

A feladatot azt gondoljuk teljes mértékben sikerült teljesítenünk, elértük a kitűzött célt.

## Továbbfejlesztési lehetőségek

Esetleg némi design változtatások, illetve az új felhasználók érdekéből egy regisztrálás funkciót és ezeknek bejelentkezését illetve saját ablak megnyitását hatékonyabban megoldani (Most jelenleg minden egyes felhasználónak van egy saját függvénye, ezt rövidebben is meg lehetne oldani.). A felhasználók profilképet állíthatnának be, ez még személyesebbé tenné az alkalmazást. Az aszinkron üzenetküldés továbbfejlesztésével valós idejű üzenetküldő rendszer is létrehozható, amely lehetővé tenné, hogy a felhasználók azonnal értesüljenek az új üzenetekről, akár értesítés formájában is.

# A képen szöveg, képernyőkép, Betűtípus, sor látható Automatikusan generált leírásIllusztráció



A képen képernyőkép, szöveg, Színesség, Multimédiás szoftver látható

Automatikusan generált leírás

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Automatikusan generált leírás

A képen szöveg, képernyőkép, szoftver, tervezés látható

Automatikusan generált leírás