UNIVERSITATEA „ALEXANDRU IOAN CUZA” IAŞI

**FACULTATEA DE INFORMATICĂ**

****

LUCRARE DE LICENŢĂ

**Reader Water Meter**

propusă de

***Matieș Petruța***

**Sesiunea:** *iulie 2020*

Coordonator ştiinţific

**Drd. Florin Olariu**

UNIVERSITATEA „ALEXANDRU IOAN CUZA” IAŞI

**FACULTATEA DE INFORMATICĂ**

LUCRARE DE LICENŢĂ

**Reader Water Meter**

propusă de

***Matieș Petruța***

**Sesiunea:** *iulie 2020*

Coordonator ştiinţific

**Drd. Florin Olariu**

Titlul, Numele și prenumele

Data \_\_\_\_\_\_\_\_\_\_\_\_

Avizat,

Îndrumător Lucrare de Licență

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Semnătura \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**DECLARAȚIE privind originalitatea conținutului lucrării de licență**

Subsemnatul(a) ………………………………………………………………………………………

domiciliul în …………………………………………………………………………………………………..

născut(ă) la data de ………………..…., identificat prin CNP ………….……………..………………..., absolvent(a) al(a) Universității „Alexandru Ioan Cuza” din Iași, Facultatea de ………………………. specializarea

…………………………………………………………, promoția …………………………., declar pe propria răspundere, cunoscând

consecințele falsului în declarații în sensul art. 326 din Noul Cod Penal și dispozițiile Legii Educației Naționale nr. 1/2011 art.143 al. 4 si 5 referitoare la plagiat, că lucrarea de licență cu titlul:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_elaborat

* sub îndrumarea dl. / d-na \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, pe care urmează să o susțină în fața comisiei este originală, îmi aparține și îmi asum conținutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licență să fie verificată prin orice modalitate legală pentru confirmarea originalității, consimțind inclusiv la introducerea conținutului său într-o bază de date în acest scop.

Am luat la cunoștință despre faptul că este interzisă comercializarea de lucrări științifice in vederea facilitării fasificării de către cumpărător a calității de autor al unei lucrări de licență, de diploma sau de disertație și în acest sens, declar pe proprie răspundere că lucrarea de față nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Dată azi, ………………………… Semnătură student …………………………

Anexa 6

ACORD PRIVIND PROPRIETATEA DREPTULUI DE AUTOR

Facultatea de Informatică este de acord ca drepturile de autor asupra programelor-calculator, în format executabil și sursă, să aparțină autorului prezentei lucrări, *Matieș Petruța.*

Încheierea acestui acord este necesară din următoarele motive:

Iași, *data*

Decan *Prenume Nume* Absolvent *Prenume Nume*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(semnătura în original) (semnătura în original)

Rezumat

În această lucrare se descrie funcționalitatea aplicației denumită Reader Water Meter împreună cu toate componentele ce o alcătuiesc. Reader Water Meter este o aplicație IOS mobilă construită cu ajutorul limbajului swift, venită în ajutorul utilizatorului pentru a ușura acestuia modalitatea prin care poate trimite indexul la apă. Pentru a putea folosi aplicația este necesar crearea unui cont care conține numele, adresa de email și locul de consum al utilizatorului. Cu ajutorul contului se poate încărca o poză cu indexul apometrului pe baza caruia se generează un istoric al consumului.

Cuvinte cheie

Aplicație IOS mobilă, swift, Firebase, MLVision, charts, text recognize

Cuprins

[Introducere si motivație 8](#_Toc43628901)

[Contribuții 9](#_Toc43628902)

[Capitolul I – Descrierea problemei 9](#_Toc43628903)

[Capitolul II – Abordări anterioare 10](#_Toc43628904)

[II.1 Meter Reader by TotalSync, Inc. 10](#_Toc43628905)

[II.2 Concluzii 10](#_Toc43628906)

[Capitolul III - Descrierea soluției 11](#_Toc43628907)

[III.1Tehnologii folosite 11](#_Toc43628908)

[III.1.1 Swift 11](#_Toc43628909)

[III.1.2 Autentificarea cu Google 11](#_Toc43628910)

[III.1.3 Charts 11](#_Toc43628911)

[III.1.4 Notificări 11](#_Toc43628912)

[III.1.5 Recunoașterea textului din imagine 12](#_Toc43628913)

[III.1.6 Încarcarea imaginii cu apometru 12](#_Toc43628914)

[III.1.7 Objective C 12](#_Toc43628915)

[III.1.8 Sistem de versionare 12](#_Toc43628916)

[III.1.8.1 BitBucket 14](#_Toc43628917)

[III.2-Arhitectura aplicației 14](#_Toc43628919)

[III.2.2 Arhitectura pe mai multe nivele (n-tier) 14](#_Toc43628920)

[III.2.2.1 Nivelul de Logică 15](#_Toc43628921)

[III.2.2.2 Nivelul de Date 16](#_Toc43628922)

[III.2.2.3 Nivelul de Prezentare 17](#_Toc43628923)

[III.3-Prezentarea generala a aplicației 17](#_Toc43628924)

[III.3.1 Desfășurarea aplicației 18](#_Toc43628925)

[III.3.2 Înregistrarea și logarea în aplicație 18](#_Toc43628926)

[III.3.2.1 Logarea folosind contul de Google 18](#_Toc43628927)

[III.3.3 Paginile principale ale aplicației 19](#_Toc43628928)

[III.3.3.1 Pagina principală 19](#_Toc43628929)

[III.3.3.2 Meniul 21](#_Toc43628930)

[III.3.3.2.1 Istoric 21](#_Toc43628931)

[III.3.3.2.2 Pagina de setări 22](#_Toc43628932)

[III.3.3.2.3 Pagina despre aplicație 23](#_Toc43628933)

[III.3.3.3 Profilul utilizatorului 23](#_Toc43628934)

[III.3.3.4 Notificare 24](#_Toc43628935)

[Concluzia lucrării 25](#_Toc43628936)

[Bibliografie 26](#_Toc43628937)

[Anexa 1 27](#_Toc43628938)

# Introducere si motivație

Punctul de plecare al acestei lucrări de licența este reprezentat de către evoluția fără precedent a tehnologiei și inserarea resurselor acesteia în toate sectoarele de activitate ale societății.

Nevoia unei astfel de implementări este subliniată, pe de o parte, de transformările din domeniul tehnologiei și, pe de altă parte, de importanța accentuată pe care resursele tehnologiilor digitale o au în viața nativilor digitali.

Prin această lucrare se încearcă rezolvarea problemelor legate trimiterea indexului la apă.

În ultima perioadă suntem tot mai presați de trecerea timpului, care este foarte prețios și nu ar trebui să ne preocupăm prea mult de lucrurile mai puțin importante, cum ar fi trimiterea indexului la apă. Această aplicație ne scapă de timpul pierdut la coadă pentru a da indexul, pe lânga acest lucru nu mai este nevoie să ne sincronizăm cu programul de la asociație.

Motivația ce a stat la baza acestei lucrări de licența este formată din doua arii principale:

* Aria non-tehnica
* Timpul pierdut pe drum până la asociație și în unele cazuri timpul petrecut la cozile de la asociație.
* Dorința de a nu mai renunța la lucrurile importante pentru a ne putea sincroniza cu programul de la asociație.
  1. Dorința de a avea o modalitate simplificată în procesul de trimitere a indexului.
* Aria tehnică
  1. Dorința de a dovedi că se poate construi o aplicație cu ajutorul căreia se poate trimite indexul la apă într-un timp mult mai scurt.

o Pasiunea pentru programarea în Swift.

# Contribuții

În urma cercetării soluțiilor propuse de către statul român cu privire la rezolvarea problemelor trimiterii indexului la apă, s-a simțit nevoia implementării unui alt sistem deoarece cele existente la ora actuală, nu rezolvă în totalitate problemele apărute. Datorită programului încarcat pe care majoritatea persoanelor îl au, în ultimul timp s-a văzut o neputință în ajutarea tuturor persoanelor de a trimite îndexul la apă fără a primi estimări.

Principala contribuție în aceasta lucrare de licență a fost găsirea și dezvoltarea unei soluții ce permite trimiterea indexului la apă de la distanță pentru a rezolva și problema deplasării până la asociație. Distanța fiind unul dintre cele mai mari impedimente pentru cei ce lucrează.

O altă contribuție ce trebuie menționată este realizarea aplicației folosind limbaj de actualitate precum Swift pentru crearea aplicațiilor iOS mobile și o integrare ușoară între cele două componente de Front-End și Back-End, aceste lucruri asigurând și o performanță la nivel de aplicație.

Modul în care a fost gândită aplicația trebuie văzut ca o contribuție deoarece prin funcționalitatea sa, aceasta propune unele măsuri prin care se evită uitarea trimiterii indexului și pierderea timpului până la asociație. O modalitate prin care se asigură trimiterea corectă a indexului este încarcarea imaginii cu apometru din care este recunoscut indexul, astfel sunt eliminate greșelile umane. O altă modalitate este interzicerea utilizatorului de a trimite un index mai mic decât cel trimis cu o lună in urmă și restricționarea acestuia de a trimite indexul de mai multe ori in aceiași luna.

# Capitolul I – Descrierea problemei

Transmiterea indexului la apă este obligatoriu pentru ca fiecare persoană să platească o sumă de bani direct proportional cu consumul de apă din fiecare lună.

Punctul de plecare al acestei aplicații este reprezentat de dorința de a îmbunătăți condițiile de trimitere a indexului la apă a cetățenilor români. În ultimul timp suntem tot mai ocupați și datorită acestui fapt, nu toți mai apucă să ajungă la asociație, în acest caz este pus un index estimativ care de multe ori este mai mare decât indexul curent.

O soluție pentru această problemă este crearea unei aplicații online, ce trebuie sa substituie procedura de transmitere a indexului la apă clasică, astfel evitând deplasarea pană la asociație și statul la coadă.

# Capitolul II – Abordări anterioare

## II.1 Meter Reader by TotalSync, Inc.

Este o aplicație care vă permite să vă monitorizați continuu consumul de apă din cursul lunii. Pentru a utiliza această aplicație nu este nevoie de crearea unui cont. Utilizatorul în pagina de setari trebuie să completeze o sumă care va fi implicită in cazul în care nu este trimis indexul, prețul la apă in euro. Această aplicație mai oferă și sugestii de verificare a consumului de apă, unele dintre ele sunt: verificarea vasului de toaletă sa nu aiba scurgeri, a boilerului sau a robinetelor.

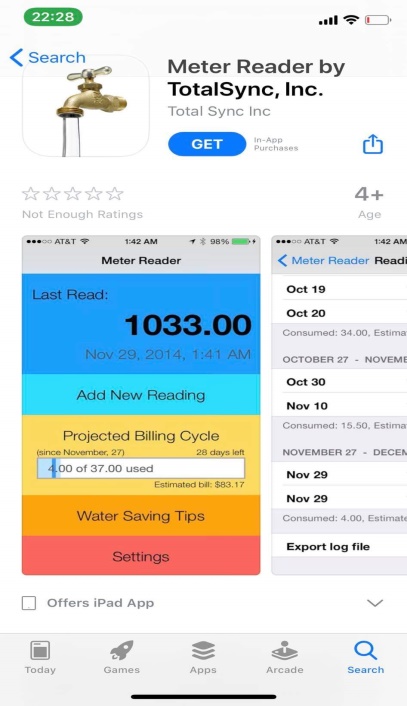


Fig. 1

Meter Reader by TotalSync, Inc

## II.2 Concluzii

În concluzie, putem vedea că aplicația de mai sus nu acoperă problema noastră și acestă soluție nu se regăsește în România la nivel practic. O astfel de aplicație trebuie să ofere posibilitatea trimiterii indexului la apă utilizând un cont. Ca urmare a acestui fapt, se simte nevoia unei soluții în problema trimiterii indexului la apă și o aplicație nouă ar fi bine-venit.

# Capitolul III - Descrierea soluției

## III.1Tehnologii folosite

Acest capitol este creat cu scopul de a prezenta tehnologiile folosite în crearea acestui proiect. Scopul aplicației este de a stoca indexul la apă a utilizatorului și de a păstra istoricul tuturor încarcărilor. Pe de altă parte, aplicația poate fi văzută și ca un proiect de cercetare pentru a vedea dacă sistemul de trimitere a indexului din Romania poate fi îmbunătățit prin folosirea acestei aplicații iOS mobile.

## III.1.1 Swift

Este o tehnologie nativă care are avantajul că utilizează de programare oficiale Google sau Apple prin care se poate accesa direct orice librărie/API și funcție nativă a sistemului de operare; de asemenea viteza de execuție și dimensiunea aplicației sunt optime.

Pentru realizarea aplicației am folosit diverse librarii din Swift:

## III.1.2 Autentificarea cu Google

Am folosit această librarie pentru a veni în ajutor utilizatorului deoarece este o alternative mai ușoară de creare a unui cont. Din Firebase am configurat Google ca metodă de sign-in. Este o metodă sigură de creare a unui cont deoarece Firebase face validarile de email respectiv parolă.

## III.1.3 Charts

Am folosit libraria Charts pentru a genera istoricul utilizatorului într-un bar chart care cuprinde valoarea facturii ce va fi eliberată și luna aferentă acesteia. Valoarea facturii reprezintă (ultimul index – penultimul index)\*prețul la apă. Luna este extrasă din data stocată o dată cu indexul trimis in baza de date.

## III.1.4 Notificări

UserNotifications este librăria pe care am folosit-o pentru serviciul de notificări pentru utilizatori.

Am creat conținutul notificarii care apare la începutul perioadei de trimitere a indexului. Această funcționalitate ajută utilizatorul să trimită in mod constant indexul la apă.

## III.1.5 Recunoașterea textului din imagine

MLVision este o librarie din Firebase care conține instant VisonTextRecognizer, ca ajutorul ei am extras textul din imaginea cu apometru.

Această funcționalitate este utilizată pentru recunoașterea indexului la apă din imaginii încărcate de un utilizator, identificând caracterele prezente într-o imagine și pune aceste caractere în cuvinte, permițând să se caute apoi și să se editeze conținutul rezultat.

Pentru a configura aplicația cu MLVision s-au urmat pașii următori:

1. Am adaugat pod Firebase/MLVision și am actualizat podul
2. Am importat Firebase in project
3. Am luat imagine incarcată de utilizator cu ajutorul librariei UIImagePickerController
4. Cu ajutorul funcției onDiviceTextRecognizer() am extras blocurile de text
5. Am extras indexul

## III.1.6 Încarcarea imaginii cu apometru

Pentru a trimite indexul la apă utilizatorul are posibilitatea să încarce o poză cu apometrul din care este extras indexul. Acestă funcționalitate a fost creată cu ajutorul librariei UIKit care conține UIImagePickerControllers, care permite utilizatorului să acceseze fotografiile din telefon și să selecteze una.

## III.1.7 Objective C

Am utilizat această tehnologie pentru a oferi utilizatorului posibilitatea sa navigeze de pe pagina principală pe meniu prin a face swipe. Pentru o aplicație mobilă este mult mai ușor de folosit această funcționalitate.

## III.1.8 Sistem de versionare

Existând întotdeauna șansele unor probleme tehnice, pentru a nu pierde toata munca depusă la aplicație s-a ajuns la concluzia că ar fi bine să se stocheze codul sursă într-un sistem de versionare. Acest sistem permite gestionarea versiunilor multiple a unor fișiere, precum și lucratul în paralel asupra unor fișiere. Sistemele de versionare au fost concepute pentru a permite membrilor mai multor echipe să opereze modificări pe același proiect, aceste modificări urmând a fi reunite într-o noua versiune a proiectului.

Avantajele folosirii unui sistem de versionare:

1. Fiind salvat istoricul modificărilor, se poate reveni cu ușurința la o variantă de proiect mai veche dacă se descoperă introducerea unor defecte în ultima versiune.
2. Utilizându-se un serviciu de hosting, codul sursă va avea mereu o copie de siguranță online
3. Dezvoltatorii proiectului vor avea mereu acces la ultima versiune a codului, făcând astfel colaborarea si sincronizarea mult mai ușoară decât în cazul trimiterii de fișiere ce conțin cod sursa.

## III.1.8.1 BitBucket

Bitbucket este un serviciu de găzduire a unui depozit de control al versiunilor web, deținut de Atlassian, pentru proiecte de cod sursă și dezvoltare care utilizează sisteme de control de revizuire Mercurial sau Git. Bitbucket oferă atât planuri comerciale, cât și conturi gratuite.

Am folosit inițial acest serviciu deoarece GitHub nu permitea incarcarea întregului proiect deoarece aveao o dimensiunea prea mare.

## III.1.8.2 GitHub

Github este un serviciu de hosting al proiectelor git (un proiect git mai este numit și repository). Acesta permite păstrarea unei copii a proiectului onlineși vizualizarea tuturor fișierelor și a modificărilor aduse acestora.

Mai multe informatii în legatură cu crearea unui nou repository se găsesc în Anexa 1.

## III.2-Arhitectura aplicației

O arhitectură reprezintă setul de decizii semnificative privind organizarea unui sistem software, selectarea elementelor structurale și a interfețelor acestora prin care este compus sistemul, împreună cu comportamentul specificat în colaborările dintre elementele respective. Aceasta va putea fi aleasă după ce vor fi studiați mai mulți factori precum posibilitatea de modificare a codului, performanța cu scopul de a îmbunătăți funcționalitatea aplicației.

## III.2.2 Arhitectura pe mai multe nivele (n-tier)

În aplicație s-a folosit o arhitectură pe trei nivele, aceasta fiind probabil una dintre cele mai cunoscute arhitecturi (n-tier). S-a recurs la acest tip de arhitectură deoarece funcționalitatea este construită în jurul unei baze de date și din dorința de a separa aplicația în mai multe nivele( nivelul de prezentare, nivelul de logica și nivelul de date) cu scopul de a gestiona ușor codul dintr-un nivel fără a afecta și celelalte nivele ca în Fig. 12.

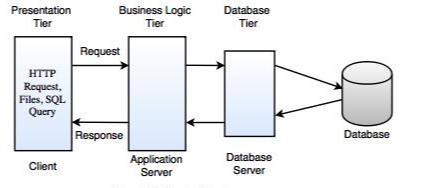


Fig. 12 Arhitectura pe 3 nivele Figura 12 este preluată și are ca sursă următorul link[2]

## III.2.2.1 Nivelul de Logică

Nivelul de logică este scris în swift și objective c și conține logica ce formează funcționalitățile de baza ale aplicației. În aplicația Reader Water Meter acest nivel este asigurat de către multe clase controller. Aceste controllere au rolul de a gestiona cererile browserului și în funcție de acestea va furniza datele dorite de către utilizator.

Un exemplu din aplicație ar fi LoginVC ce gestionează partea de logare în aplicație a unui utilizator. Când un request de login este trimis de la user către aplicație, în metoda din controller se verifică autenticitatea datelor iar dacă totul este în regulă, se permite userului sa intre în aplicație.

Un alt exemplu ar fi ProfileVC care în urma apasării pe numele utilizatorului din pagina de meniu se populează pop up-ul cu datele acestuia, ca de exemplu: numele, adresa de email, locul de consum, și pictograma cu background-ul pe care și la setat la crearea contului.

## III.2.2.2 Nivelul de Date

Acest nivel de arhitectura are rolul de a gestiona baza de date și de a furniza datele din aceasta. Cuprinde baza de date/sistemul de stocare a datelor și stratul de acces la date. Exemple de altfel de sisteme sunt MySql, Oracle, PostgreSQL, Microsoft SQL Server, MongoDB, etc. Acest nivel este compus la rândul lui din domeniul de date și persistenta datelor. Pentru acestest nivel s-a folosit Firebase. Am ales să folosesc Firebase deoarece la înregistrarea utiluzatorului nu mai este nevoie de validări separate doarece sunt facute de Firebase, cum ar fi adresa de email și parola. În cazul în care acestea nu sunt corecte apare un mesaj de eroare. Un alt motiv pentru care am ales această bază de date este posibilitatea de logare cu contul de Google și numarul mare de librarii pe care le pune la dispoziție, cum ar fi pentru recunoșterea textului, charts.

**Domeniul de Date**

Acest domeniu este compus din toate modelele ce sunt folosite pentru a genera baza de date

1. aplicației. Fiecare model are propriile sale proprietăți ce vor corespunde proprietăților unei tabele din baza de date. Câteva exemple de proprietăți pentru modelul Profile din aplicație ar fi numele, adresa de email, locul de consum și pictograma utilizatorului ca în Fig. 2.

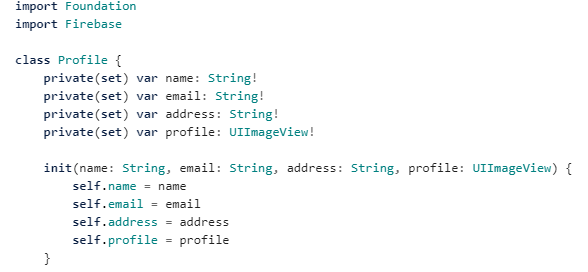
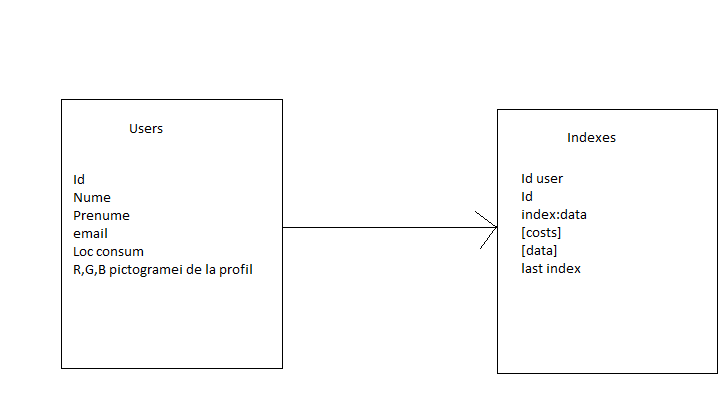


Fig. 2

Proprietăți din model

**Structura bazei de date**

Pentru a înțelege cea mai importantă funcționalitate din aplicație, cea de trimitere a indexului la apă, trebuie să știm structura bazei de date. Pentru faza de date am folosit Firebase unde am creeat două colectii. Colecția users care conține numele, locul de consum, pretul la apă, gen, data la care a fost creeat contul, iar pentru utilizatorii care sunt inregistrați pe baza formularului se mai adaugă și codurile(r,g,b) culorilor. Aceste coduri sunt utilizate in generarea culorii de la pictograma profilului. Colecția indexes conține indexul pe care utilizatorul îl încarcă înainte de a trimite primul index, un vector de date în care sunt stocate lunile aferente consumului, un vector de costuri de tip numeric care reprezintă valoarea de plată a fiecarei luni și istoricul indecșilor încarcați, iar fiecărui index îi este asociat data încarcării. Aceste două colecții sunt legate prin id-ul utilizatorului.



## III.2.2.3 Nivelul de Prezentare

Nivelul de prezentare este stratul de front end din arhitectura pe trei nivele și constă în interfața cu utilizatorul. Această interfața este ceea ce vede utilizatorul și cu ce interacționează acesta. Aici intră informațiile introduse de utilizator, iar acest nivel se comportă ca o interfața între nivelul de date și utilizator, transmițând diferitele acțiuni ale utilizatorului către nivelul logic.

Acest nivel este creat cu ajutorul storyboard-ului.

## III.3-Prezentarea generala a aplicației

Aceasta aplicație este reprezentată de o parte de back-end construită în Swift și o parte de front-end construită în Storyboard. Scopul ei principal este cel de a oferi o modalitate cât mai simplă pentru a trimite indexul la apă online și de a evita deplasarea până la asociație.

## III.3.1 Desfășurarea aplicației

Reader Water Meter este o aplicație ce necesită logare utilizând una din cele două variante, prima este crearea unui cont folosind pagina de înregistrare și a doua cu contul de Google.

Atunci când un nou utilizator va dori să folosească aplicația, el va fi întâmpinat de prima pagină, unde i se va sugera să se înregistreze. În cazul în care acesta va avea deja creat un cont, se va loga în aplicație.

## III.3.2 Înregistrarea și logarea în aplicație

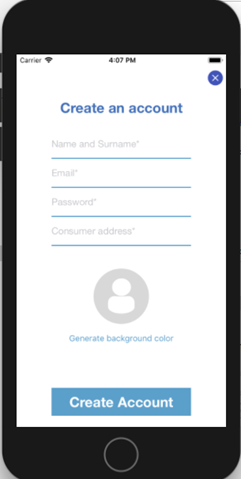


Fig. 10 Înregistrarea unui utilizator

Figura 10 reprezintă modalitatea de înregistrare pentru utilizatori. Principalul avantaj al utilizării aplicației este faptul că utilizatorul poate sa acceseze aplicația folosind contul de Google. Dacă dorește să creeze un cont folosind metoda clasică trebuie să introducă numele, adresa de email, parola, locul de consum și obțional dacă dorește să iși personalizeze contul poate alege o culoare. Înainte de creare contului sunt făcute validări pentru adresa de email și parolă.

## III.3.2.1 Logarea folosind contul de Google

## 

Fig. 11

Logarea

Fig. 11 reprezintă pagina de logare unde avem posibilitatea de a alege “Login with Google” pentru a sări peste pașii clasici de inregistrare. Aceasta funcționalitate este realizată cu ajutorul platformei Firebase.

## III.3.3 Paginile principale ale aplicației

După logare utilizatorul este direcționat pe pagina principă de unde poate naviga catre pagina de meniu, care cuprinde: istoricul indecșilor trimiși, setări, informații despre aplicație și profilul utilizatorului.

## III.3.3.1 Pagina principală

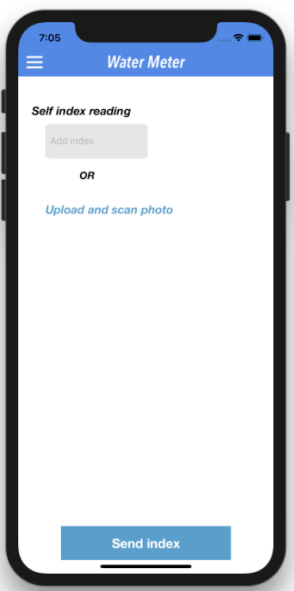


Fig. 12

Pagina principală a aplicație

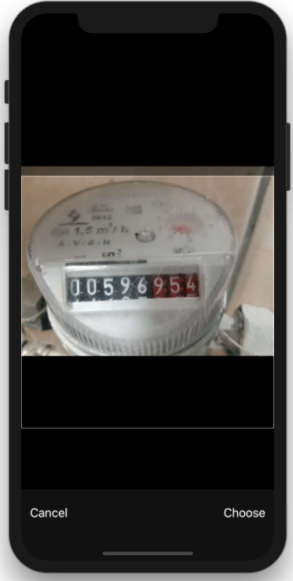


Fig. 13

Imagine cu apometru

Fig. 12 reprezintă pagina principală a aplicației unde este redirecționat utilizatorul după logare. Aceasta pagină este una din cele mai importante deoarece aici utilizatorul introduce indexul la apă și il trimite.

După creearea unui cont nou utilizatorul trebuie să pună în pagina de setari ultimul index, care reprezintă indexul cu o lună în urmă, altfel nu poate transmite indexul.

Indexul poate fi introdus manual sau se poate incarca o imagine cu apometrul. Pentru recunoașterea indexului din imagine am folosit libraria FirebaseMLVision, care recunoaște textul dint-o imagine.

Indexul apometrului are la final 3 cifre roșii care nu sunt luate in considerare la trimiterea indexului, ca in Fig. 13, de aceea când trimit indexul o sa am un număr intreg făra să țin cont de cifrele roșii. Folosind Fig. 13 ca exemplu, indexul este 596,954 și în baza de date va fi stocat 596.

După incarcarea imaginii este afișat pe pagina principală indexul în câmpul Add index, dar și imaginea. După apăsarea butonului Send index, indexul este stocat in baza de date împreună cu data la care acesta a trimis indexul și apare un pop up prin care utilizatorul este informat că indexul a fost trimis cu success ca in figura 14.

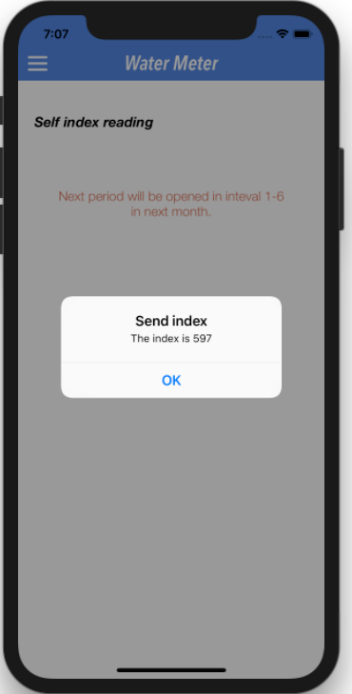
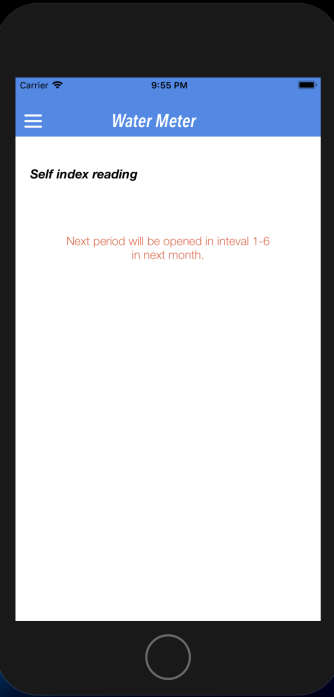
 

Fig. 15

Mesaj după trimiterea indexului

Fig. 14

Pop up trimitere index

După ce pop up-ul este închis pe pagina principală este afișat mesajul “Next period will be opened in interval 1-6 in next month” ca in Fig. 15, care limitează utilizatorul ca să trimită de mai multe ori in aceeași luna indexul.

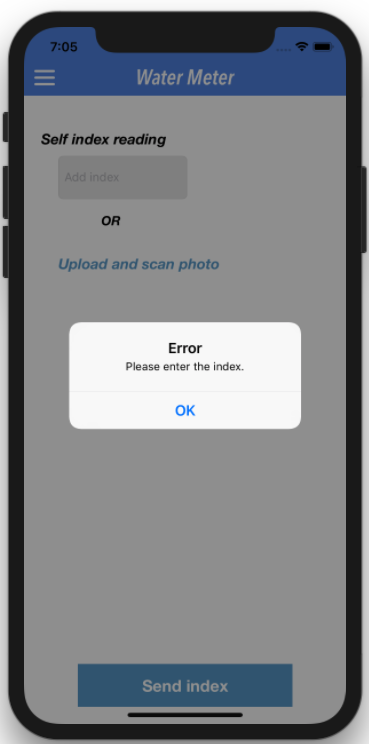


Fig. 16

Mesaj de eroare la trimiterea indexului

Dacă utilizatorul apasă butonul Send index, fără să introducă indexul, apare pop ul-ul cu mesaj de eroare care ii transmite utilizatorului că trebuie să introducă indexul la în Fig. 16.

În cazul în care utilizatorul vrea să trimită un index mai mic decât cel stocat în baza de date, apare un pop up de eroare care il informează ca indexul este greșit.

## III.3.3.2 Meniul

De pe pagina principal utilizatorul poate naviga în meniu care arată ca în Fig. 17, fiind o aplicație mobilă acest lucru pe poate face fie apasând pe meniu sau facând swipe la dreapta, ultima posibilitate a fost realizată cu ajutorul limbajului Objective C.

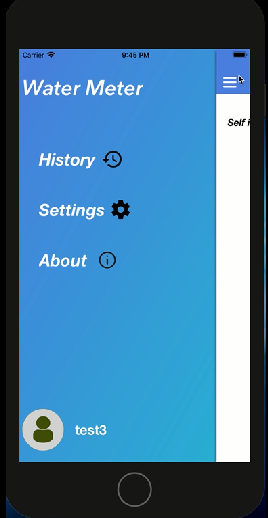


Fig. 17

Meniu

## III.3.3.2.1 Istoric

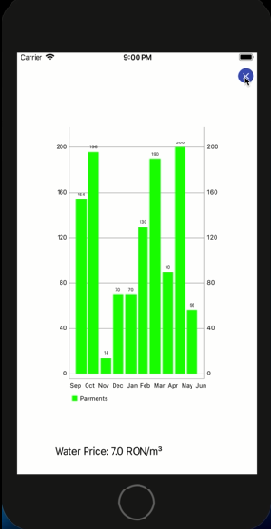


Fig. 18

Istoric

După ce a fost trimis indexul, utilizatorul poate vizualiza istoricul său. Acesta arată ca în Fig. 18 și este realizat cu ajutorul librariei Charts din Firebase. După cum am menționat mai sus când este trimis indexul se stochează in baza de date și data la care acesta a fost trimis. Pe baza acestor informații se face diferența dinte ultimul index trimis si cel current, această diferența se inmulțește cu prețul pe de apă. Această valoare este afișată impreună cu luna în care a fost trimis indexul.

Acest chart ne ajută să ținem o evidența a consumului de apă.

## III.3.3.2.2 Pagina de setări

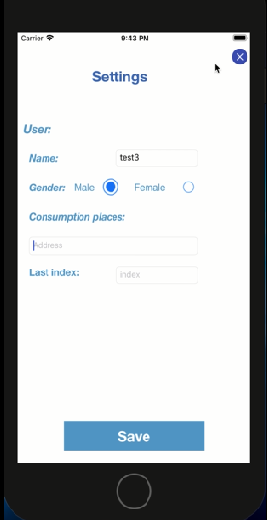


Fig. 19

Pagina de setări

În Fig. 19 este reprezentată pagina de setări unde utilizatorul după crearea contului trebuie să stocheze in baza de dateinformații despre gen(masculine, feminin), locul de consum și ultimul index. Cel din urmă, odată trimise toate informațiile, nu mai este vizibil deoarece utilizatorul o singura dată poate să trimită indexul in funcție de care se calculează prima valoare din istoric.

## III.3.3.2.3 Pagina despre aplicație

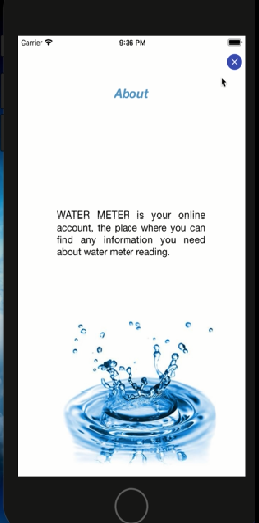


Fig. 20

Despre aplicație

În Fig. 20 se regăsesc informații despre aplicație.

## III.3.3.3 Profilul utilizatorului

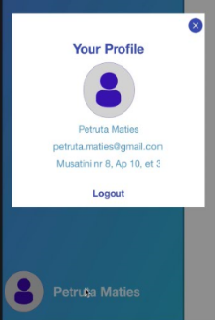


Fig. 21

Profiul utilizatorului

În Fig. 21 este prezentat profilul utilizatorului, care este un pop up și cuprinde numele, adresa de email și locul de consum pentru care se face trimiterea indexului. Pentru a modifica aceste informații este necesară navigarea în pagina de setari

## III.3.3.4 Notificare

Fig. 22

Notificare cu privire la inceperea perioadei de trimitere a indexului



În fiecare lună la începutul perioadei de transmitere a indexului, utilizatorul primește o notificare ca in Fig. 22. Această fincționalitate este benefică in cazul in care uităm să trimitem indexul

# Concluzia lucrării

Această lucrare de licență prezintă principalele motive pentru care odată cu dezvoltarea tehnologiei trebuie aduse îmbunătățiri și într-o activitate precum trimiterea indexului la apă, care la o primă vedere pare simplă. Dar în contradicție cu simplitatea acestei activități, stă importanța ei, deoarece în ultima perioadă suntem tot mai presați de timp și doar folosind tehnologia putem să evităm deplasarea până la asociație.

Contribuția personala în această aplicație constă în migrarea sistemului de transmitere a indexului la apă clasic în mediul online. Această migrare vine cu o serie de lucruri positive. Principalele beneficii care trebuie amintite ar fi trimiterea indexului in mod constant fără a fi nevoie de prea mult timp. Nu mai este nevoie să reținem în ce perioadă poate fi trimis indexul deoarece primim notificari de la aplicație. Un alt benefici ar fi vizibilitatea istoricului și calcularea imediată a costului.

Baza oferită de această aplicație poate fi ușor extinsă pentru a oferi o gamă mai mare de funcționalități, precum configurarea pe mai multe sisteme de operare. De asemenea, pe partea de UI există multe posibilități de îmbunătățire a experienței utilizatorului prin adăugarea de elemente interactive, cum ar fi adăugarea posibilității de a folosi camera pentru incarcarea indexului.

În concluzie, s-a dovedit că această procedură de trimitere a indexului la apă, se poate implementa și într-o aplicație din mediul online, putând să substituie cu succes clasica metodă.

# Bibliografie

[1] Aplicații asemanatoare

Meter Reader by TotalSync, Inc <https://apps.apple.com/us/app/meter-reader-by-totalsync-inc/id887038348>

1. Firebase <https://console.firebase.google.com/u/0/>
2. Documentație Git <https://teaching.alexcoman.com/resurse/tutorial/git/notiuni-introductive/>
3. Documentație Arhitectură pe mai multe nivele(n-tier) - <https://stackify.com/n-tier-architecture/>
4. Dezvoltarea aplicațiilor iOS mobile - <https://www.udemy.com/course/devslopes-ios12/learn/lecture/11450478?start=15#overview>
5. Bar Chart - <https://www.appcoda.com/ios-charts-api-tutorial/>
6. Text Recognize - <https://firebase.google.com/docs/ml-kit/ios/recognize-text>
7. Google sign-in <https://firebase.google.com/docs/auth/ios/google-signin>

# Anexa 1

Configurarea proiectului cu Firebase

Pentru a se putea folosi Firebase în aplicație, este necesară descarcarea unui plist care este adugat în structura proiectului. Se instalează pods file în care se adaugă firebase. Se crează baza de date care este structurată in colecții și documente. De la autentification din sign-in method se activează email pentru a permite utilizatorului să se logheze cu adresa de email.

Pentru a putea face toate acestea este nevoie de un crearea unui cont la link-ul: <https://console.firebase.google.com/>

Crearea unui repository nou

Pașii ce au fost urmați pentru a crea un nou repository:

1. Crearea unui cont de github la următorul link: <https://github.com/>
2. Crearea unui nou repository. Se poate face atât online de pe linkul anterior cât si de la linia de comandă, local. Dacă se optează pentru o variantă creată local va trebui în prealabil git-ul să fie instalat apoi, va trebui să se creeze un folder unde se va dori să se țina codul sursa, iar apoi, de la linia de comandă va trebui rulată comanda git init Nume-Repository.
3. Instalarea git de la următorul link: <https://git-scm.com/downloads>
4. După ce repository-ul s-a creat cu succes va trebui sa configurăm contul nostru de github prin introducerea următoarelor comenzi
   * **git config --global user.email** [**nume@exemplu.com**](mailto:nume@exemplu.com)
   * **git config --global user.name „nume”**

Odată cu crearea unui repository se va începe si lucrul la aplicație, acest fapt ducând la dorința de a salva codului pe GitHub. Pentru a salva fișierele modificate vor fi câteva comenzi ce vor trebui rulate în Command Prompt sau Git Bash în interiorul repository-ului.

* **git status** -atunci când se va dori verificarea fișierelor modificate de la ultimulcommit.
* **git add nume-fișier** când se va dori adăugarea unui fișier sau **git add .** pentruadăugarea tuturor fișierelor modificate de la ultimul commit, în staging pentru un viitor commit.
* **git commit -m „NumeComit”** pentru aadăugaschimbările permanent înrepository însoțit de parametrul -m dacă se va dori să se ofere respectivului commit un mesaj dorit.

Într-un repository de git există și posibilitatea de a lucra pe mai multe branches(ramuri)

cu scopul lucrului în paralel la două versiuni diferite. De exemplu, puteți crea un branch "de testare" și un branch "de producție" pentru a urmări modificările pentru diferite versiuni. În această aplicație, s-a lucrat pe un singur branch, de development, deoarece o singură persoană a lucrat în dezvoltarea aplicației și nu a fost necesar lucrul pe branch separat.