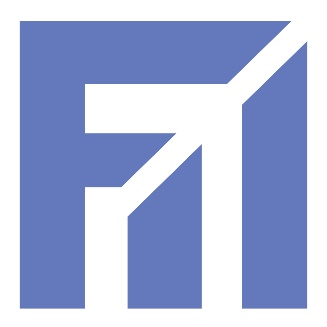
UNIVERSITATEA “ALEXANDRU IOAN CUZA” DIN IAȘI

**FACULTATEA DE INFORMATICĂ**



LUCRARE DE LICENȚĂ

**FII Dashboard**

propusă de

**Teodora Chiosa**

**Sesiunea**: Februarie, 2018

**Coordonator științific**

**Drd. Florin Olariu**

**Conf. Dr. Adrian Iftene**

**UNIVERSITATEA “ALEXANDRU IOAN CUZA” DIN IAȘI**

**FACULTATEA DE INFORMATICĂ**

**FII Dashboard**

**Teodora Chiosa**

**Sesiunea**: Februarie, 2018

**Drd. Florin Olariu**

**Conf. Dr. Adrian Iftene**

**DECLARAȚIE PRIVIND ORIGINALITATE ȘI RESPECTAREA**

**DREPTURILOR DE AUTOR**

Prin prezenta declar că Lucrarea de licență cu titlul „FII Dashboard” este scrisă de mine și nu a mai fost prezentată niciodată la o altă facultate sau instituție de învățământ superior din țară sau din străinătate. De asemenea, declar că toate sursele utilizate, inclusiv cele preluate de pe Internet, sunt indicate în lucrare, cu respectarea regulilor de evitare a plagiatului:

- toate fragmentele de text reproduse exact, chiar și în traducere proprie din altă limbă, sunt scrise între ghilimele și dețin referința precisă a sursei;

- reformularea în cuvinte proprii a textelor scrise de către alți autori deține referința precisă;

- codul sursă, imaginile etc. preluate din proiecte open-source sau alte surse sunt utilizate cu respectarea drepturilor de autor și dețin referințe precise;

- rezumarea ideilor altor autori precizează referința precisă la textul original.

Iași, data

Absolvent *Teodora Chiosa*

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

(semnătura în original)

**DECLARAȚIE DE CONSIMȚĂMÂNT**

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul „FII Dashboard”, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de Informatică.

De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea „Alexandru Ioan Cuza” din Iași, să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Iași, data

Absolvent *Teodora Chiosa*

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

(semnătura în original)

**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, Teodora Chiosa.

Încheierea acestui acord este necesară din următoarele motive:

MOTIVE HERE; DON’T FORGET

Iași, data

Absolvent *Teodora Chiosa*

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

(semnătura în original)

CUPRINS

[Introducere 1](#_Toc515894009)

[Contribuții 3](#_Toc515894010)

[Aspecte teoretice 4](#_Toc515894011)

[User Experience 4](#_Toc515894012)

[Ce este și de ce este important? 4](#_Toc515894013)

[UX Personas 6](#_Toc515894014)

[UX Writing 8](#_Toc515894015)

[Principii fundamentale ale interacțiunii 9](#_Toc515894016)

[Care e diferența între UX și UI? 10](#_Toc515894017)

[User Interface 11](#_Toc515894018)

[Alegerea elementelor de interfață 11](#_Toc515894019)

[*Best practices* pentru un design de interfață 11](#_Toc515894020)

[Procesul UI Design 12](#_Toc515894021)

[UI Development 14](#_Toc515894022)

[SCSS 14](#_Toc515894023)

[SCSS: Avantaje 15](#_Toc515894024)

[SCSS: Dezavantaje 17](#_Toc515894025)

[CSS3 17](#_Toc515894026)

[Front-end 17](#_Toc515894027)

[Aspecte practice 18](#_Toc515894028)

[User experience 18](#_Toc515894029)

[User interface 27](#_Toc515894030)

[UI development 31](#_Toc515894031)

[Bibliografie 34](#_Toc515894032)

# Introducere

Prin lucrarea de față doresc să vin în ajutorul studenților Facultății de Informatică și să ofer informații centralizate asupra notelor, taxelor, prezențelor, orarului, materialelor de studiu, temelor, noutăților din cadrul facultății, prin intermediul unei interfețe prietenoase.

În momentul actual, informația este dispersată. Lipsa de uniformitate și lipsa update-urilor aduse site-urilor, atât ca interfață cât și ca tehnologii utilizate, nu prezintă un beneficiu pentru imaginea Facultății de Informatică și nu reflectă ideea de inovație în domeniu.

Lipsa de uniformitate se exemplifică prin următoarele:

* folosirea multiplelor conturi pentru accesul la informații:
  + pe eSIMS se pot găsi datele personale ale studentului, notele finale și obligațiile de plată, autentificarea fiind făcută pe baza numărului matricol
  + pe emailul facultății (Webmail) se primesc notificări de la profesori sau secretare cu privire la notele de pe parcursul semestrului, adeverințele CTP, orarul obișnuit si cel de examene
  + fără nici un cont, oricine are acces la orar și notele intermediare sau finale ale studenților ce se află pe paginile publice ale profesorilor (cu excepția celor protejate de o parolă)
* pentru a avea acces la materialele și notele unor discipline, studentul trebuie sa rețină numele site-ului profesorului și, unde este cazul, parolele anunțate la curs/laborator
* formula de calcul a notei finale diferă de la o disciplină la alta și trebuie căutată în fișa disciplinei
* se observă lipsa de update-uri, starea nefinalizata a paginilor, ceea ce duce la o experiență nu tocmai user-friendly pe aceste site-uri; de exemplu: titlul paginii eSIMS este “Untitled page”, limbajul folosit pe prima pagină nu inspiră profesionalism, butonul de autentificare este greu de reperat la o primă accesare a paginii
* inconsistența este relevată și de paginile profesorilor: întâlnim site-uri ușor de folosit dar și site-uri *plain text* ce nu folosesc link-uri, navigarea fiind făcută prin a tasta un nume de pagină direct în URL

Având in vedere acestea, voi realiza prin prezenta lucrare:

* crearea unui portal special destinat studenților
* folosirea unui singur cont pentru toate informațiile prezentate mai sus
* restructurarea si rearanjarea informațiilor în pagină pentru a facilita experiența utilizatorului pe acest site
* standardizarea paginilor profesorilor pentru facilitarea găsirii informațiilor
* posibilitatea înregistrării prezențelor online
* includerea știrilor site-ului Facultății de Informatică
* includerea notificărilor legate de Webmail
* aducerea unei perspective fresh, în concordanță cu trendurile actuale din domeniile UI/UX si Front-end

Primul capitol al acestei lucrări, intitulat „Contribuții”, va prezenta elementele inedite pe care le aduce această lucrare, si anume unificarea site-urilor existente în prezent, sub accesul unui singur cont, cu o interfață ușor de folosit/user-friendly.

În paginile ce urmează, voi trata următoarele aspecte in această ordine:

* UX: experiența utilizatorului pe acest site (e.g. cât de intuitivă și rapidă este navigarea în aplicație, găsirea de informații fără multe confuzii si click-uri),
* UI: design-ul acestuia (e.g. cât de plăcut din punct de vedere estetic este, dacă reflectă imaginea universității/facultății, dacă inspiră profesionalism, modernizare si inovație)
* Preprocesoare CSS (mai exact SASS/SCSS): avantaje, reutilizarea codului/DRY, folosirea optimă, arhitectura stilurilor
* CSS 3: noutăți si obstacole
* Framework-ul Angular 5: folosirea acestuia împreună cu SASS și HTML

Tehnologiile/plugin-urile folosite vor fi: Angular 5, SASS/SCSS, componente Angular Bootstrap (fără CSS).

Codul sursă al aplicației se poate găsi pe Team Foundation Server la adresa: <https://teochiosa.visualstudio.com/FIIDashboard> (transfera spre Github in iunie - public)

# Contribuții

Având drept ghid funcționalitățile actual implementate de către facultate/universitate, am creat o aplicație web ce este mai ușor și mai plăcut de folosit de către studenți, dându-i o înfățișare nouă și folosind tehnologii moderne.

Drept îmbunătățiri majore precizez:

* Experiența mai bună a utilizatorului pe site
* Design simplu și atractiv
* Posibilitatea de a vedea aplicația pe un ecran de mobil (responsive), un „must” al zilelor noastre
* Stilizarea aplicației folosind preprocesorul de CSS: SASS – un mare ajutor în etapele de dezvoltare și mentenanță a aplicației
* Folosirea limbajului Typescript pentru organizarea mai bună a codului și definirea mai strictă a variabilelor
* Separarea Front-end-ului de Back-end pentru modularitate

# Aspecte teoretice

În cele ce urmează, voi detalia aspecte teoretice (folosind articole online sau imprimate și imagini) și practice (cu exemple din aplicație sau exemple externe).

După cum bine știm, Facultatea de Informatică ne oferă bazele programării si dorește să ne trezească interesul spre a învăța, nu numai ce se preda la cursuri, ci mai ales concepte, limbaje si tehnologii noi, capacitatea de adaptare la nou fiind esențială în domeniu.

Ieșit din băncile facultății, un student se va lovi de asemenea provocări, de colaborarea în echipă, de metode de lucru etc. Astfel își va da seama că scopul unei aplicații nu este doar să efectueze sarcinile detaliate de client, ci să ofere funcționalități și într-un mod intuitiv, atractiv, ușor de folosit și eficient din punct de vedere al timpului.

## User Experience

### Ce este și de ce este important?

Potrivit definiției oferită de ISO (International Organization for Standardization), alineatul 3.23, Experiența utilizatorului este “percepția si reacția unei persoane, ce rezultă din folosirea și/sau folosirea anticipată a unui produs, sistem sau serviciu”, nota nr. 2 precizând și că este „o consecință a imaginii brand-ului, prezentării, funcționalității, performanței sistemului, comportamentului interactiv, capabilităților de asistență a sistemului interactiv, a stării interne și fizice a utilizatorului rezultate din experiențe precedente, atitudine, abilități, personalitate și contextul de utilizare”.

(ISO/IEC JTC 1/SC 7, 2016)

Într-o formă mai ușor de înțeles, UX este tot ceea ce gândești si simți în timpul folosirii unui produs, serviciu etc. Experiența poate fi una bună, atât de mulțumitoare încât nu iți vine în minte nimic de reproșat la acel produs, sau, în extrema cealaltă, atât de nesatisfăcătoare încât simți confuzie, nervozitate, regret asupra timpului pierdut și dorința puternică de a renunța la folosirea acelui produs.

Un rol foarte important în UX îl are experiența anterioară a utilizatorului în contextul produselor cu rol similar: spre exemplu, pentru închiderea unui element (ce poate fi o fereastră de sistem de operare, un link spre *Sign in/Sign out*, sau un modal într-o aplicație web, utilizatorul și-a format reflexul de a îndrepta cursorul mouse-ului spre colțul din dreapta sus al ecranului). Rolul unui UX designer este de a intra în pielea unui utilizator obișnuit și de a empatiza cu nevoile și obiceiurile sale, în același timp urmând trendurile actuale în domeniu, pentru crearea unei interfețe adecvate audienței țintă.

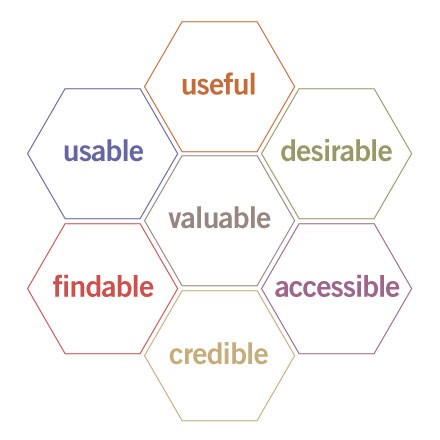


Figura - User Experience Honeycomb, Peter Morville (Morville, 2004)

Figura 1 - User Experience Honeycomb, Peter Morville (Morville, 2004) dorește să ilustreze principalele atribute specifice UX: valoros, ușor de găsit, folositor, utilizabil, dezirabil, accesibil și credibil.

În procesul său de lucru, un UX designer va trebui să își pună următoarele întrebări referitoare la produsul ce urmează a fi dezvoltat:

**Produsul poate fi folosit? (Usable)**

Dacă produsul final prezentat în fața clientului este incomplet sau prost făcut și deci nu poate fi folosit, continuarea interacțiunii cu acesta ar fi în van. Acesta este atributul de bază.

**Poți găsi ușor informația de care ai nevoie? (Findable)**

Este meniul intuitiv? Bara de căutare este unde te-ai aștepta să fie? Dacă trebuie să depui efort pentru a găsi ceva, înseamnă că UX-ul lasă de dorit. Același lucru îl putem spune și despre drumul parcurs până la produsul precizat: e.g. motorul de căutare a găsit ce trebuie în primele rezultate, sau nu s-a investit în Metadata pentru indexarea paginilor aplicației?

**Produsul satisface nevoile clientului? (Useful)**

Există situații în care designul și facilitatea de operare sunt impecabile, însă produsul nu oferă ceea ce clientul dorește iar acesta își va pierde rapid interesul.

**Dorești să folosești produsul? (Desirable)**

Designul este intuitiv? Este o adevărata plăcere sa folosești acel produs? Atunci vei dori să-l folosești din nou, poate chiar să îl recomanzi altora.

**Produsul are valoare? (Valuable)**

Poți spune că ai salvat timp, efort fizic, bani sau că nu ai mai văzut ceva similar pe piață? Dacă un produs nu are valoare, nu va prezenta interes pentru mult timp, oricât de frumos și ușor de folosit ar fi.

**Ai încredere în produs? (Credible)**

Se aplică mai ales magazinelor online (nu ai dori să dezvălui datele cardului de credit oricui). Limbajul folosit în scrierea conținutul, designul atractiv si minimalist pot contribui la o imagine profesională și credibilă.

**Este produsul accesibil? (Accesible)**

Oricine ar trebui să aibă acces la produs, indiferent de dizabilitățile sale.

(DeRome, 2015)

### UX Personas

O *persona* reprezintă un grup de utilizatori ce manifestă tipare comportamentale similare precum: decizii de cumpărare, utilizarea de tehnologii sau produse, stilul de viață etc.

(O’Connor, 2011)

De ce avem nevoie de *personas*?

* Pentru a ști care este audiența țintă a produsului
* Pentru a înțelege mai bine care sunt problemele și nevoile audienței
* Pentru a concentra efortul echipei asupra creării unui produs utilizabil
* Pentru a nu întâmpina discuții legate de funcționalități ce nu vor fi utilizate
* Pentru ca echipa de dezvoltatori să înțeleagă mai bine motivele și modalitățile de interacțiune cu produsul

Un număr ideal de *personas* ar fi până la 4, ideea fiind explorarea și adresarea nevoilor diverșilor utilizatori pentru a crea un produs pentru o audiență cât mai largă, fără a target-a pe toată lumea în același timp. Spre exemplu, pentru un produs destinat unei companii mari, cu utilizatori purtând diverse titluri și având pregătiri diferite (șefi, muncitori obișnuiți, etc.), va fi nevoie de un număr mai mare de *personas* pentru a trata toate aceste categorii de angajați.

(Tools, 2014)

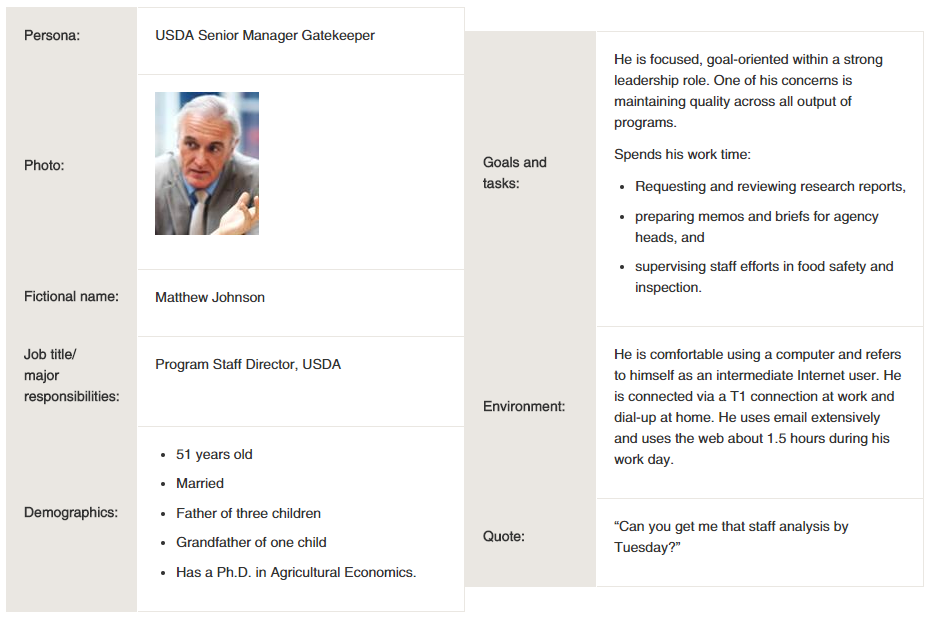


Figura - Persona dezvoltată de către Departamentul U.S. al Agriculturii (USDA) Economic Research Service (ERS)

După cum se vede în Figura 2 - Persona dezvoltată de către Departamentul U.S. al Agriculturii (USDA) Economic Research Service (ERS), elementele unei *persona* sunt următoarele:

- grupul din care face parte (e.g. manager)

- numele fictiv

- funcția la locul de muncă și responsabilitățile sale

- vârsta, educația, etnia, genul, statutul familial

- motivul pentru care ar folosi produsul dezvoltat

- starea fizică, socială și tehnică

- un citat ce rezumă funcționalitatea dorită de la produs

- o poză ce reprezintă această *persona*

(Usability.gov, Personas, fără an)

### UX Writing

Un UX writer folosește cuvinte pentru a crea interfețe, prin a scrie conținut text (numit „copy”) ce ajută utilizatorul să-și atingă scopul, să înțeleagă mai bine ce urmări au acțiunile sale într-o aplicație. Pentru a crea interfețe mai ușor de utilizat, conținutul nu trebuie lăsat la sfârșit. Dacă ne gândim la site-ul nostru ca la o conversație cu utilizatorul, conținutul are prioritate, și nu amplasarea sa în pagină.

Pe scurt, UX writing înseamnă:

* Scrierea de conținut pentru design, unde cuvintele au scop și au în vedere dorințele utilizatorului (Figura 3 - exemplu de UX writing pe Mailchimp - oferirea de sugestii relevante în procesul de logare)
* Luarea în considerare a modului de gândire și a sentimentelor utilizatorului
* Evitarea textului robotic și impersonal

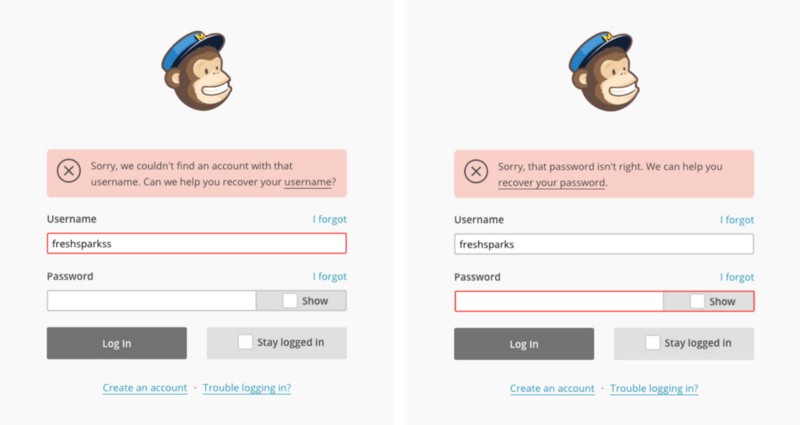


Figura - exemplu de UX writing pe Mailchimp - oferirea de sugestii relevante în procesul de logare

(uxwriter.cc, 2017)

### Principii fundamentale ale interacțiunii

Experiența cu un produs este crucială deoarece determină cu cât drag își amintește utilizatorul de interacțiunile sale cu acel produs. Experiența în ansamblu a fost una pozitivă sau frustrantă și confuză? Când interacționăm cu un produs, trebuie să ne dăm seama cum funcționează (ce face, ce operațiuni sunt posibile), ceea ce vom numi detectabilitate (discoverability). Câteva concepte ce o alcătuiesc sunt următoarele: *affordances*, *signifiers* și *feedback*.

**Affordances**

Termenul de *affordance* se referă la o relație între proprietățile unui obiect și capabilitățile agentului ce determină cum poate fi folosit acel obiect. Spre exemplu, sticla permite transparență dar în același timp blochează trecerea altor obiecte fizice – deci sticla ne permite suport și posibilitatea de a vedea prin ea dar nu și trecerea aerului sau a altor obiecte (cea din urmă fiind considerată *anti-affordance*, împiedicarea interacțiunii). Pentru a fi eficace, acestea două (*affordance*, *anti-affordance*) trebuie să fie sesizabile. Dacă nu pot fi percepute, atunci dorim semnalarea prezenței acestora prin *signifier*s.

**Signifiers**

În timp ce *affordance* determină ce acțiuni sunt posibile, *signifier* comunică unde trebuie să aibă loc acțiunea. Utilizatorii vor căuta indicii sau orice semne ce i-ar putea face să înțeleagă produsul ce urmează a-l folosi. Astfel, termenul de *signifier* se referă la orice semn sau sunet, orice indicator perceptibil ce comunică un comportament unei persoane (cuvinte, o ilustrație grafică, etc.).

**Feedback**

*Feedback* înseamnă comunicarea rezultatelor unor acțiuni. Acesta trebuie să fie instant (o întârziere de o secundă deja va ridica semne de întrebare iar dacă întârzierea este prea lungă utilizatorii vor renunța la folosirea unui produs). Nu în ultimul rând, *feedback*-ul trebuie să fie informativ (e.g. simpla atenționare asupra unei erori de sistem nu comunică suficient unui utilizator, pe când detalierea erorii și oferirea unor sugestii de acțiuni ajută enorm).

(Norman, 2013)

### Care e diferența între UX și UI?

Adeseori, aceste două concepte sunt privite ca unul și același lucru. Deși in majoritatea cazurilor, UI designer-ul are cunoștințe de UX (având titlul de UI/UX designer), acestea sunt de fapt lucruri diferite. Ca o analogie, gândiți-vă la corpul uman: “UX reprezintă tot ce este intern și face corpul să funcționeze bine, iar UI este ce se află la suprafață, tot ce face corpul să arate frumos și atrăgător”.

(DeRome, 2015)

## User Interface

Interfața utilizatorului se focusează pe anticiparea nevoilor utilizatorilor și asigurarea unei interfețe cu elemente ușor de accesat și ușor de înțeles. UI combină conceptele UX, Arhitectura Informației și Design.

### Alegerea elementelor de interfață

Utilizatorii s-au familiarizat cu un anumit comportament al unor elemente de interfață; de aceea este indicat ca produsul dezvoltat să fie consecvent și predictibil în decizii și layout/așezarea elementelor. Astfel facilităm completarea eficientă a taskurilor utilizatorului și satisfacția acestuia.

Câteva elemente de interfață sunt:

* Input-uri – butoane, câmpuri text/numerice/etc., casete de bifare/selectare, dropdowns, toggles
* Componente de navigare – breadcrumbs, câmp de căutare, paginare, iconițe
* Componente de informare – tooltips, iconițe, bare de progres, notificări, modale, toasters
* Elemente container – acordeoane, tab-uri

### *Best practices* pentru un design de interfață

**Simplitatea** – folosirea limbajului clar și a adnotărilor, evitarea elementelor inutile

**Consecvența dată de elemente comune/generale** – odată învățat comportamentul unui element, utilizatorul poate folosi acea informație în alte pagini sau zone ale aplicației

**Aranjarea în pagină cu scop** – Spațierea și ordinea elementelor în pagină dictează importanța acestora și sporește găsirea informațiilor relevante utilizatorului

**Culorile și texturile** – direcționarea atenției folosind culoare, luminozitate, contrast, textură (sau redirecționarea atenției de la un element la altul)

**Tipografia în crearea ierarhiei și clarității** – folosirea diferitelor mărimi, fonturi, aranjamente ale textului poate crește capacitatea de scanare și lizibilitatea

**Comunicarea dintre sistem și utilizator** – informarea în permanență relativ la acțiuni, schimbări în sistem, erori folosind elemente UI (modale, toasters, etc.)

**Valorile predefinite** – setările sau valorile default în câmpuri pot salva din timpul utilizatorului

(Usability.gov, User Interface Design Basics, fără an)

### Procesul UI Design

Cunoscând elementele de bază cu care poate lucra și specificațiile clientului, un UI Designer poate începe să elaboreze interfața unei aplicații, creând o schiță, un wireframe, un mockup și în cele din urmă un prototip. Acestea sunt necesare pentru a clarifica împreună cu clientul care sunt așteptările sale asupra aplicației și pentru a salva timp și resurse financiare în comunicarea cu dezvoltatorii.

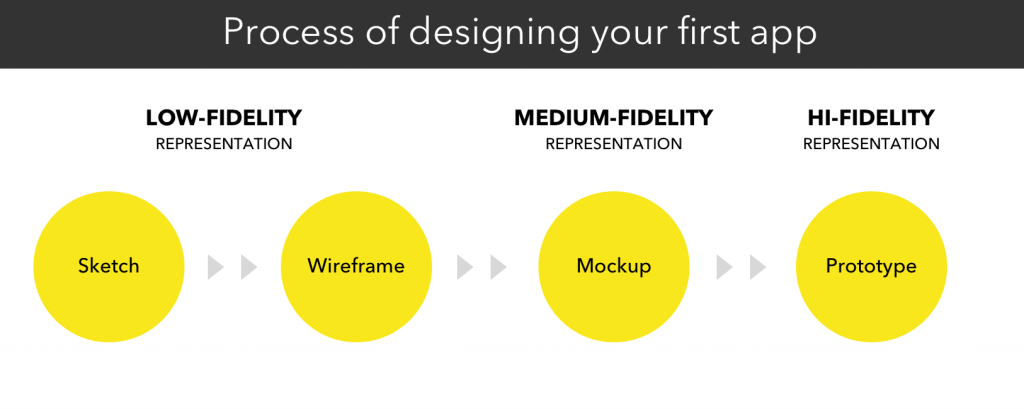


Figura - Procesul de design al unei aplicații

Pe scurt, după cum se vede în Figura 4 - Procesul de design al unei aplicații, Figura 5 - Rezumat al procesului de UI design, se începe cu o schiță simplă ce poate fi realizată pe hârtie. Apoi wireframe-ul este creat, unde conținutul și funcționalitățile sunt organizate. După adăugarea de culori, iconițe, imagini, logo-uri obținem mockup-ul iar după adăugare elementelor interactive, obținem prototipul.

**Schiță** – reprezentare a aplicației cu precizie scăzută, poate fi făcută pe o foaie de hârtie sau într-un program specializat, folositoare pentru a da formă ideilor venite în urma unei sesiuni de brainstorming

**Wireframe** – reprezentare a aplicației cu precizie scăzută, echivalent al unui schelet al aplicației, folosit pentru a organiza și ordona conținut și funcționalități, nu include culori, font-uri etc.

**Mockup** - reprezentare a aplicației cu precizie medie, implică adăugarea de culori, font-uri, conținut text, imagini, logo-uri

**Prototip** - reprezentare a aplicației cu precizie ridicată, folosește ca bază mockup-ul la care adaugă concepte de UX, interacțiune, animații; oferă senzația de folosirea a aplicației finalizate dar nu conține logică/funcționalitate, ci doar imagini conectate între ele

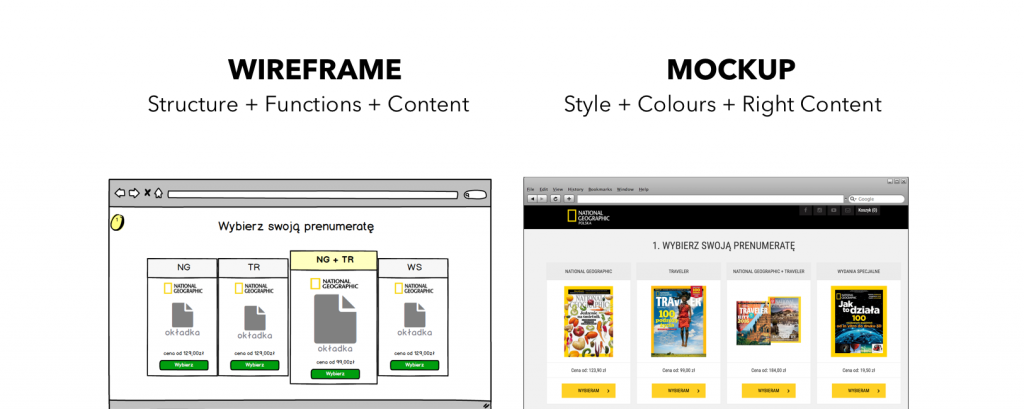


Figura - Rezumat al procesului de UI design

(Warcholinski, 2016)

## UI Development

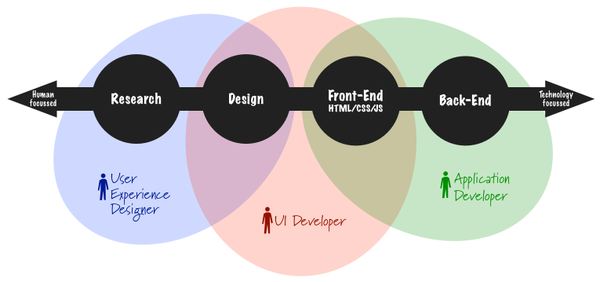


Figura - Vedere de ansamblu asupra procesului de dezvoltare a unei aplicații - de la idei, concepte și design spre implementarea propriu-zisă

Un UI/UX Designer combină cercetarea și abilitățile de design pentru a înțelege nevoile utilizatorului și produce concepte/soluții/design-uri. Accentul cade pe comportamentul uman și psihologie și implică creativitate în rezolvarea problemelor.

Un Application Developer este în general persoana care face produsul să funcționeze, folosind cod, logică și creează o interfață sumară, doar pentru a ilustra acele funcționalități.

Persoana de legătură, care se asigură că implementare în cod este în concordanță cu design-ul se numește UI Developer (Figura 6 - Vedere de ansamblu asupra procesului de dezvoltare a unei aplicații - de la idei, concepte și design spre implementarea propriu-zisă). Aceasta combină cunoștințe de design cu cele tehnice, realizează un produs care arată bine dar și funcționează și înțelege foarte bine comportamentele de redare ale browser-elor.

(Melbourne, 2011)

### SCSS

În dezvoltarea unei aplicații web, tehnologiile folosite pe latura Front-end sunt HTML, CSS și un limbaj ce oferă logică și conținut dinamic. La fel cum HTML a evoluat de-a lungul timpului, renunțând la etichete învechite și adăugând altele noi, unele urmărind să înlocuiască tehnologia Flash (HTML5), CSS a ținut pasul și a introdus variabile și proprietăți noi (CSS3). Cu toate acestea, s-a simțit în continuare lipsa organizării și repetabilitatea codului, mai ales în aplicațiile mari.

*Syntactically Awesome StyleSheets* intră în peisaj și oferă dezvoltatorilor UI soluții pentru aceste probleme.

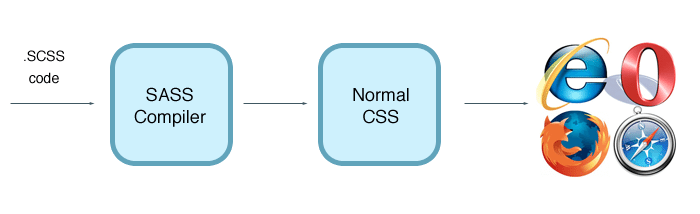


Figura - Procesul de compilare SCSS

(The Power of Sass and Why You Should Embrace CSS Preprocessors, 2017)

### SCSS: Avantaje

**Preprocesare** – din cauza faptului că fișierele CSS pot devenit extrem de mari, complexe și greu de întreținut, SASS oferă o experiență mai plăcută dezvoltatorului în timp ce scrie cod sau întreține și adaugă cod iar în final va genera CSS obișnuit ce va fi interpretat de browser (Figura 7 - Procesul de compilare SCSS).

**Variabile** – pot fi declarate global sau local, sporesc consecvența prin reutilizare, la compilare variabila este înlocuită cu informația stocată în ea

**Imbricarea/Nesting** – simularea ierarhiei HTML într-o formă CSS, oferă o vedere mai clară asupra structurii elementelor, reduce repetabilitatea

**Fișiere parțiale** – fișiere SCSS prefixate cu linie de subliniere care *nu* vor fi compilate spre CSS ci se comportă ca referințe (util pentru variabile, funcții, mixin-uri), sporesc modularitatea codului, se folosesc împreună cu directiva de @import

**Importarea** – fișierelor obișnuite sau parțiale SCSS, fișiere CSS simple; la compilarea se face concatenarea tuturor acestora iar apoi sunt servite browser-ului

**Mixin-uri** – blocuri de cod declarate cu denumire și parametri ce pot fi invocate oriunde în cod, cel mai des folosite pentru a reține prefixele unei proprietăți specifice fiecărui browser

**Extinderea/Moștenirea** – folosită împreună cu declarații *placeholder* (ce nu vor fi transformate în CSS dar pot fi extinse), noul element ce extinde placeholder-ul va avea în interior proprietățile placeholder-ului la care pot fi adăugate altele

**Operatori** – suport pentru operatori de bază (+, -, \*, /, %)

(Sass Basics, fără an)

Pe lângă acestea, SCSS mai oferă și:

* Posibilitatea folosirii sintaxei preferate de dezvoltator: SASS, ce este similar limbajului Python în sensul că indentarea dictează unde începe și unde se termină un bloc de cod, sau SCSS, care este similar CSS-ului și folosește acolade în delimitarea codului
* Tipuri de date suportate de variabile: numere, string-uri, culori, valori booleene, valori nule, liste, hărți(maps), referințe la funcții
* Funcții specifici culorilor (transparentize, adjust-hue, invert, complement etc.)
* Funcții și operații pentru string-uri, liste, map-uri
* Interpolare
* Directive de control (@if, @for, @each, @while)
* Pasarea unui bloc de cod drept parametru al unui mixin prin variabila @content
* Și multe altele

(Sass Documentation, fără an)

### SCSS: Dezavantaje

Proiectele mari cu multe pagini ce au nevoie de o interfață uniformă și consecventă pot profita la maxim de funcționalitățile SCSS. Însă, dacă se dorește dezvoltarea unei aplicații mici sau a unui site personal, SCSS poate însemna timp pierdut pe instalare și compilare și poate îngreuna procesul de dezvoltare. Pe lângă acest lucru, clasicul CSS ar putea lansa la un moment dat funcționalități comparabile cu cele oferite de SCSS iar trecerea înapoi la sintaxa CSS ar putea fi una grea.

### CSS3

CSS3 este ultima versiune a limbajului Cascading Style Sheets și aduce numeroase noutăți: colțuri rotunjite, umbre, gradiente, tranziții și animații, precum și tipuri de layout: multi coloană, flex și grid. Tranzițiile și animațiile (tranziții cu mai multe faze decât început și sfârșit) rulează fluid în orice browser modern și oferă un comportament natural elementelor din pagină. Din păcate, unele dintre aceste noi funcționalități nu sunt suportate, au suport redus sau necesită prefixe în browser-e precum Internet Explorer sau Safari.

## Front-end

Tehnologiile abordate pe latura front-end sunt Angular și Typescript.

Angular este un framework Javascript ce creează SPAs (single page applications) și a câștigat popularitate datorită faptului că clientul (device-urile utilizatorului) a devenit din ce în ce mai performant încât nu a mai fost nevoie ca serverul să preia toată procesarea. Astfel paginile sunt încărcate dinamic, fiind generat cod nou HTML, CSS și Javascript pentru fiecare componentă încărcată.

Datorită faptului că versiunile de Angular au suferit numeroase modificări (unele majore precum trecerea de la versiunea 1 la 2) iar proiectele *boilerplate* (proiecte deja create, cu unele dependințe instalate ce ar trebui să ruleze *out of the box*) găsite pe internet nu funcționau mereu cum trebuie, din dorința de standardizare a apărut Angular-CLI. Astfel, cu ajutorul unor comenzi simple, se pot crea și actualiza proiecte.

# Aspecte practice

## User experience

Aplicația, în stadiul curent, nu atinge toate atributele specifice unui bun UX deoarece nu dorește să fie decât un “proof of concept”. Însă, ne putem imagina că există o versiune finalizată a aplicației care utilizează date reale și comunică cu un server și o bază de date pentru a-și îndeplini scopul. Prin urmare putem spune că aplicația ar putea fi:

* valoroasă (deoarece nu mai există un produs similar destinat studenților facultății și deoarece satisface majoritatea nevoilor studenților)
* ușor de găsit (presupunând că va fi la fel de ușor de găsit de către motoarele de căutare precum actualele site-uri ale facultății și funcționalității aplicației sunt ușor de reperat)
* folositoare (studenții vor putea urmări știrile facultății, notele, absențele și taxele, notificările de emailuri etc.)
* utilizabilă (din nou, presupunând că versiunea completă există și își face treaba bine, fără ca server-ul să fie indisponibil sau alte probleme)
* dezirabilă (prin interfața sa prietenoasă și vioaie), accesibilă (pe desktop dar și pe ecrane mobile)
* credibilă (prezentând sigla oficială în concordanță cu ghidul brandului UAIC).

Pentru a avea o idee clară asupra audienței țintă, am cumulat experiența proprie cu o persona reprezentată în Tabelul 1.

|  |  |
| --- | --- |
| **PERSONA** | Student în anul 3 |
| **POZĂ** |  |
| **NUME FICTIV** | Ionescu Claudia |
| **FUNCȚIE/RESPONSABILITĂȚI** | * A fi la curent cu noutățile din cadrul universității/facultății pentru a ridica adeverință, semna contracte, plăti taxe etc. * A plăti taxele la timp * A învăța pentru seminare și examene * A obține note de trecere |
| **DETALII DEMOGRAFICE** | * Sex: feminin * Vârstă: 21 de ani * Stare civilă: necăsătorită * Naționalitate: română |
| **SCOP ȘI ACTIVITĂȚI** | * Studioasă și foarte organizată * Dorește să găsească informațiile de interes cât mai rapid * Atunci când nu studiază și nu este la facultate, iese în oraș cu prietenii, desenează și folosește intens aplicații de socializare pe telefonul mobil |

Tabela - Persona folosită pentru aplicația FIIDashboard

Din punct de vedere al conținutului de tip text, am încercat să folosesc cuvinte sugestive pentru link-uri și butoane iar mesajele de validare și pagina de “404 - pagina nu a fost găsită” oferă mesaje ajutătoare ce nu critică utilizatorul pentru greșelile sale ci îl îndrumă în direcția corectă (spre exemplu, la completarea greșită a unui câmp de email, se vede un mesaj de eroare ce ulterior oferă o sugestie de completare – “textul introdus respectă tiparul nume@domeniu.com?”).

În continuare voi aduce la lumină câteva probleme ale paginii actuale eSims, din punct de vedere UX. Logarea in contul studentului/profesorului este principala funcționalitate a site-ului, însă butonul de logare este greu de reperat, există simboluri care creează confuzie (ghilimelele «») iar logo-ul universității nu a fost updatat în conformitate cu ghidul brandului. De ce ar fi cea din urmă importantă? Deoarece creează confuzie (spre exemplu, la găsirea pe internet a unei pagini similare Facebook, ce folosește logo-ul său dar cu mici modificări de font și culoare, audiența ar deveni nedumerită și ar putea avea suspiciuni că se află pe un site de phishing – înșelăciune electronică care constă în obținerea unor date confidențiale precum credențiale sau date bancare prin manipularea identității unei persoane sau instituții).

Pentru exemplificare voi folosi poze din aplicația curentă.

**LOGIN**

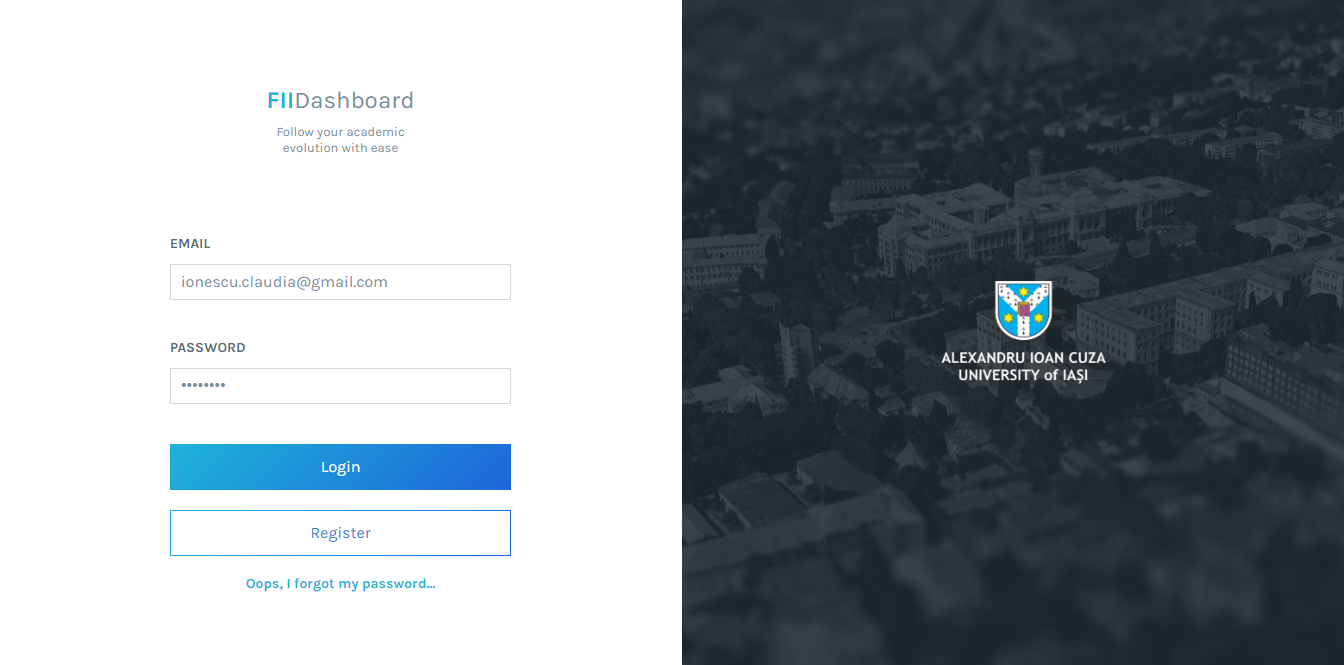


Figura – Logarea FIIDashboard pe desktop

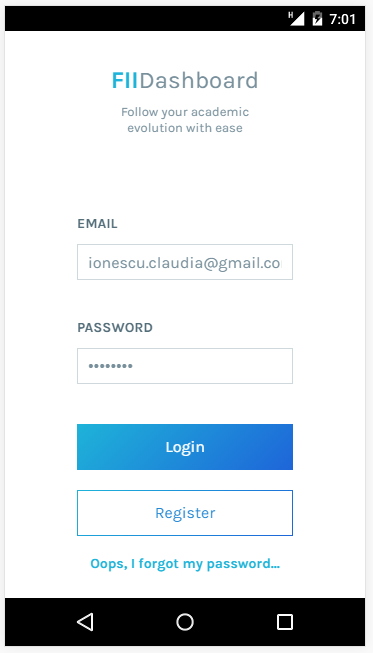


Figura – Logarea FIIDashboard pe mobil

De aceea, am creat o pagină de start ce va cere în primul rând logarea utilizatorului (Figura 8 – Logarea FIIDashboard pe desktop, Figura 9 – Logarea FIIDashboard pe mobil), prezentând titlul și scopul aplicației pe scurt. Drept câmpuri ce trebuie completate am optat pentru email si parolă din cauza faptului că nici un student nu își cunoaște numărul matricol pe de rost. Pentru a facilita completarea datelor, am introdus mesaje de validare ajutătoare. Butonul de logare captează prima oară atenția utilizatorului datorită fundalului său viu colorat iar dacă se dorește înregistrarea unui nou cont, atenția va cădea asupra butonului de înregistrare aflat dedesubtul celui de logare. În josul paginii se poate găsi și un link spre resetarea parolei în cazul uitării acesteia. Prin această primă pagină am comunicat utilizatorului câteva elemente ușor de reperat în restul aplicației: înfățișarea input-urilor, mesajelor de eroare/validare și gradul de importanță al butoanelor.

**LAYOUT**

Odată logat în aplicație, utilizatorul este întâmpinat de o pagină ce conține 3 zone majore: meniul din stânga, antetul și zona dinamică a aplicației.

**LEFT SIDE**

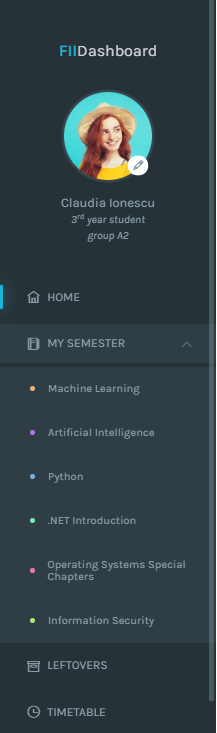


Figura - Meniul implementat

În zona de meniu (Figura 10 - Meniul implementat) se poate vedea titlul aplicației/logo-ul. Deoarece titlul este inițial informația cea mai importantă a paginii iar majoritatea utilizatorilor citesc de la stânga la dreapta, logo-ul a fost plasat în colțul din stânga sus. Imediat dedesubt a fost plasată o confirmare a identității utilizatorului logat (feedback primit după logare), reprezentată prin poza de profil și câteva detalii (nume, an și grupă). Imaginii de profil îi este asociată un indicator/signifier: o iconiță de editare ce va duce la pagina de profil.

**PROFILE**

Se oferă utilizatorului posibilitatea de a-și schimba poza de profil, e-mail-ul și numărul de telefon (deoarece acestea se schimbă cel mai de des), restul informațiilor fiind doar afișate/read-only. Consecvența se observă prin repetarea elementelor precum input-uri, mesaje de validare, butoane în aceeași manieră ca pe pagina de logare.

**MENU**

Tot în partea stângă am plasat meniul (Figura 10 - Meniul implementat) sub formă de acordeon și cu un număr acceptabil de link-uri, compactând pe cele similare într-un sub-meniu. Se oferă feedback utilizatorului prin afișarea unei benzi albastre lângă linkul spre pagina curentă și se oferă un signifier/indicator spre sub-meniu (dacă acesta există) printr-o săgeată ce indică “jos”. Dacă se dorește închiderea sub-meniului, săgeata ca fi direcționată în sus pentru a indica colapsarea acestuia. Acești indicatori se pot vedea la plasarea cursorului deasupra unui item din meniu. Animațiile, atât rotirea săgeții cât și expandarea/colapsarea sub-meniurilor, oferă feedback printr-o mișcare naturală, spre deosebire de schimbarea/apariția lor bruscă.

Sub-meniurile au fost folosite pentru obiectele din semestrul curent sau semestrele trecute și folosesc color-coding: pentru a repera mult mai ușor obiectele ce prezintă interes, au fost asignate 6 culori la cele 6 obiecte per semestru. Astfel utilizatorul poate naviga mai rapid în aplicație pe baza culorilor odată cu memorarea lor, nemaifiind nevoie să citească integral titlurile. Dacă studentul repetă obiecte din ani trecuți, cele 6 culori se vor repeta deoarece introducerea mai multor culori în aplicație poate fi copleșitor pentru utilizator.

Deoarece majoritatea studenților posedă un telefon mobil de tip smartphone, am decis ca aplicația să se adapteze nevoilor acestor studenți. Astfel, pentru browser-e rulând pe device-uri mici, meniul este ascuns și poate fi activat prin butonul *hamburger* (simbol universal pentru meniu)din colțul stânga sus. Acesta va ocupa tot ecranul pentru lizibilitate maximă și poate fi închis prin butonul de închidere din dreapta sus. Pe varianta pentru mobil se observă și opțiunea de delogare la finalul meniului.

**HEADER**

În partea de sus a ecranului am afișat un mesaj de „bun-venit” iar în partea dreaptă se află link-ul pentru delogare, ce a fost plasat în colț deoarece seamănă cu acțiunea de închidere a unui element (fereastră, modal etc.).

Dedesubtul antetului se află zona dinamică a site-ului, aceasta fiind populată cu alte elemente odată cu schimbarea paginii.

**HOME PAGE**

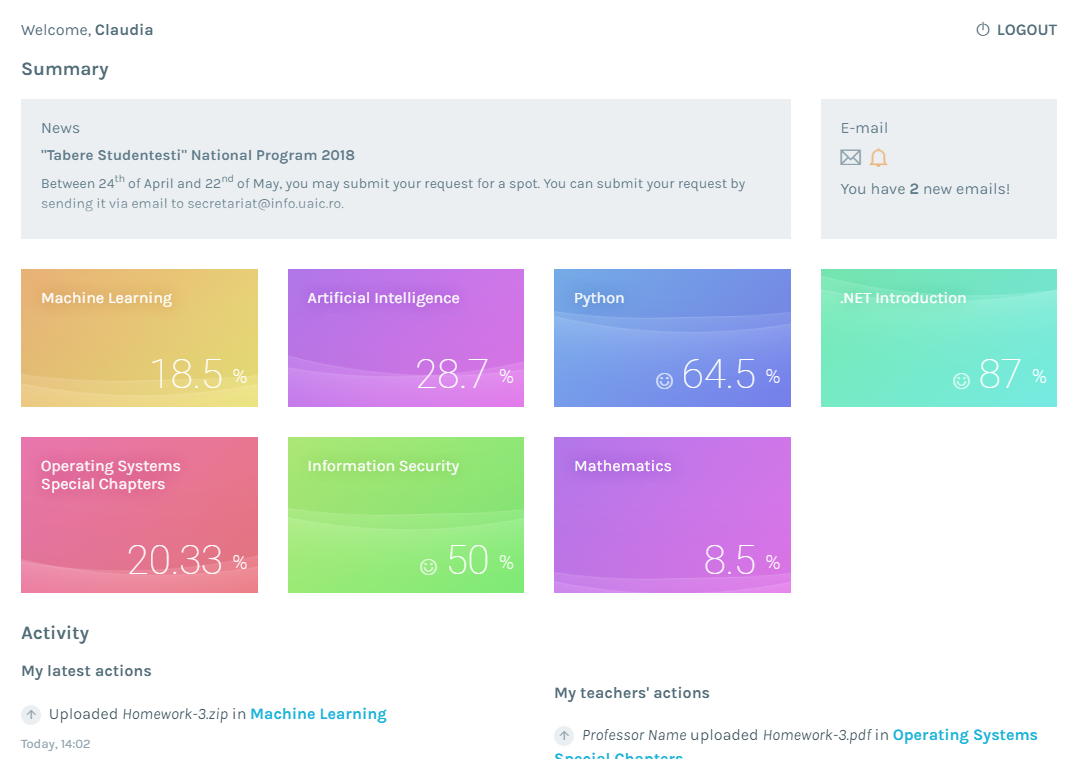


Figura - Pagina de start implementată (desktop)

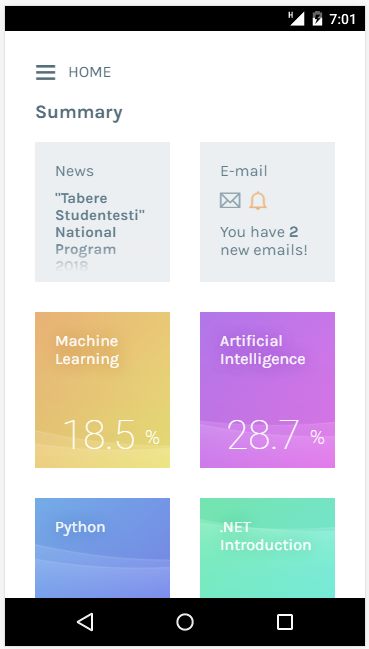


Figura - Pagina de start implementată (mobil)

Pagina de start (Figura 11 - Pagina de start implementată, Figura 12 - Pagina de start implementată (mobil)) oferă scurtături către cele mai probabil a fi accesate zone ale aplicației. Așadar, putem observa următoarele arii de interes:

* News – zonă ce va arăta ultimul anunț important scris de angajații facultății. Momentan, la click, duce spre pagina oficială de știri a Facultății
* Email – zonă ce indică utilizatorului câte email-uri necitite are. Iconița imită prin animația sa mișcarea unui clopot real și va apărea doar dacă există email-uri noi
* Tile-uri pentru obiectele studiate în semestrul curent cât și cele ce trebuie refăcute. Am refolosit color-coding pentru a repera mai repede obiectul de interes pe baza culorii acestuia. Pentru a oferi o vedere de ansamblu, am introdus animația unor valuri de apă, unde nivelul acestora indică subtil promovabilitatea. Procentajul de promovabilitate este scris în clar și este însoțit de o față zâmbitoare atunci când valoarea sa întrece 50%. Procentele sunt însoțite la acțiunea de *hover* de un mesaj ce confirmă situația curentă (e.g. „You passed!”). Fiecare dintre aceste tile-uri direcționează la click spre pagina disciplinei.
* Activity – secțiune a paginii ce oferă un istoric al acțiunilor studentului în aplicație dar și ultimelor acțiuni efectuate de cadrele didactice (încărcarea de fișiere, teme etc.). Aceste log-uri sunt însoțite de link-uri spre pagina în care s-au făcut modificări cât și de date temporale.

**TIMETABLE**

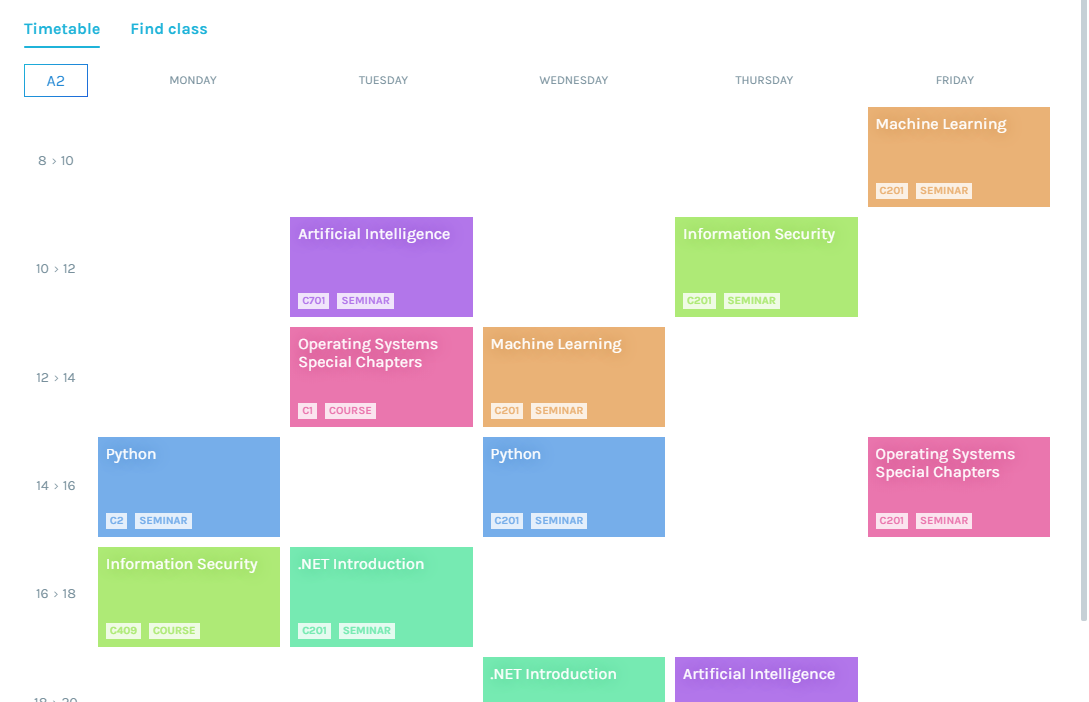


Figura - Orarul pe desktop

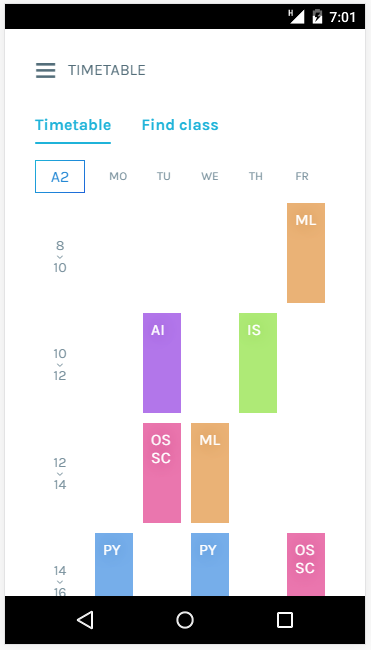


Figura - Orarul pe mobil

Una din paginile cele mai frecventate de studenți este cea a orarului (Figura 13 - Orarul pe desktop, Figura 14 - Orarul pe mobil).

Am folosit în continuare cele 6 culori aferente disciplinelor și am reorganizat informațiile într-un mod plăcut vizual. La intersecția zilei a săptămânii cu intervalul orar se află un element cu titlul disciplinei și cele mai relevante detalii: locația și tipul (curs/seminar). La acțiunea de click ne întâmpină un modal cu mai multe detalii precum numele profesorului și anul de studiu iar în josul modalului se află un buton scurtătură spre pagina disciplinei (pentru a nu fi nevoit utilizatorul să dea prea multe click-uri să ajungă în acea pagină prin ieșirea din modal și apoi prin meniu).

Butonul ce indică grupa selectată are rolul și de a schimba grupa ce poate fi vizualizată. Dacă se dorește o căutare avansată asupra tuturor orarelor, se poate folosi tab-ul “Find class” ce va încărca un tabel detaliat cu următoarele filtre posibile: numele disciplinei, numele profesorului, tipul disciplinei, locația, intervalul orar, grupa, ziua din săptămână și anul de studiu, toate oferind la tastare sugestii relevante de completare.

Aceleași funcționalități sunt păstrate și pe versiunea de mobil dar cu câteva modificări cerute de lipsa de spațiu: numele disciplinei s-a transformat în acronimul acesteia iar detaliile aferente se observă numai în modal.

Continuare…

## User interface

Aplicația oferă elemente cunoscute utilizatorului (butoane, input-uri, link-uri etc.) ce se comportă în mod consecvent în orice pagină. Interfața are un aspect curat și aerisit. Simplitatea este dată de limbajul concis, și spațierea elementelor. Informațiile sunt structurate în pagină după importanța lor (e.g. pagina de „acasă” în care găsim știri, notificări, situația generală a disciplinelor și ultimele modificări aferente acestora). Culorile folosite sunt neutre, gri-urile fiind folosite pentru a nu distrage atenția utilizatorului de la ce este mai important: elementele ce țin de discipline și folosesc una din cele 6 culori de accent sau albastrul folosit în aplicație pentru a semnala acțiuni (butoane, link-uri). Titlurile se diferențiază de textul simplu prin grosimea și mărimea fontului, ceea ce sporește capacitatea de scanare prin conținutul aplicației.

Primii pași în procesul de design sunt cercetarea și stabilirea nevoilor clientului. Utilizând propria experiență cu site-urile facultății și întrebările și nemulțumirile studenților revărsate pe grupurile de Facebook, am stabilit câteva funcționalități de bază.

În continuare am făcut o schiță mintală a aplicației iar apoi am desenat wireframe-ul, scheletul aplicației. Software-ul folosit în acest scop se numește Figma și este o aplicație online ce lucrează cu elemente vectoriale pentru a nu pierde din calitatea design-urilor create.

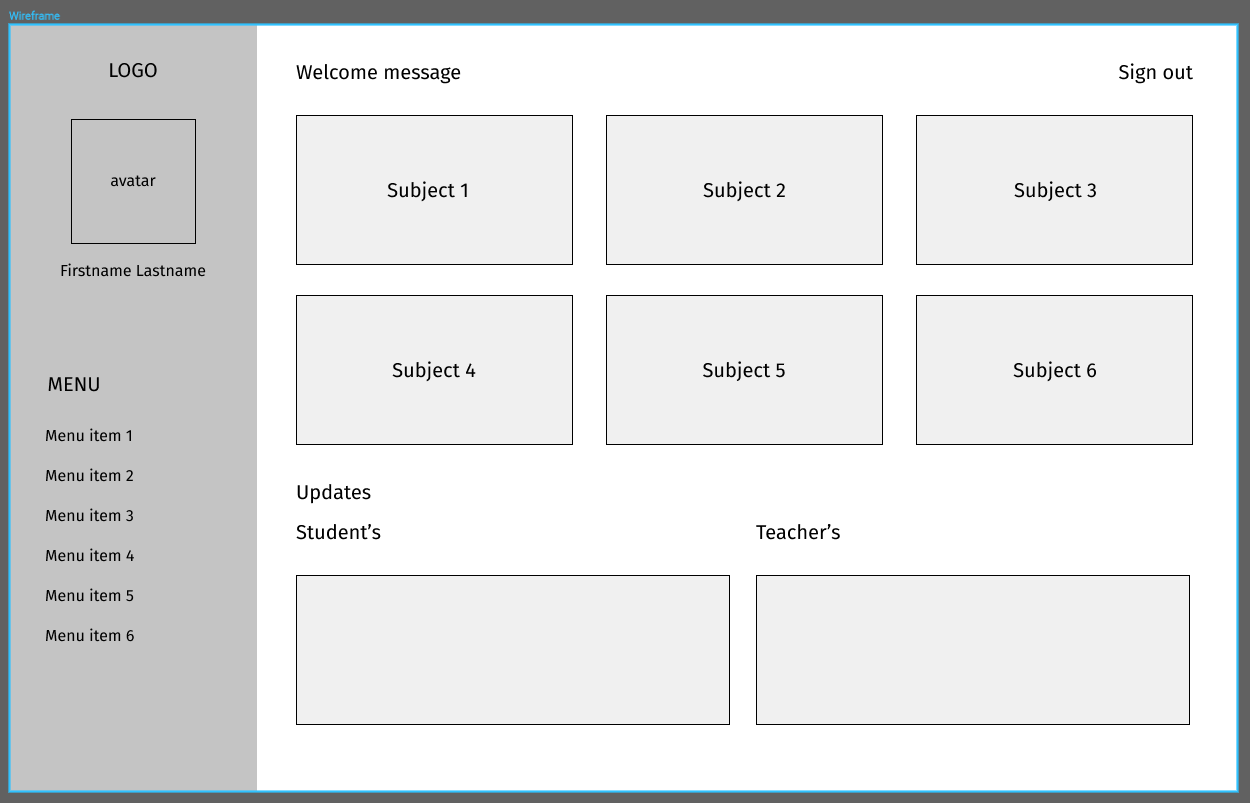


Figura - FIIDashboard - wireframe făcut pentru pagina de start

După efectuarea wireframe-ului (ilustrat în Figura 15 - FIIDashboard - wireframe făcut pentru pagina de start), tot în Figma am creat mockup-ul – adăugarea de culori, fonturi, elemente detaliate. Am creat și un scurt Styleguide (Figura 16 - Scurt Styleguide pentru FIIDashboard – prezentarea unor elemente generice) – un ghid sau un set de reguli ce asigură consecvența și uniformitatea designului.

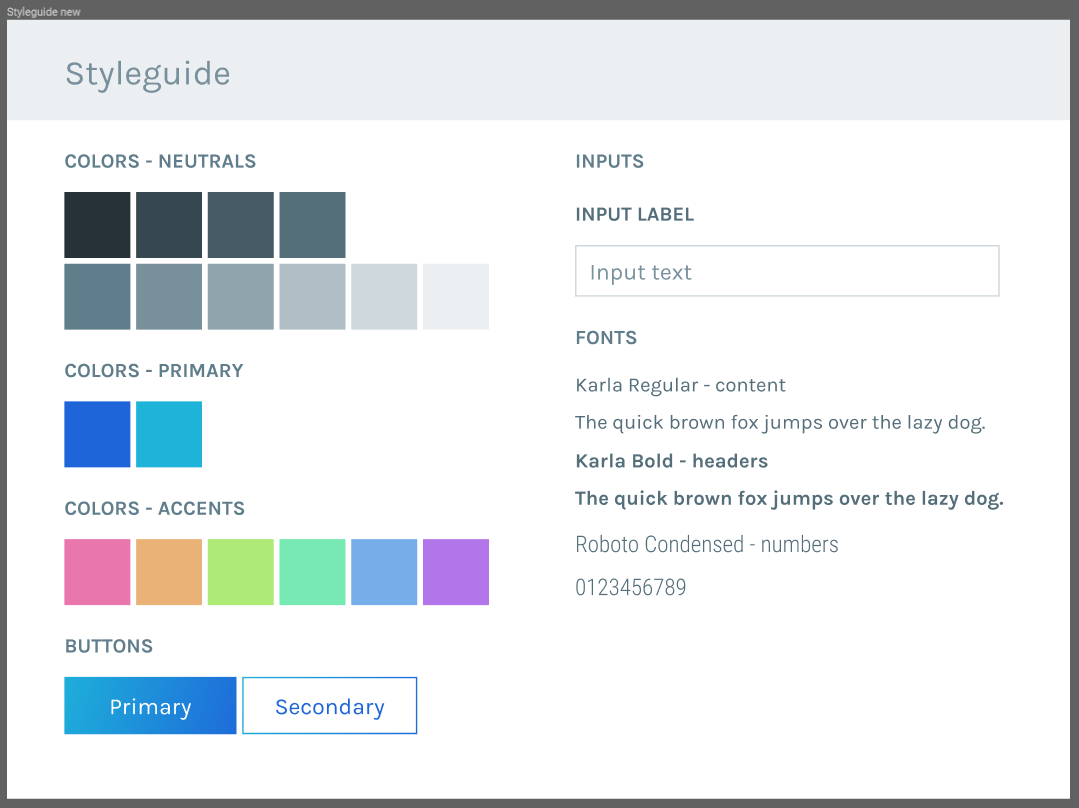


Figura - Scurt Styleguide pentru FIIDashboard – prezentarea unor elemente generice

Primele mockup-uri create au avut în vedere atmosfera generală a aplicației (look and feel) și nu sunt complete deoarece nu vizează toate paginile și nu prezintă toate datele și funcționalitățile ce doresc a fi implementate. De obicei designer-ul și dezvoltatorul sunt persone diferite, însă în acest caz am continuat design-ul prin implementarea acestuia și pe parcurs am improvizat și varianta de mobil.

Pentru început am exersat folosirea styleguide-ului în câteva pagini esențiale (pagina de login și cea de start) în Figura 17 - Mockup inițial pentru pagina de login FIIDashboard, Figura 18 - Mockup inițial al paginii de start, Figura 19 - Mockup detaliat pentru meniul aplicației.

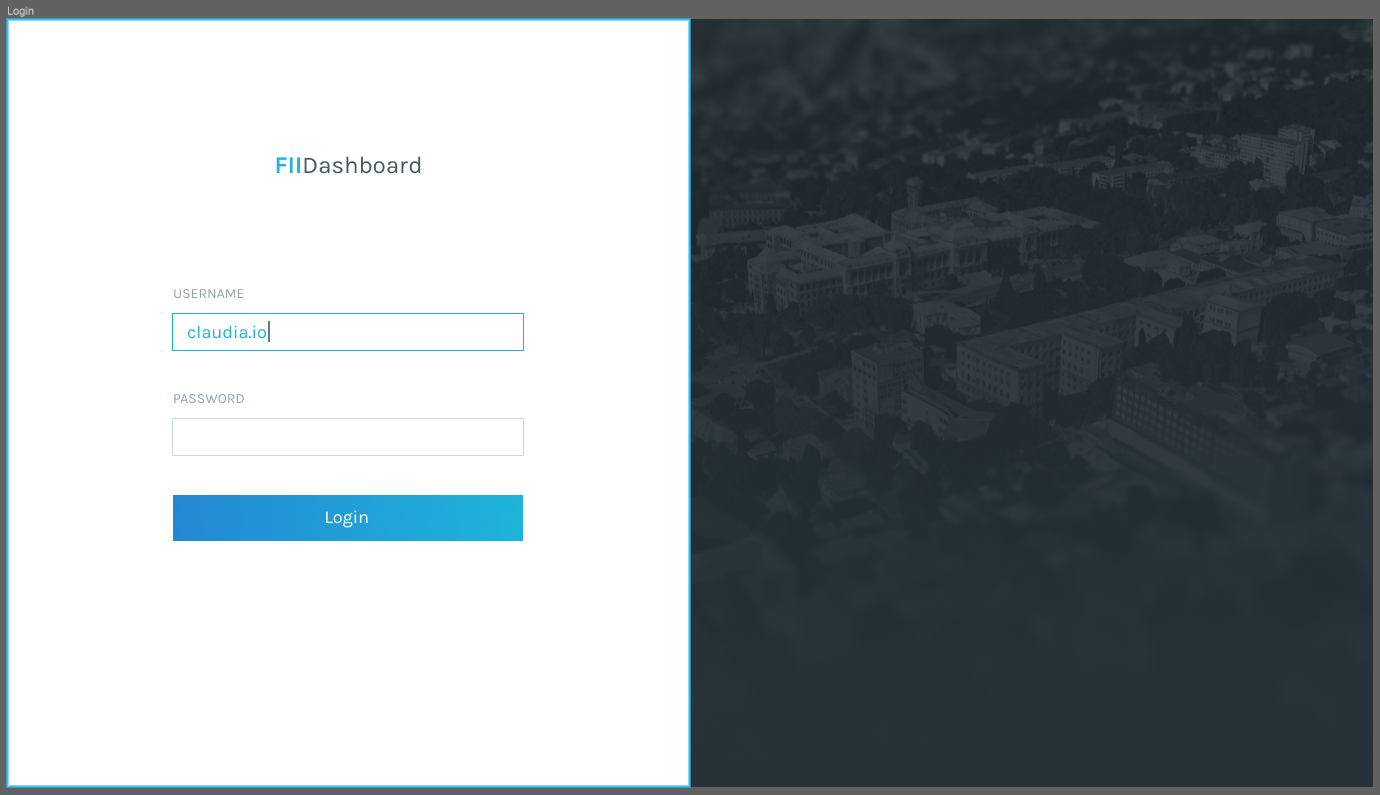


Figura - Mockup inițial pentru pagina de login FIIDashboard

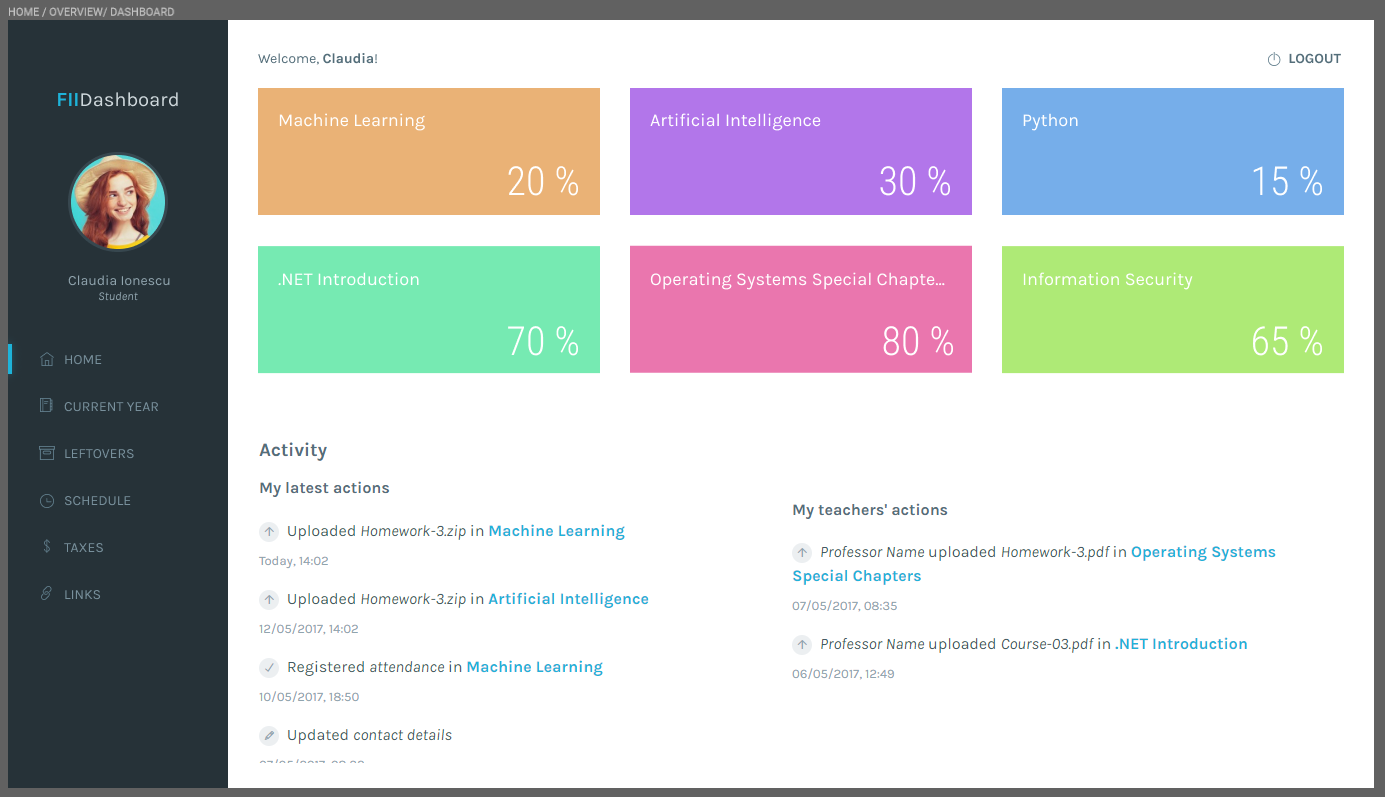


Figura - Mockup inițial al paginii de start

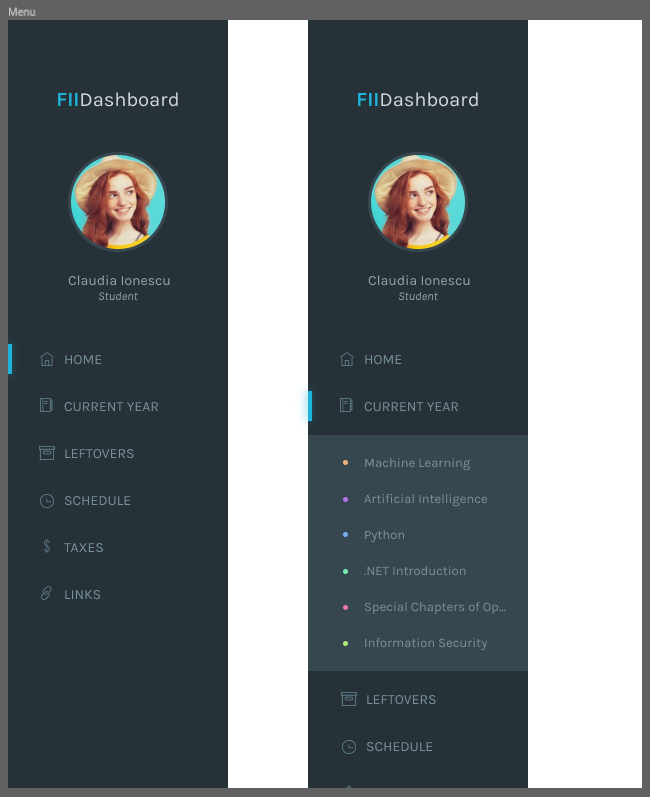


Figura - Mockup detaliat pentru meniul aplicației

În final ar trebui creat prototipul pentru a explica clientului și dezvoltatorilor cum se poate interacționa cu aplicația. Nu am creat și un prototip deoarece se poate arăta drept demo aplicația implementată.

## UI development

Din dorința de a avea un cod organizat și ușor de întreținut, am optat pentru Angular, tehnologie ce permite încapsularea stilului din componentă. Acest lucru înseamnă că stilurile unei componente vor fi aplicate doar acelei componente și nu vor putea modifica elemente din restul paginii. Astfel, dezvoltatorii au o vedere mai bună asupra codului, nefiind nevoiți să lucreze cu fișiere css foarte mari, ci lăsând asta doar în seama browser-ului.

Deși încapsularea stilului reprezintă un avantaj pentru lucrul cu componente, există și o cantitate mare de stil comun/generic ce trebuie organizat. Nu trebuie să abuzăm componentele dacă structura HTML este extrem de simplă și logica lipsește (e.g. butoanele se repetă în aplicație, însă nu are rost să creăm o componentă, ci vom folosi doar fișiere scss).

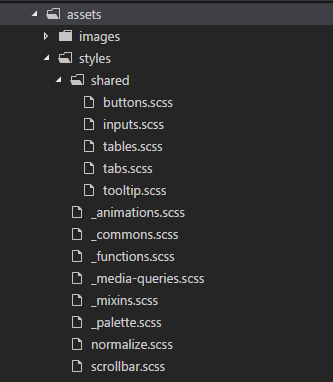


Figura - Arhitectura stilurilor SCSS

După cum se poate vedea în Figura 20 - Arhitectura stilurilor SCSS, pentru a nu aglomera fișierul standard *styles.scss*, am creat un număr de fișiere parțiale (cele prefixate de caracterul *underscore*) iar într-un fișier intitulat „shared” am plasat stil pentru elemente ce ar putea fi componente (deși în cazul de față nu a fost nevoie) dar trebuie să rămână consecvente în toată aplicația.

Printre fișierele parțiale se regăsesc:

* animațiile sub formă de keyframe-uri, responsabile pentru mișcarea de deschidere a sub-meniurilor sau pentru animația de încărcare a paginii etc.
* funcțiile ce generează cod css pe baza unor parametri (e.g. funcția *make-boxshadow* va cere ca parametru o culoare, pentru a genera două box-shadow-uri suprapuse cu nuanțe obținute din parametru)
* media-query-urile ce sunt de fapt variabile ce conțin segmente de media-query-uri ce pot fi combinate sau folosite de sine stătător; segmentele folosesc valori prag pentru mobil, tabletă tip portret, tabletă tip peisaj/landscape și desktop
* mixin-urile ce sunt blocuri de cod ce se pot repeta în diverse componente sau ce pot fi aplicate o singură dată global
* paleta de culori ce conține un map împărțit după 3 chei: “primary” - culorile de bază ale aplicației care nu sunt neutre), „accents” - accentele de culoare și „mono” - culorile neutre ale aplicației; accentele au fost generate folosind o culoare de bază iar apoi, prin funcția *hue* din scss, am rotit axa culorile oprindu-mă în 5 puncte ale axei

Aceste fișiere parțiale au fost adunate într-un alt parțial numit *commons* ce a fost impotat în fișierul *styles*. Astfel, codul va fi transformat în CSS doar la folosirea acestora.

Un fișier important este normalise.css ce are scopul de a dicta un comportament consistent pe toate browser-ele populare - Chrome, Edge, Firefox ESR+, IE 10+, Safari 8+, Opera (e.g. pentru etichetele <sup> <sub> se va aplica aceeași mărime a fontului și aceeași aliniere pentru toate browser-ele).

(Gallagher, fără an)

Pentru a crea aceeași experiență pe toate browser-ele, am folosit un plugin pentru un scrollbar, stilul fiind suprascris de către fișierul *scrollbar* din imagine.

Cât despre directorul *shared*, fișierele dinăuntrul său vor putea fi folosite doar la nevoie (e.g. nu vom importa *buttons.scss* dacă componenta nu folosește butoane deoarece nu dorim să generăm cod CSS în eticheta <head> în mod inutil).

Ordinea de scriere css!!!

# Bibliografie

DeRome, J. (2015, August 13). *What is User Experience?* Preluat de pe User Testing Blog: https://www.usertesting.com/blog/2015/08/13/what-is-user-experience/

Gallagher, N. (fără an). *normalize.css*. Preluat de pe Normalize.css: https://necolas.github.io/normalize.css/

ISO/IEC JTC 1/SC 7. (2016, 06). *ISO/IEC 25066:2016(en) - Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Common Industry Format (CIF) for Usability — Evaluation Report*. Retrieved from ISO: https://www.iso.org/obp/ui/#iso:std:iso-iec:25066:ed-1:v1:en:term:3.23

Melbourne, B. (2011, 11 10). *The difference between a UX Designer and UI Developer*. Preluat de pe asinthecity.com: https://asinthecity.com/2011/11/10/the-difference-between-a-ux-designer-and-ui-developer/

Morville, P. (2004, June 21). *User Experience Design*. Preluat de pe Semantic Studios: http://semanticstudios.com/user\_experience\_design/

Norman, D. (2013). *The Design of Everyday Things.* New York: Basic Books.

O’Connor, K. (2011, March 25). *Personas: The Foundation of a Great User Experience*. Preluat de pe UX Magazine: https://uxmag.com/articles/personas-the-foundation-of-a-great-user-experience

*Sass Basics*. (fără an). Preluat de pe sass-lang.com: https://sass-lang.com/guide

*Sass Documentation*. (fără an). Preluat de pe sass-lang.com: https://sass-lang.com/documentation/file.SASS\_REFERENCE.html

*The Power of Sass and Why You Should Embrace CSS Preprocessors*. (2017, 02 28). Preluat de pe 1stwebdesigner.com: https://1stwebdesigner.com/power-sass-why-use-css-preprocessors/

Tools, U. (2014). *How To Create UX Personas*. Preluat de pe Unamo: https://unamo.com/blog/general/how-to-create-ux-personas

Usability.gov. (fără an). *Personas*. Preluat de pe Usability.gov: https://www.usability.gov/how-to-and-tools/methods/personas.html

Usability.gov. (fără an). *User Interface Design Basics*. Preluat de pe Usability.gov: https://www.usability.gov/what-and-why/user-interface-design.html

uxwriter.cc. (2017, May 28). *What is a UX Writer?* Preluat de pe http://uxwriter.cc: http://uxwriter.cc/what-is-a-ux-writer/

Warcholinski, M. (2016, 04 20). *What Is The Difference Between Wireframe, Mockup and Prototype?* Preluat de pe brainhub.eu: https://brainhub.eu/blog/difference-between-wireframe-mockup-prototype/