Softbinator
TECHNOLOGIES

Dezvoltare software pentru dispozitive mobile

Laborator 1

Profesori de laborator

- Stefan Pavel (joi de la 6 la 8)
- Dorneanu Eduard-Gabriel (vineri de la 4 la 6)

Laboratoarele contin aceeasi informatie, ramane la alegerea studentului la care vrea sa participe.

Metoda de notare

- Aplicatii mobile. Aplicatia poate fi scris in orice limbaj de programare/framework (React-Native, Flutter, Native Android, Native IOS, etc.).
- Aplicatia va fi prezentata, iar codul va fi incarcat pe Github.

* Prezenta la laborator nu este obligatorie si nici nu se puncteaza.
** Cerintele proiectului le vom prezenta in saptamanile ce urmeaza.

Structura

La laborator vom lucra cu React Native, prin platforma Expo.

1. Introducere in JS, intro React Native, basic stuff, documentatie
2. Advanced React, styling, first app
3. React Native Navigation
4. State management (Context API, Redux), Fetch API, Async Storage
5. Misc:
    a. Native API (Camera, Sensors)
    b. Network state detect
    c. Screen size details
    d. Animations
    e. Notifications (with Firebase)

Project setup - Expo

Requirements:

1. Node.js + Npm/Yarn (https://nodejs.org/en/download)
    a. In terminal: npm install -g yarn (daca doriti sa folositi yarn)
2. VS Code/Webstorm sau orice editor
3. Expo Go (aplicatie de Android si iOS)
4. Expo CLI
    a. In terminal: npm install -g expo-cli

Project Creation

Dupa ce ati instalat Expo CLI, deschideti un terminal si introduceti comanda: `expo init my-app`

Selectati ultima optiune: “minimal, just the essentials to get you started”

Intrati in folderul proiectului si dati: `expo start`

Aceasta comanda va deschide un server HTTP care compileaza cod Javascript folosind Babel, si il serveste catre aplicatia Expo instalata pe telefon.

Daca intrati pe http://localhost:19002/ veti avea si o interfata grafica. In terminal aveti diferite comenzi.

How to run the app

Asa cum am mai spus, Expo este o platforma care ofera acces la multe tool-uri folositoare.
Un tool interesant este modalitatile prin care poti rula aplicatia:

1. **Run in web browser**, va deschide aplicatia in browser. PS: In browser s-ar putea ca unele elemente sa nu arate cum trebuie, browser-ul nu are functionalitati Native de telefon e.g. camera, giroscop, etc.
2. **Run in iOS simulator**, va deschide un simulator de iOS (pentru cei care au Mac)
3. **Run in Android device/emulator**, va deschide aplicatia pe un device Android conectat la computer, sau un emulator de Android daca aveti instalat.
4. Selectati CONNECTION: LAN. Sub se va genera un cod QR. Daca aveti instalata aplicatia Expo si sunteti pe aceeasi retea, puteti scana codul QR pentru a rula aplicatia atat pe iOS cat si pe Android.

JS Introduction

Speram ca nu este nevoie de prea multa introducere, in ideea ca s-a predat si invatat Javascript la facultate in anul 2 / 3. In esenta este un limbaj de programare destul de simplu, loosely typed, adica are tipuri de date e.g. number, string, array, object, dar acestea nu trebuie mentionate.

Recomandarile noastre ar fi folosirea Typescript, dar nu face obiectivul cursului predarea unui limbaj nou, si vom lucra exclusiv cu Javascript.

React Native Introduction

React Native este un mobile framework dezvoltat si mentinut in mare parte de Facebo… pardon, Meta.

Este folosit pentru a dezvolta aplicatii **native** de Android, iOS, Android TV, Windows, MacOS, etc, sub pretextul ca scrii o singura data cod pentru toate platformele, nu dezvolti 10 aplicatii pentru fiecare platforma in parte.

Codul de React Native pe care il vom scrie, va fi compilat si transformat in cod nativ specific platformei pe care rulam.

O aplicatie de React Native este formata din “componente”. Aceste componente pot fi “nested” la fel ca in HTML, si putem crea layouts si UI pe baza lor. Acest mod de lucru incurajeaza “code reusability”, un principiu important in programare.

Basic components

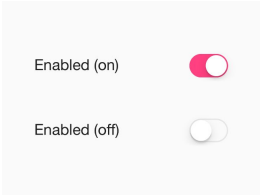
1. Cea mai basic componenta este componenta **“View”** care intr-o oarecare masura este echivalentul unui “div” din HTML. Componenta View suporta layout cu “flexbox” si are “touch handling”. Componenta “View” poate avea 0 sau mai multi copii.
2. **“Text”** este o componenta pentru a afisa text pe ecran. Mai multe componente “Text” pot fi nested daca se doreste stilizarea diferita a unei anumite parti din text.
3. **“Image”** este o componenta pentru a afisa o imagine.
4. **“TextInput”** este o componenta pentru a introduce text intr-un camp folosind tastatura device-ului. Are multe functionalitati deja implementate de ex auto-correction, auto-capitalization, placeholder, keyboard types (numeric keyboard, password, text etc.)
5. **“ScrollView”** este o componenta care va crea un “View” ce poate fi scrollable, atat orizontal cat si vertical. “ScrollView” va randa tot ce este in el pe ecran, lucru care poate duce in anumite cazuri la probleme de performanta.

Basic components

6. **“Button”.** Nu sunt prea multe de spus aici, in afara faptului ca “Button” nu este foarte stilizabil. In cazul in care avem nevoie de un buton custom, putem folosi:

7. **“TouchableOpacity”**. Asa cum implica numele, TouchableOpacity defineste o zona din ecran care poate fi apasata, si face trigger la o functie definita de noi.

8. **“Switch”** este o componenta customizabila pentru a randa un switch:



9. **“FlatList”** este o componenta asemanatoare cu “ScrollView”, dar rezolva problema de performanta mentionata. FlatList va randa pe ecran doar ce este necesar.

Acestea sunt cateva dintre componentele Basic din React Native. Pe parcurs vom introduce altele noi.

Documentatie

Ecosistemul React si React Native este foarte bine documentat. Aici vom lasa cateva referinte bune pentru cei interesati de mai multe detalii:

- https://reactnative.dev/docs/getting-started
- https://docs.expo.dev/
- https://digitalya.co/blog/how-react-native-works/