

**IMPLEMENTACIJA SERVERLESS APLIKACIJE ZA IOS UREĐAJE POMOĆU FIREBASE SERVISA****IMPLEMENTATION OF SERVERLESS APPLICATION FOR IOS DEVICES WITH FIREBASE SERVICE**

Siniša Kolarević, *Fakultet tehničkih nauka, Novi Sad*

**Oblast: RAČUNARSTVO I AUTOMATIKA**

**Kratak sadržaj** – U radu je opisano rešenje implementacije serverless aplikacije za iOS uređaje pomoću Firebase servisa. Prezentovane su i analizirane mogućnosti Firebase servisa kao Baas (backend-as-a-service) usluge. Specificirana je i implementirana mobilna aplikacija za komentarisanje proizvoda identifikovanih barkodom za iOS uređaje pomoću usluga koje nudi Firebase Baas servis.

**Ključne reči:** Mobilne aplikacije, Serverless, iOS

**Abstract** – This paper describes the solution of implementation of serverless application for iOS devices using Firebase service. Firebase services are presented and analyzed as Baas (backend-as-a-service) services. A mobile application for commenting on products identified by barcodes has been specially designed and implemented for iOS devices using the services offered by Firebase Baas service.

**Keywords:** Mobile applications, Serverless, iOS

**1. UVOD**

U ovom radu opisan je dizajn i implementacija serverless [1] iOS [2] aplikacije za komentarisanje barkodova uz korišćenje Firebase Cloud [3] platforme. Takođe, opisani su i moduli koji komuniciraju u okviru sistema, njihove međusobne veze i odgovornosti. Na najvišem nivou apstrakcije mogu se definisati 2 modula u sistemu:

- iOS aplikacija (koju koristi krajnji korisnik)
- Firebase Cloud platforma – Cloud platforma koja služi za prijem, obradu i skladištenje podataka

**1.1. Komunikacija između modula**

Tok komunikacije između modula, na najvišem nivou apstrakcije, može se opisati na sledeći način:

- Krajnji korisnik inicira određenu akciju koristeći iOS aplikaciju
- iOS aplikacija (po potrebi) kontaktira Firebase Cloud platformu radi izvršenja date akcije

**NAPOMENA:**

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Branko Milosavljević, red. prof.

- Firebase Cloud platforma prima zahtev od iOS aplikacije, procesira ga i vraća odgovor iOS aplikaciji
- iOS aplikacija prima odgovor od Firebase Cloud platforme, procesira ga i prikazuje odgovor na određenu akciju krajnjeg korisnika

**2. SPECIFIKACIJA SISTEMA**

U ovom poglavlju biće prikazana specifikacija sistema.

**2.1. Funkcionalnosti iOS aplikacije**

iOS aplikacija treba da obezbedi krajnjim korisnicima sledeće funkcionalnosti:

- Mogućnost kreiranja korisničkog naloga
- Mogućnost logovanja sa kreiranim korisničkim nalogom
- Mogućnost skeniranja proizvoljnog barkoda koristeći kameru uređaja
- Prikaz komentara za skenirani barkod
- Mogućnost komentarisanja skeniranog barkoda (sa ili bez slike)
- Mogućnost prikaza skeniranih barkodova i komentara za barkodove za korisnika
- Mogućnost podešavanja korisničkog imena
- Mogućnost podešavanja korisničke slike

**2.2. Funkcionalnosti Firebase Authentication modula**

Firestore Authentication modul treba da obezbedi sledeće funkcionalnosti:

- Mogućnost kreiranja korisničkih naloga
- Mogućnost logovanja korisnika
- Mogućnost čuvanja podataka o korisnicima
- Mogućnost izmene korisničkog imena
- Mogućnost izmene korisničke slike

**2.3. Funkcionalnosti Firebase Realtime Database modula**

Firestore Realtime Database baza podataka treba da obezbedi sledeće funkcionalnosti:

- Mogućnost čuvanja podataka o skeniranim barkodovima
- Mogućnost čuvanja podataka o kreiranim komentarima za skenirane barkodove

## 2.4. Funkcionalnosti Firebase Cloud Storage modula

*Firebase Cloud Storage* modul treba da obezbedi sledeće funkcionalnosti:

- Mogućnost čuvanja podataka o slikama za barkodove
- Mogućnost čuvanja podataka o slikama za korisnike

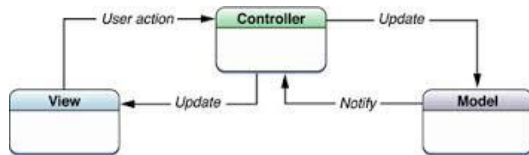
## 3. IMPLEMENTACIJA SISTEMA

U ovom poglavlju biće prikazana implementacija sistema.

### 3.1. Implementacija iOS aplikacije

*iOS* aplikacija je dizajnirana, implementirana i testirana u *Xcode* [4] razvojnom okruženju, koristeći *Swift* [5] programski jezik i *Cocoa Touch framework* [6]. *Xcode* je razvojno okruženje za razvoj aplikacija za *iOS* operativne sisteme. *Swift* je programski jezik u kome se pišu aplikacije za *iOS* operativne sisteme. *Cocoa Touch framework* je skup funkcionalnosti (skup klasa) koje dolaze sa *Xcode* razvojnim okruženjem, a koje se koriste za razvoj aplikacija za *iOS* operativne sisteme.

Pre prikaza implementacije ekrana *iOS* aplikacije biće objašnjen *MVC (Model, View, Controller)* [7] dizajn šablon koji se koristi u razvoju softvera, a posebno se koristi u razvoju aplikacija za *iOS* operativne sisteme. *MVC (Model, View, Controller)* dizajn šablon podrazumeva razdvajanje modela podataka od njihovog prikaza i rukovanja datim podacima. Na slici 3.1 prikazan je *MVC* dizajn šablon.



Slika 3.1. *MVC* dizajn šablon [7]

*iOS* aplikacija ima sledeće ekrane koji su na raspolaganju krajnjem korisniku:

- *Welcome* ekran
- *Login* ekran
- *Register* ekran
- *Scan barcode* ekran
- *No comments for barcode* ekran
- *Comments for barcode* ekran
- *Create comment for barcode* ekran
- *Feed* ekran
- *Menu* ekran
- *Housekeeping* ekran

### 3.2. Implementacija Firebase Authentication modula

*Firebase Authentication* modul obezbeđuje *backend* servise za autentifikaciju korisnika. Podržava autentifikaciju korišćenjem lozinki, brojeva telefona, kao i *Google*, *Facebook* i *Twitter* autentifikaciju.

Takođe se integriše i sa drugim *Firebase* servisima. U ovom radu korišćena je metoda autentifikacije putem *email*-a i lozinke.

### 3.3. Implementacija Firebase Realtime Database modula

*Firebase Realtime Database* modul obezbeđuje *NoSQL cloud-hosted* bazu podataka. Podaci se čuvaju u *JSON* formatu i sinhronizuju se u realnom vremenu. *Firebase Realtime Database* baza podataka se sastoji od 3 tipa entiteta: *topic*, *comment* i *feed*.

*Topic* predstavlja skenirani barcode i sastoji se od sledećih atributa:

- *code* - predstavlja barkod idnetifikator koji se očitava prilikom skeniranja barkoda

*Comment* predstavlja komentar za određeni barkod i sastoji se od sledećih atributa:

- *avatarUrl* – putanja do korisničke slike (ukoliko postoji) u okviru *Firebase Storage* modula
- *code* – barkod identifikator za koga je kreiran dati komentar
- *commentDate* – datum komentarisanja u *timestamp* formatu
- *commentPhotoUrl* – putanja do slike za komentar (ukoliko postoji) u okviru *Firebase Storage* modula
- *commentText* – tekst komentara
- *displayName* – korisničko ime
- *userId* – identifikator korisnika

*Feed* lista predstavlja listu skeniranih barkodova i komentara za barkodove za svakog korisnika i sastoji se od sledećih atributa:

- *feedType* – predstavlja tip feed elementa: Može imati 2 vrednosti: *COMMENT\_CREATED* i *TOPIC\_SCANNED*
- *timestamp* – datum kreiranja
- *topicCode* – barkod identifikator
- *commentText* – tekst komentara (ukoliko postoji). Ovo polje je popunjeno u slučaju kada je *feedType* tipa *COMMENT\_CREATED*

### 3.4. Implementacija Firebase Cloud Storage modula

*Firebase Cloud Storage* obezbeđuje skladištenje i serviranje sadržaja kao što su slike ili video zapisi. U okviru implementacije ovog rada, *Firebase Cloud Storage* se koristi za skladištenje i serviranje slika za komentare korisnike.

## 4. ZAKLJUČAK

U radu je predstavljeno rešenje implementacije *iOS* aplikacije za komentarisanje barkodova korišćenjem *Firebase Cloud* platforme.

U rešenju koje je opisano u okviru ovog rada, *iOS* aplikacija se integriše sa 3 *Firebase* modula:

- *Firebase Authentication*
- *Firebase Realtime Database*
- *Firebase Cloud Storage*

*Firebase Cloud* platforma pored navedenih modula nudi još i mnogo drugih korisnih modula, kao što su:

- *Firebase Cloud Functions*
- *Firebase ML Kit*
- *Firebase Hosting*
- *Firebase Crashlytics*
- *Firebase Performance Monitoring*
- *Firebase Google Analytics*
- *Firebase Cloud Messaging*
- *Firebase Remote Config*

Jedan od glavnih ciljeva *Firebase Cloud* platforme je brži razvoj mobilnih (i *Web*) aplikacija, bez potrebe vođenja računa o infrastrukturi. *Firebase Cloud* platforma je ustvari *BaaS (Backend-as-a-Service)* platforma.

Generalno govoreći, *Web* i mobilne aplikacije zahtevaju sličan skup serverskih funkcija uključujući čuvanje podataka, obradu podataka, notifikacije, integraciju sa društvenim mrežama itd. Svaki od navedenih serverskih servisa se mora posebno integrisati za svaku od aplikacija, što je proces koji je istovremeno dugotrajan a može biti i relativno komplikovan, u zavisnosti od servisa koji se integriše. *BaaS (Backend-as-a-Service)* platforme, odnosno, *Firebase Cloud* platforma, kao jedna od instanci *BaaS (Backend-as-a-Service)* platformi, pokušava da reši dati problem tako što na uniforman način spaja serverske funkcije sa mobilnim i *Web* aplikacijama.

*BaaS (Backend-as-a-Service)* platforme predstavljaju relativno skorašnji razvoj u okviru *Cloud Computing* oblasti računarstva, sa većinom platformi pokrenutih od 2011 godine i kasnije. Iako još uvek mlada industrija, trendovi do sada pokazuju da se usluge *BaaS (Backend-as-a-Service)* platformi sve više i više koriste, razvijaju i dobijaju na značaju.

## 5. LITERATURA

- [1] Serverless - [https://en.wikipedia.org/wiki/Serverless\\_computing](https://en.wikipedia.org/wiki/Serverless_computing) (Serverless computing allows you to build and run applications and services without thinking about servers)
- [2] iOS - <https://en.wikipedia.org/wiki/IOS> (iOS is a mobile operating system created and developed by Apple Inc. exclusively for its hardware)
- [3] *Firebase Cloud* - <https://firebase.google.com/> (Firebase helps mobile app teams succeed. Firebase is a Backend-as-a-Service - BaaS. Firebase is your server, your API and your datastore)
- [4] *Xcode* - <http://en.wikipedia.org/wiki/Xcode> (Xcode is an integrated development environment (IDE) for macOS containing a suite of software development tools developed by Apple for developing software for macOS, iOS, watchOS, and tvOS)
- [5] *Swift* - <https://developer.apple.com/swift/> (Swift is a powerful and intuitive programming language for macOS, iOS, watchOS and tvOS)
- [6] *Cocoa Touch Framework* - <https://developer.apple.com/library/archive/documentation/General/Conceptual/DevPedia-CocoaCore/Cocoa.html> (Cocoa and Cocoa Touch are the application development environments for OS X and iOS, respectively)
- [7] *MVC (Model-View-Controller) Design Pattern* - <http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller> (Model–view–controller is an architectural pattern commonly used for developing user interfaces that divides an application into three interconnected parts)

### Kratka biografija:



**Siniša Kolarević** rođen je u Vukovaru 1986. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Internet Softverske Arhitekture odbranio je 2019.god. kontakt: [sinisa.kolarevic@gmail.com](mailto:sinisa.kolarevic@gmail.com)