|  |  |  |
| --- | --- | --- |
|  | УНИВЕРЗИТЕТ У НОВОМ САДУ  **ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У НОВОМ САДУ** |  |

Петар Ћоћић

**Веб апликација за размену информација између корисника**

ЗАВРШНИ РАД

- Основне струковне студије -

Нови Сад, 2023.

|  |  |  |  |
| --- | --- | --- | --- |
| Редни број, **РБР**: | |  | |
| Идентификациони број, **ИБР**: | |  | |
| Тип документације, **ТД**: | | Монографска документација | |
| Тип записа, **ТЗ**: | | Текстуални штампани рад | |
| Врста рада, **ВР**: | | Завршни (Bachelor) рад | |
| Аутор, **АУ**: | | Име презиме | |
| Ментор, **МН**: | | Марко Марковић | |
| Наслов рада, **НР**: | | Наслов | |
| Језик публикације, **ЈП**: | | Српски | |
| Језик извода, **ЈИ**: | | Српски | |
| Земља публиковања, **ЗП**: | | Република Србија | |
| Уже географско подручје, **УГП**: | | Војводина | |
| Година, **ГО**: | | Година  Ауторски репринт  Нови Сад, трг Доситеја Обрадовића 6  6/ 52/ 0/ 10/ 36/ 0/ 0  Електротехничко и рачунарско инжењерство | |
| Издавач, **ИЗ**: | | Ауторски репринт | |
| Место и адреса, **МА**: | | Нови Сад, Трг Доситеја Обрадовића 6 | |
| Физички опис рада, **ФО**: (поглавља/страна/цитата/табела/слика/графика/прилога) | | .../.../.../.../.../.../... | |
| Научна област, **НО**: | | електротехничко и рачунарско инжењерство | |
| Научна дисциплина, **НД**: | |  | |
| Предметна одредница/Кључне речи, **ПО**: | | Кључне речи | |
| **УДК** | |  | |
| Чува се, **ЧУ**: | | Библиотека Факултета техничких наука, Трг Д. Обрадовића 6, Нови Сад | |
| Важна напомена, **ВН**: | |  | |
| Извод, **ИЗ**: | | Сажетак | |
| Датум прихватања теме, **ДП**: | |  | |
| Датум одбране, **ДО**: | |  | |
| Чланови комисије, **КО**: | Председник: | Председник |
|  | Члан: | Члан | Потпис ментора |
|  | Члан, ментор: | др Марко Марковић, доцент |  |

Образац **Q2.НА.04-05** - Издање 1

|  |  |  |  |
| --- | --- | --- | --- |
| Accession number, **ANO**: | |  | |
| Identification number, **INO**: | |  | |
| Document type, **DT**: | | Monographic publication | |
| Type of record, **TR**: | | Textual printed material | |
| Contents code, **CC**: | | Bachelor Thesis | |
| Author, **AU**: | | First and Last name | |
| Mentor, **MN**: | | Marko Marković | |
| Title, **TI**: | | Title | |
| Language of text, **LT**: | | Serbian | |
| Language of abstract, **LA**: | | Serbian | |
| Country of publication, **CP**: | | Republic of Serbia | |
| Locality of publication, **LP**: | | Vojvodina | |
| Publication year, **PY**: | | Year | |
| Publisher, **PB**: | | Author’s reprint | |
| Publication place, **PP**: | | Faculty of Technical Sciences, Trg Dositeja Obradovića 6, Novi Sad | |
| Physical description, **PD**: (chapters/pages/ref./tables/pictures/graphs/appendixes) | | .../.../.../.../.../.../... | |
| Scientific field, **SF**: | | electrical and computer engineering | |
| Scientific discipline, **SD**: | |  | |
| Subject/Key words, **S**/**KW**: | | Keywords | |
| **UC** | |  | |
| Holding data, **HD**: | | The Library of Faculty of Technical Sciences, Novi Sad, Serbia | |
| Note, **N**: | |  | |
| Abstract, **AB**: | | Abstract | |
| Accepted by the Scientific Board on, **ASB**: | |  | |
| Defended on, **DE**: | |  | |
| Defended Board, **DB**: | President: | President |
|  | Member: | Member | Menthor's sign |
|  | Member, Mentor: | Marko Marković, assist. prof. |  |

Obrazac **Q2.НА.04-05** - Izdanje 1

*(Податке уноси предметни наставник - ментор)*

| Студијски програм: | Назив студијског програма | | |
| --- | --- | --- | --- |
| Студент: | Име Презиме | Број индекса: | Бр. индекса |
| Степен и врста студија: | Основне академске/струковне студије | | |
| Област: |  | | |
| Ментор: | Марко Марковић | | |
| НА ОСНОВУ ПОДНЕТЕ ПРИЈАВЕ, ПРИЛОЖЕНЕ ДОКУМЕНТАЦИЈЕ И ОДРЕДБИ СТАТУТА ФАКУЛТЕТА  ИЗДАЈЕ СЕ ЗАДАТАК ЗА ЗАВРШНИ РАД, СА СЛЕДЕЋИМ ЕЛЕМЕНТИМА:   * проблем – тема рада; * начин решавања проблема и начин практичне провере резултата рада, ако је таква провера неопходна; | | | |

**НАСЛОВ ЗАВРШНOГ РАДА:**

|  |
| --- |
| Наслов |

**ТЕКСТ ЗАДАТКА:**

|  |
| --- |
| 1. Анализирати стање у области.  2. Израдити спецификацију захтева софтверског решења.  3. Израдити спецификацију дизајна софтверског решења.  4. Имплементирати софтверско решење према израђеној спецификацији.  5. Тестирати имплементирано софтверско решење.  6. Документовати (1), (2), (3), (4) и (5). |

|  |  |
| --- | --- |
| Руководилац студијског програма: | Ментор рада: |
|  |  |

САДРЖАЈ

[1. УВОД 1](#_Toc58685298)

[2. ПРЕГЛЕД СЛИЧНИХ АПЛИКАЦИЈА И КОРИШЋЕНИХ СОФТВЕРСКИХ ТЕХНОЛОГИЈА 2](#_Toc58685299)

[2.1 Преглед сличних апликација 2](#_Toc58685300)

[2.2 Преглед коришћених софтверских технологија 7](#_Toc58685301)

[3. СПЕЦИФИКАЦИЈА ЗАХТЕВА 9](#_Toc58685302)

[3.1 Функционални захтеви 9](#_Toc58685303)

[3.2 Нефункционални захтеви 17](#_Toc58685304)

[4. СПЕЦИФИКАЦИЈА ДИЗАЈНА 18](#_Toc58685305)

[4.1 Модел података 18](#_Toc58685306)

[4.2 Архитектура система 19](#_Toc58685307)

[5. ИМПЛЕМЕНТАЦИЈА СИСТЕМА 20](#_Toc58685308)

5.1 Клијентски део апликације

5.2 Серверски део апликације

[6. ДЕМОНСТРАЦИЈА 34](#_Toc58685309)

[7. ЗАКЉУЧАК 35](#_Toc58685310)

[8. ЛИТЕРАТУРА 36](#_Toc58685311)

# УВОД

Проналаском информационих технологија, отворене су многобројне могућности које савременом човеку знатно олакшавају живот. Рачунари у неким задацима могу у потпуности да замене човека, док у неким случајевима могу бити само помоћ човеку како би лакше обавио неки задатак. Од тренутка када је World Wide Web [1] пронађен, наше могућности су се прошириле и отворени су путеви за решавање комплексних проблема.

Осврнимо се на период док World Wide Web није постојао и начин на који су људи размењивали информације.

Писана пошта је представљала основни начин комуникације при чему би људи писали писма, стављали та писма у коверте, залепили маркицу и послали је поштом. Ово је био спорији начин комуникације у поређењу са данашњом електронском поштом, али је такође омогућавао људима да размењују мисли и информације на даљину.

Телефонирање је још један основни вид комуникације који и дан данас постоји и активно се користи. То је вид комуникације који се користи за директне разговоре између људи. Мобилни телефони су такође постојали, али нису били тако распрострањени и моћни као што су данашњи паметни телефони.

Савремени човек је доживео револуцију захваљујући информационим технологијама. У последњим деценијама, свет се трансформисао на начине који су били незамисливи до пре само неколико година. На пример, према недавном истраживању, више од 60% светске популације сада има приступ интернету, што отвара врата безбројним могућностима. Овај развој информационих технологија није променио само начине комуникације између људи, већ је променио и начин на који обављамо послове.

У овом раду је представљена веб апликација која омогућава размењивање информација између људи широм света под називом Твитер Клон (енг. Twitter Clone). Ово софтверско решење има за циљ да укључи људе широм света у дискусију на неку тему, да би се видело колики проценат људи дели исто мишљење. Сваки корисник може покренути било коју тему за дискусију при чему она мора бити прикладна за све узрасте.

У наставку рада, поглавља су организована на следећи начин: друго поглавље пружа увид у сродне апликације које су послужиле као инспирација за развој веб апликације Twitter Clone и п редстављене су технологије које су коришћене за развој, у трећем поглављу је дата спецификација функционалних и нефункционалних захтева ове апликације, четврто поглавље садржи спецификацију дизајна апликације, пето поглавље пружа увид у имплементацију ове веб апликације, шесто поглавље демонстрира рад апликације, а седмо поглавље садржи закључак.

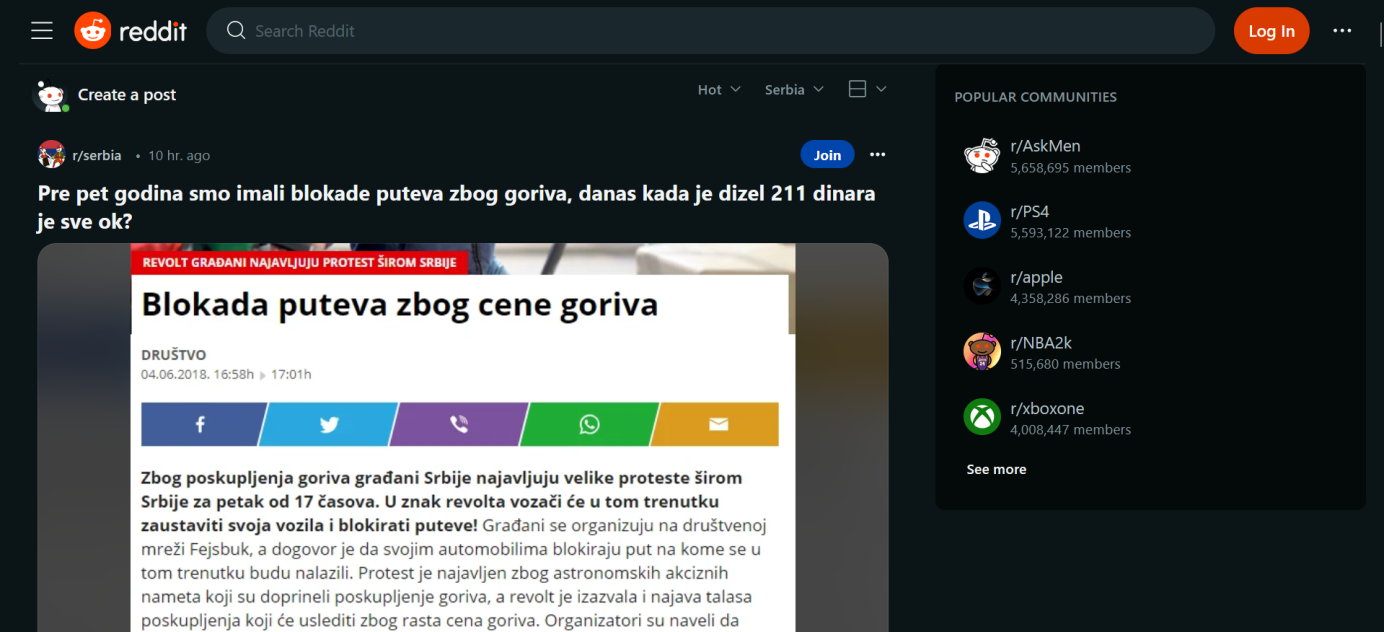
# ПРЕГЛЕД СЛИЧНИХ АПЛИКАЦИЈА И КОРИШЋЕНИХ СОФТВЕРСКИХ ТЕХНОЛОГИЈА

У овом поглављу су наведена и објашњена сродна решења за које се сматра да су донела револуцију у размени информација, а потом је дат преглед софтверских технологија које су коришћене за израду апликације Twitter Clone.

## Преглед сличних апликација

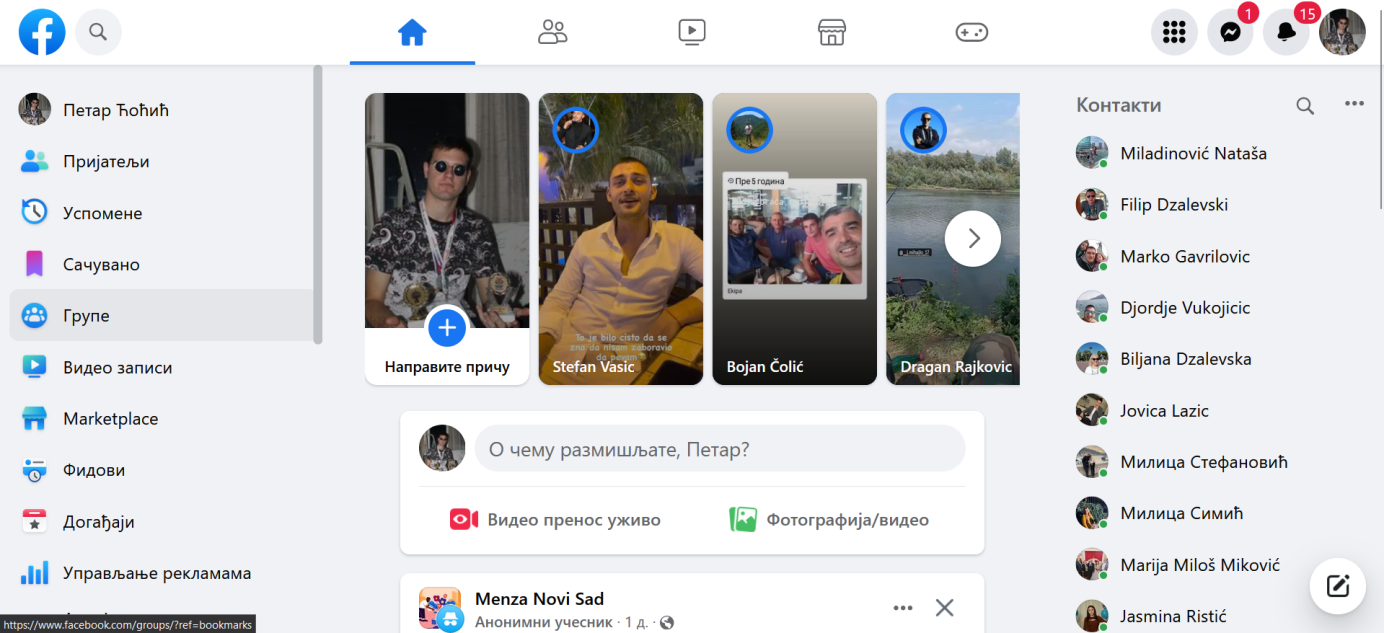
Овај одељак доноси преглед апликација које су иницијатива приликом имплементације Twitter Clone апликације. Неке од таквих апликација су:

* Reddit [2]
* Facebook [3]
* LinkedIn [4]
* Instagram [5]

Reddit апликација је популарна друштвена мрежа чија је основна замисао на креирању заједница, које су познате као енг. „subreddits“, које се фокусирају на различите теме, интересе и дискусије. Корисници могу постављати питања, делити информације, учествовати у расправама и објављивати садржај у тим заједницама. Reddit је познат по анонимности корисника, а гласови (енг. Upvote, Downvote) утичу на видљивост садржаја. Reddit има широк спектар заједница, од општих интереса до специјализованих подручја, укључујући оне на српском језику, попут r/serbia, r/programiranje. Ове заједнице нуде простор за размену искустава, мишљења и ресурса. Изглед Reddit друштвене мреже је приказан на слици 1.  
  


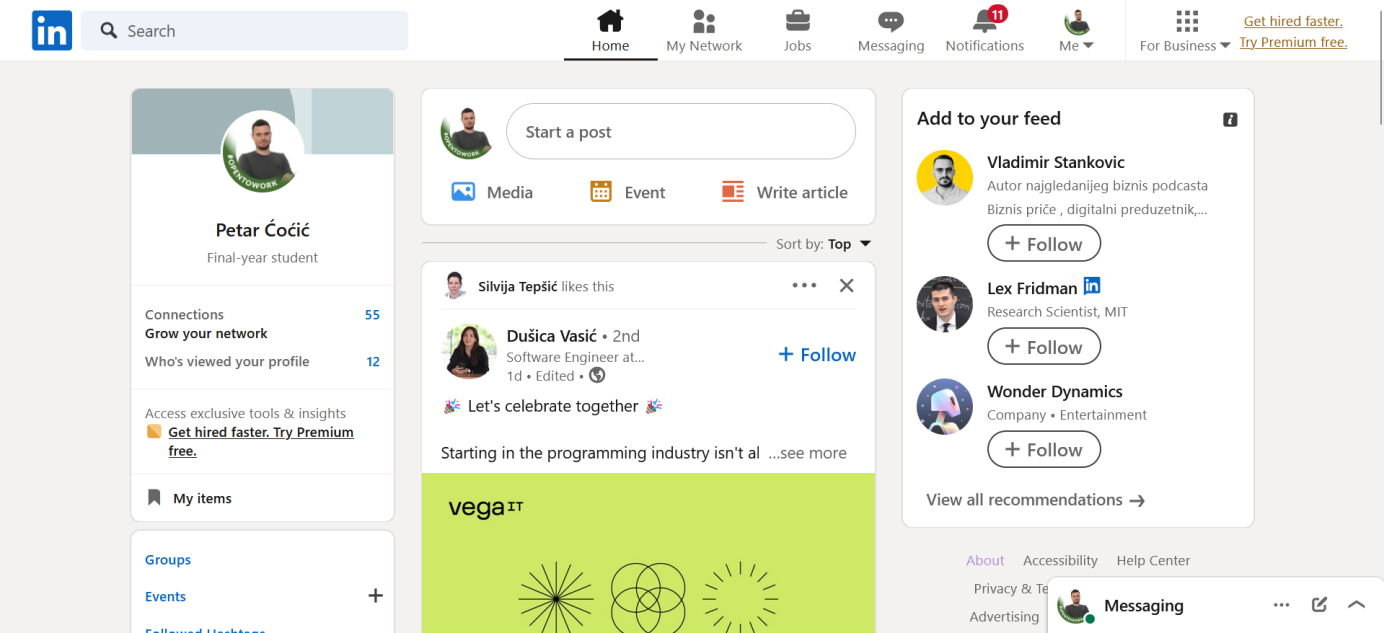
**Слика 1 – Почетна страница Reddit друштвене мреже**

Facebook је друштвена мрежа која повезује људе широм света. Ова платформа омогућава корисницима да креирају профиле, повежу се са својим пријатељима и породицом и деле своје животне тренутке, фотографије и мисли. Фејсбук нуди различите опције за интеракцију, укључујући лајковање, коментаре и дељење садржаја. Такође, корисници се могу придружити различитим групама и заједницама које се баве интересантним темама. Facebook је такође постао платформа за пословно повезивање, омогућавајући компанијама да промовишу своје производе и услуге. С обзиром на своју популарност, Facebook је постао кључни део свакодневног живота за многе људе, пружајући могућност да остану повезани са светом и прате најновије вести, догађаје и трендове. На слици 2 је приказана почетна страница апликације Facebook.



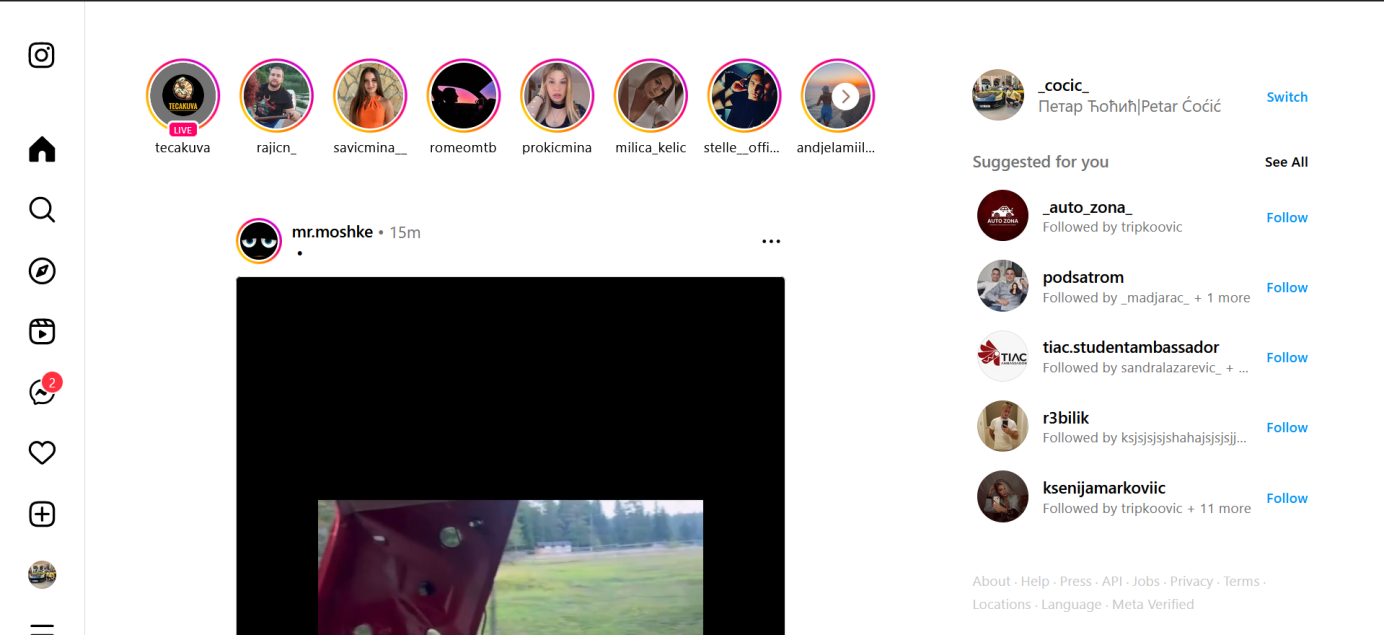
**Слика 2 – Почетна страница Facebook друштвене мреже**

LinkedIn је професионална друштвена мрежа која омогућава повезивање запослених из различитих индустрија и области. Корисници могу креирати своје профиле, навести своје образовање, радно искуство и вештине, а такође истраживати могућности за запошљавање и пословно умрежавање. Ова платформа омогућава објављивање професионалних садржаја, као што су чланци и виђења, који помажу у изградњи личног бренда и деле сазнања из одређених индустрија. LinkedIn такође служи за проналажење потенцијалних пословних кандидата, регрутовање и стварање пословних прилика. Коришћење LinkedIn-а може помоћи у развоју каријере, обавештавању о новим трендовима у индустрији и умрежавању са другим професионалцима. Ова платформа је незамењива за области како запошљавања, тако и за лични и професионални развој. На слици 3 је приказана почетна страница апликације LinkedIn.



**Слика 3 – Почетна страница LinkedIn друштвене мреже**

Instagram је друштвена мрежа која је фокусирана на дељење фотографија и видео снимака. Корисници могу креирати своје профиле, објављивати садржај, и пратити друге кориснике који су им интересантни. Основна идеја Instagramа је дељење животних тренутака и искуства преко слика. Ова платформа омогућава корисницима да примене филтре и ефекте на своје фотографије и видео снимке како би их учинили креативнијим и занимљивијим. Такође, Instagram подржава размену порука и сторија, што корисницима омогућава да деле садржај који нестане након 24 сата. Ова платформа се користи за разне сврхе, укључујући лично дељење животних тренутака, проналажење инспирације за путовања, моду, исхрану, и развој личних или брендираних профила. Instagram је стекао огромну популарност широм света и стално се развија у оквиру социјалних мрежа. На слици 4 је приказан изглед почетне странице Instagram.



**Слика 4 – Почетна страница Instagram друштвене мреже**

## Преглед коришћених софтверских технологија

## Софтверске технологије које су коришћене у изради овог софтверског решења су следеће:

## Docker [6]

## Docker Compose [7]

## REST API [8]

## npm **(Node.js [9] Package Manager)** [10]

## Go Modules [11]

## Message Worker [12]

* Firebase Cloud Messaging [13]

## Elastic Search [14]

## NATS Message Broker [15]

## Docker је популарна платформа за контејнеризацију која је донела револуционаран начин на који се развијају, пакују и испоручују апликације. Ова технологија омогућава програмерима и инжењерима да изолују делове апликација и све њихове зависности у контејнере (енг. Containers) [16].

## Docker Compose је популаран алат који омогућава дефинисање и извршавање вишекомпонентних апликација у Docker окружењу на једноставан и конзистентан начин. Овај алат олакшава управљање контејнерима и њиховим зависностима, чиме се унапређује процес развоја и тестирања апликација. Docker Compose користи YAML [17] фајл за дефинисање сервиса, мрежа и складишта, чиме се омогућава једноставна репродукција комплексних окружења.

Rest Api (енг. Representational State Transfer Application Programming Interface) представља архитектуру за дизајнирање и изградњу веб сервиса који комуницирају преко HTTP [18] протокола. Ова архитектура се фокусира на ресурсе, који су представљени као URL-ови, и операције које се извршавају на њима. REST API пружа начин да се лако комуницира са апликацијом преко мреже, било да се ради о захтевима за добијање, додавање, ажурирање или брисање података (енг. CRUD operations [19]).

npm (Node.js Управљач пакетима) представља стандардни алат за управљање зависностима и библиотекама у развоју апликација користећи Node.js платформу. Овај механизам омогућава програмерима да лако инсталирају, управљају и ажурирају различите библиотеке и пакете који су неопходни за развој њихових апликација. Са npm-ом, развојници могу брзо и једноставно додавати библиотеке и зависности у своје пројекте користећи једноставне команде. Овај алат такође омогућава креирање и управљање "package.json" [20] фајловима, који садрже информације о зависностима и конфигурацији пројекта. Ово помаже у управљању верзијама библиотека и осигурава смислену структуру пројекта.

Go Modules представљају систем у Go [21] програмском језику који омогућава управљање зависностима и верзијама библиотека у развоју Go апликација. Овај систем је уведен као начин да се ефикасније и боље управља зависностима у Go пројектима и да се осигура репродукција и сингуларност зависности у различитим окружењима.

Message Worker је концепт у развоју софтвера који се односи на компоненту или модул који обрађује асинхроне поруке у апликацији. Ова технологија омогућава апликацијама да извршавају задатке или обраде поруке у позадини, без блокирања корисничког интерфејса или главног тока извршавања.

Firebase Cloud Messaging (FCM) је услуга компаније Гугл (енг. Google [22]) која омогућава размену порука између сервера и клијентских апликација на различитим платформама. Ова услуга је саставни део Firebase платформе, која пружа различите алате и услуге за развој апликација.

Elastic search је отворено и дистрибуирано складиште података за анализу и претрагу текстуалних и структурираних података. Ова технологија развијена је на основу Apache Lucene пројекта и представља моћан алат за претрагу и анализу великих обима података.

NATS Message Broker је лак, брз и отворени изворни систем за објављивање порука и директну обраду порука. Он је дизајниран да буде једноставан и ефикасан, идеалан за изградњу скалабилних апликација које комуницирају у реалном времену.

# СПЕЦИФИКАЦИЈА ЗАХТЕВА

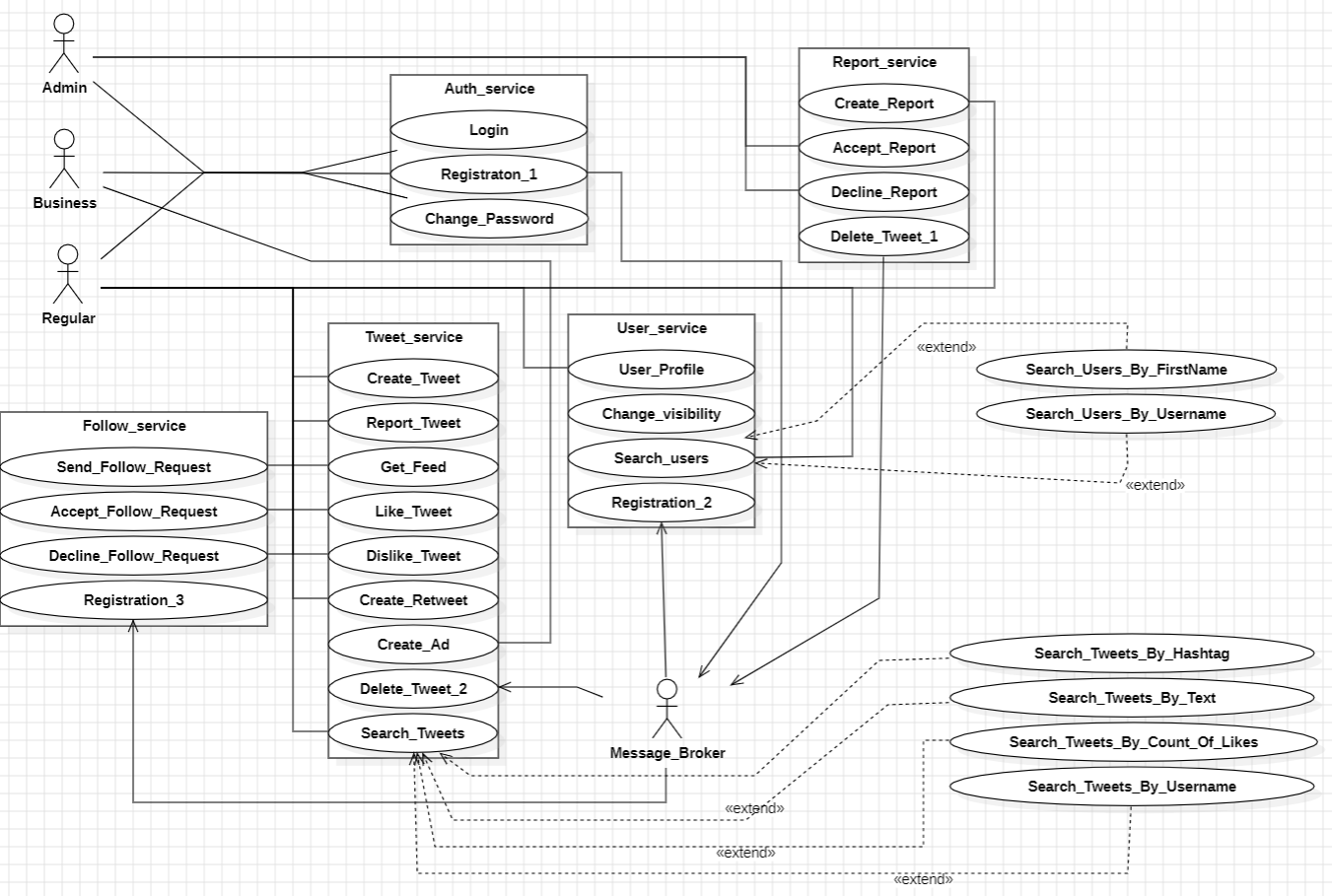
У овом поглављу су наведени захтеви које је било неопходно имплементирати за комплетан рад апликације. Најпре су наведени функционални захтеви, а затим су наведени и нефункционални захтеви.

## Функционални захтеви

Процеси унутар система који морају обезбедити корисницима одређене функционалности и могућности су приказани помоћу дијаграма случајева коришћења (Use-case diagram[23]). У овом раду су објашњене функционалности за следеће сервисе:

1. Auth service
2. Report service
3. User service

На слици 5 је приказан дијаграм случајева коришћења.



**Слика 5 – дијаграм случајева коришћења апликације Twitter-Clone**

У наставку су појединачно објашњени случајеви коришћења сваког горе наведеног сервиса сукцесивно.

У табели 1 објашњен је случај коришћења „Регистрација корисника“.

|  |  |
| --- | --- |
| **Случај коришћења** | **Регистрација корисника** |
| Учесници | Корисник |
| Предуслови | 1. Корисник не постоји у систему |
| Опис | 1. Корисник кликом на дугме „Registration“ приступа страници за регистровање новог корисника 2. Корисник редом селектује поља за унос и уписује потребне информације 3. Корисник потврђује регистрацију |
| Изузеци | * Нису попуњена сва поља * Корисничко име већ постоји у систему |
| Резултати | Креиран је кориснички налог |

**Табела 1 – Случај коришћења „Регистрација корисника“**

У табели 2 објашњен је случај коришћења „Пријављивање корисника“.

|  |  |
| --- | --- |
| **Случај коришћења** | **Пријављивање корисника** |
| Учесници | Корисник |
| Предуслови | 1. Корисник мора бити регистрован на систем |
| Опис | 1. Корисник кликом на дугме „Login“ приступа страници за пријављивање корисника 2. Корисник селектује поље за унос корисничког имена и уноси своје корисничко име 3. Корисник селектује поље за унос лозинке и уноси своју лозинку 4. Корисник потврђује пријављивање на систем |
| Изузеци | * Корисник није регистрован на систем * Лозинка није валидна |
| Резултати | Корисник је пријављен на апликацију |

**Табела 2 – Случај коришћења „Пријављивање корисника“**

У табели 3 објашњен је случај коришћења „Промена лозинке“.

|  |  |
| --- | --- |
| **Случај коришћења** | **Промена лозинке** |
| Учесници | Корисник |
| Предуслови | 1. Корисник мора бити регистрован на систем 2. Корисник мора бити пријављен на систем |
| Опис | 1. Корисник кликом на дугме „Change password“ приступа страници за промену лозинке 2. Корисник селектује поље за унос тренутне лозинке и уноси тренутну лозинку 3. Корисник селектује поље за унос нове лозинке и уноси нову лозинку 4. Корисник селектује поље за поновно уношење лозинке, и поново уноси нову лозинку 5. Корисник потврђује промену лозинке |
| Изузеци | * Корисник не постоји у систему * Корисник није пријављен на систем |
| Резултати | Корисник је успешно променио своју лозинку |

**Табела 3 – Случај коришћења „Промена лозинке“**

У табели 4 објашњен је случај коришћења „Пријављивање неприкладне објаве“.

|  |  |
| --- | --- |
| **Случај коришћења** | **Пријављивање неприкладне објаве** |
| Учесници | Корисник |
| Предуслови | 1. Корисник мора бити пријављен на систем 2. Корисник мора пратити профил помоћу кога је креирана објава |
| Опис | 1. Корисник кликом на дугме „Report“ приступа страници за пријављивање објаве 2. Корисник одабира један од понуђених разлога пријаве објаве 3. Корисник потврђује пријаву објаве |
| Изузеци | * Корисник није пријављен на систем * Корисник не прати профил са кога је креирана објава |
| Резултати | Корисник је успешно пријавио неприкладну објаву |

**Табела 4 – Случај коришћења „Пријављивање неприкладне објаве“**

У табели 5 објашњен је случај коришћења „Прихватање поднесене пријаве“.

|  |  |
| --- | --- |
| **Случај коришћења** | **Прихватање поднесене пријаве** |
| Учесници | Админ |
| Предуслови | 1. Админ мора бити предефинисан у систему 2. Админ мора бити пријављен на систем |
| Опис | 1. Корисник кликом на дугме „Reports“ приступа страници за излиставање свих пријава 2. Корисник привхата поднесену пријаву |
| Изузеци | * Админ није предефинисан у систему * Админ није пријављен на систем |
| Резултати | Админ је успешно прихватио поднесену пријаву |

**Табела 5 – Случај коришћења „Прихватање поднесене пријаве“**

У табели 6 објашњен је случај коришћења „Одбијање поднесене пријаве“.

|  |  |
| --- | --- |
| **Случај коришћења** | **Одбијање поднесене пријаве** |
| Учесници | Админ |
| Предуслови | 1. Админ мора бити предефинисан у систему 2. Админ мора бити пријављен на систем |
| Опис | 1. Корисник кликом на дугме „Reports“ приступа страници за излиставање свих пријава 2. Корисник одбија поднесену пријаву |
| Изузеци | * Админ није предефинисан у систему * Админ није пријављен на систем |
| Резултати | Админ је успешно одбио поднесену пријаву |

**Табела 6 – Случај коришћења „Одбијање поднесене пријаве“**

У табели 7 објашњен је случај коришћења „Брисање објаве“.

|  |  |
| --- | --- |
| **Случај коришћења** | **Брисање објаве** |
| Учесници | Админ |
| Предуслови | 1. Админ мора бити предефинисан у систему 2. Админ мора бити пријављен на систем |
| Опис | 1. Админ прихвата трећу по реду пријаву за одређену објаву |
| Изузеци | * Админ није предефинисан у систему * Админ није пријављен на систем |
| Резултати | Објава за коју је три пута прихваћена пријава се брише |

**Табела 7 – Случај коришћења „Брисање објаве“**

У табели 8 објашњен је случај коришћења „Кориснички профил“.

|  |  |
| --- | --- |
| **Случај коришћења** | **Кориснички профил** |
| Учесници | Корисник |
| Предуслови | 1. Корисник мора бити пријављен на систем |
| Опис | 1. Корисник кликом на дугме „My Profile“ приступа страници са приказом персоналних информација |
| Изузеци | * Корисник није пријављен на систем |
| Резултати | Корисник је успешно приступио персоналним информацијама |

**Табела 8 – Случај коришћења „Кориснички профил“**

У табели 9 објашњен је случај коришћења „Промена видљивости профила“.

|  |  |
| --- | --- |
| **Случај коришћења** | **Промена видљивости профила** |
| Учесници | Корисник |
| Предуслови | 1. Корисник мора бити пријављен на систем |
| Опис | 1. Корисник кликом на дугме „My profile“ приступа страници за приказ персоналних информација 2. Корисник кликом на компоненту слајдер (енг. Toggle Switch [24]) мења видљивост профила на приватан/јаван |
| Изузеци | * Корисник није пријављен на систем |
| Резултати | Корисник је успешно променио видљивост свог профила |

**Табела 9 – Случај коришћења „Промена видљивости профила“**

У табели 10 објашњен је случај коришћења „Претрага корисника по имену“.

|  |  |
| --- | --- |
| **Случај коришћења** | **Претрага корисника по имену** |
| Учесници | Корисник |
| Предуслови | 1. Корисник мора бити пријављен на систем |
| Опис | 1. Корисник кликом на дугме „Search People“ приступа страници за претрагу корисника 2. Корисник одабира врсту претраге из опадајућег менија и селектује опцију„FirstName“ 3. Коринсик потврђује претрагу |
| Изузеци | * Корисник није пријављен на систем |
| Резултати | Корисник је претражио корисника на основу унетог имена |

**Табела 10 – Случај коришћења „Претрага корисника по имену“**

У табели 11 објашњен је случај коришћења „Претрага корисника по корисничком имену“.

|  |  |
| --- | --- |
| **Случај коришћења** | **Претрага корисника по корисничком имену** |
| Учесници | Корисник |
| Предуслови | 1. Корисник мора бити пријављен на систем |
| Опис | 1. Корисник кликом на дугме „Search People“ приступа страници за претрагу корисника 2. Корисник одабира врсту претраге из опадајућег менија и селектује опцију „Username“ 3. Коринсик потврђује претрагу |
| Изузеци | * Корисник није пријављен на систем |
| Резултати | Корисник је претражио корисника на основу унетог имена |

**Табела 11 – Случај коришћења „Претрага корисника по корисничком имену“**

У табели 12 објашњен је случај коришћења „Претрага објаве по тексту“.

|  |  |
| --- | --- |
| **Случај коришћења** | **Претрага објаве по тексту** |
| Учесници | Корисник |
| Предуслови | 1. Корисник мора бити пријављен на систем |
| Опис | 1. Корисник кликом на дугме „Search Tweets“ приступа страници за претрагу објава 2. Корисник одабира врсту претраге из опадајућег менија и селектује опцију „Теxt“ 3. Коринсик потврђује претрагу |
| Изузеци | * Корисник није пријављен на систем |
| Резултати | Корисник је претражио објаве на основу унетог текста објаве |

**Табела 12 – Случај коришћења „Претрага објаве по тексту“**

У табели 13 објашњен је случај коришћења „Претрага објаве по хештагу“.

|  |  |
| --- | --- |
| **Случај коришћења** | **Претрага објаве по хештагу (енг. Hashtag[25])** |
| Учесници | Корисник |
| Предуслови | 1. Корисник мора бити пријављен на систем |
| Опис | 1. Корисник кликом на дугме „Search Tweets“ приступа страници за претрагу објава 2. Корисник одабира врсту претраге из опадајућег менија и селектује опцију „Hashtag“ 3. Коринсик потврђује претрагу |
| Изузеци | * Корисник није пријављен на систем |
| Резултати | Корисник је претражио објаве на основу унетог хештага објаве |

**Табела 13 – Случај коришћења „Претрага објаве по хештагу“**

У табели 14 објашњен је случај коришћења „Претрага објаве по корисничком имену“.

|  |  |
| --- | --- |
| **Случај коришћења** | **Претрага објаве по корисничком имену** |
| Учесници | Корисник |
| Предуслови | 1. Корисник мора бити пријављен на систем |
| Опис | 1. Корисник кликом на дугме „Search Tweets“ приступа страници за претрагу објава 2. Корисник одабира врсту претраге из опадајућег менија и селектује опцију „Username“ 3. Коринсик потврђује претрагу |
| Изузеци | * Корисник није пријављен на систем |
| Резултати | Корисник је претражио објаве на основу унетог корисничког имена корисника који је креирао објаву |

**Табела 14 – Случај коришћења „Претрага објаве по корисничком имену“**

У табели 15 објашњен је случај коришћења „Претрага објаве по броју лајкова“.

|  |  |
| --- | --- |
| **Случај коришћења** | **Претрага објаве по броју лајкова** |
| Учесници | Корисник |
| Предуслови | 1. Корисник мора бити пријављен на систем |
| Опис | 1. Корисник кликом на дугме „Search Tweets“ приступа страници за претрагу објава 2. Корисник одабира врсту претраге из опадајућег менија и селектује опцију „Number of likes“ 3. Коринсик потврђује претрагу |
| Изузеци | * Корисник није пријављен на систем |
| Резултати | Корисник је претражио објаве на унетог текста објаве |

**Табела 15 – Случај коришћења „Претрага објаве по броју лајкова“**

## Нефункционални захтеви

Нефункционални захтеви који су имплементирани и неопходни за апликацију су следећи:

1. Комуникација микросервиса (енг. Microservices [26]) сервиса помоћу NATS Message Brokera
2. Saga pattern [27]
3. Circuit Breaker [28]
4. Пуш нотификације (енг. Push Notifications [29])

Комуникацијом између миркосервиса помоћу NATS Message Brokera омогућава самосталних микросервиса да размењују податке између себе уз помоћ порука.

Конкретно у раду је коришћен Saga Pattern који се ослања на функције NATS Message Brokera, при чему омогућава чување информација унутар апликације на више места без директне активности корисника. Корисник није ни свестан да је акција покренута, већ апликације све обавља сама у позадини.

Circuit Breaker је механизам који дужности да заштити апликацију од грешака, при чему он функционише као и обичан прекидач који може бити отвореn, полуотворен или затворен. Отворено стање не пропушта захтеве што значи да се грешке константно појављују. Полуотворено стање пропушта један захтев у одређеном временском интервалу да провери да ли ће поново доћи до истих грешака. Затворено стање пропушта захтеве у једном смеру, што доводи до закључка да грешке не постоје и да се комуникација одвија непрекидно.

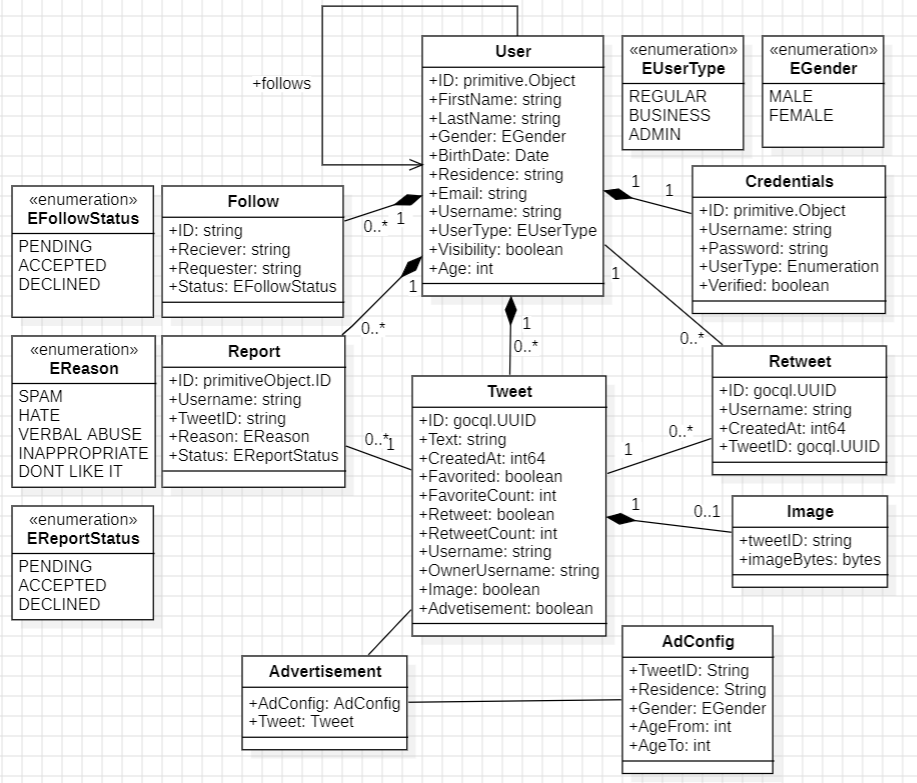
Push Notifications је механизам који побољшава корисничко искуство приликом коришћења неке веб или мобилне апликације где корисник бива обавештен у било ком тренутку о догађајима које се дешавају унутар апликације а на које се он претходно пријавио.

# СПЕЦИФИКАЦИЈА ДИЗАЈНА

У овом поглављу су приказани модел података којима апликација барата и архитектура система.

## Модел података

У овом одељку на слици 6 приказан је модел података апликације Twitter Clone помоћу UML класног дијаграма (енг. Class Diagram [30]).

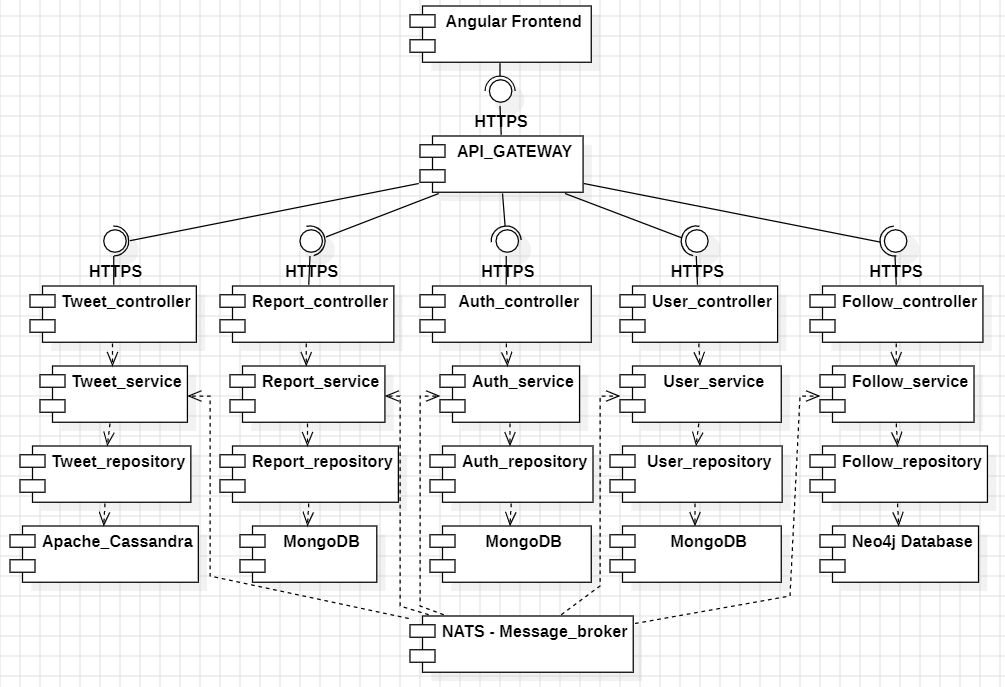


**Слика 6 – Дијаграм класа апликације Twitter Clone**

Класе које су приказане на слици 6 су по истом шаблону чуване и у базама података.

## Архитектура система

Архитектура овог система се заснива на комуникацији између клијентског дела апликације са серверским делом апликације. Комуникација се остварује помоћу HTTPS [32] протокола. Даље серверски део апликације комуницира са базама података (енг. Database [33]). На слици 7 је приказан дијаграм компоненти који сликовито описује архитектуру система.



**Слика 7 – Дијаграм компоненти апликације Twitter Clone**

# ИМПЛЕМЕНТАЦИЈА СИСТЕМА

У овом поглављу је представљена имплементација апликације за размену информација и мишљења између корисника. Апликација се састоји од клијентског дела и серверског дела који су објашњени у наставку.

## Клијентски део апликације Twitter Clone

Клијентски део апликације развијен је помоћу радног оквира који се назива Angular. Састоји се од низа компоненти које упарене заједно представљају кориснички интерфејс који омогућава кориснику да комуницира са серверским делом апликације.

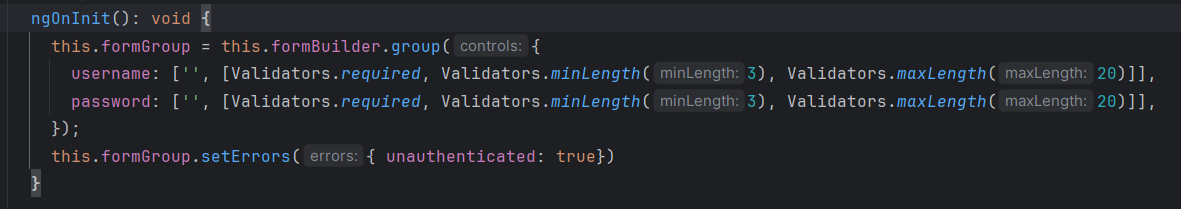
Приликом покретања апликације, приказује се страница за пријављивање корисника.

На листингу 1 је приказана Логин компонента за пријављивање корисника.



**Листинг 1 – Логин компонента**

Класа LoginComponent имплементира методу OnInit() чија је улога да се изврши при генерисању компоненте, што је у овом случају иницијализовање FormGroup објекта као што је приказано на листингу 2.



**Листинг 2 – Иницијализовање FormGroup објекта**

Метода onSubmit() има задатак да пошаље HTTP захтев ка серверском делу апликације, при чему се на самом почетку методе иницијализује објекат LoginDTO који представља податке који ће бити послати у телу захтева. Затим се прикупљају подаци из форме на основу кључа који је задат за сваки Input елемент унутар форме и додељују се одређеном атрибуту унутар LoginDTO објекта. Слање HTTP захтева се остварује позивањем методе subscribe(loginDTO) као што је приказано на листингу 3.



**Листинг 3 – Слање креденцијала за пријављивање на систем**

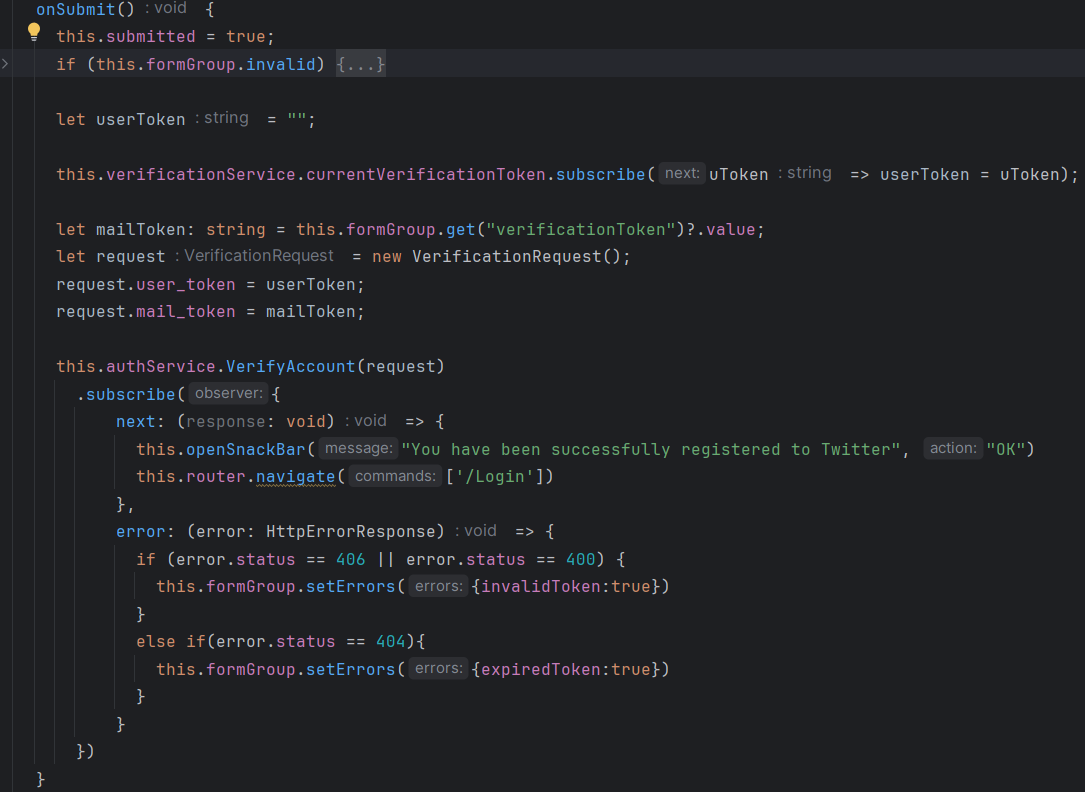
Приликом успешног пристизања одговора од серверског дела апликације, одговор је текстуалног типа који представља JWT [34] токен који нам служи за проверавање приступа одређеним деловима апликације. Токен се складишти унутар дела претраживача који се назива LocalStorage под кључем „authToken“. Уколико је корисникова улога Админ, корисник бива редиректован на страницу за излиставање свих пријављених објава. Свака друга улога бива редиректована на почетну страницу апликације.

Када корисник жели да се региструје, потребно је да приступи RegisterRegularComponent и да унесе своје информације. На листингу 4 је приказана метода onSubmit()унутар које се прикупљају унете вредности унутар форме и додељују се објекту User који се шаље серверском делу апликације.

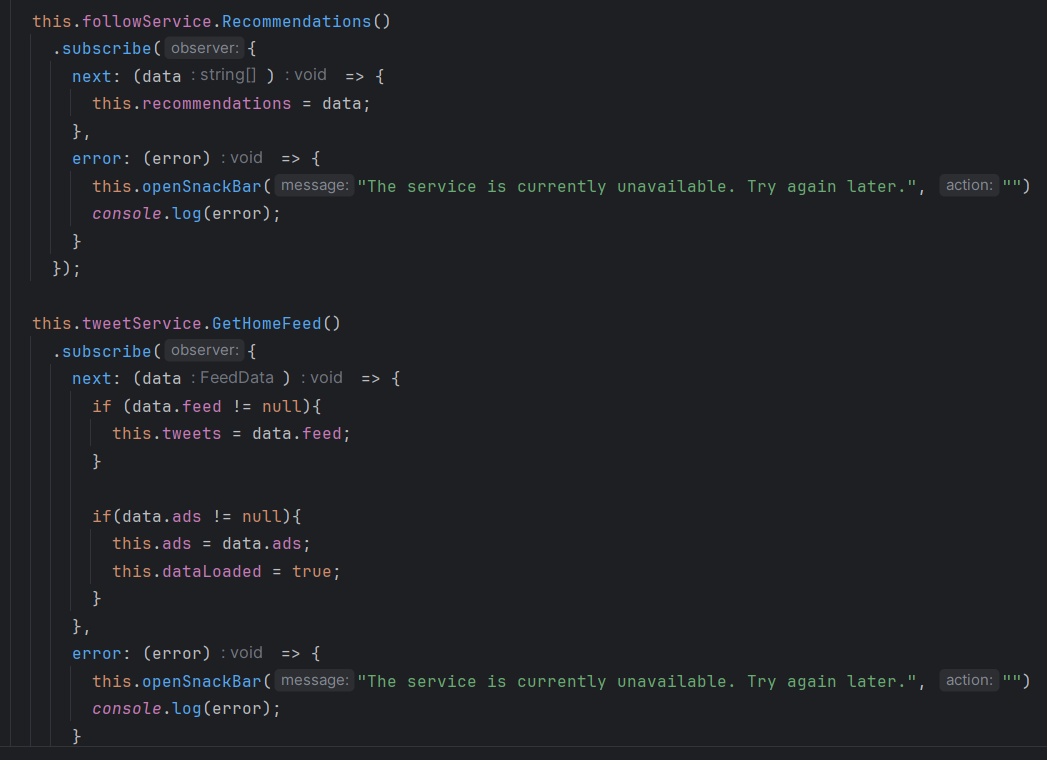


**Листинг 4 – Слање података за регистровање корисника у систему**

Након што се успешно корисник регистровао на систем, на регистрован e-mail му пристиже токен којим ће верификовати регистрацију на систем. На листингу 5 је приказана метода за верификацију корисничког налога која се остварује уношењем пристиглог токена на e-mail.

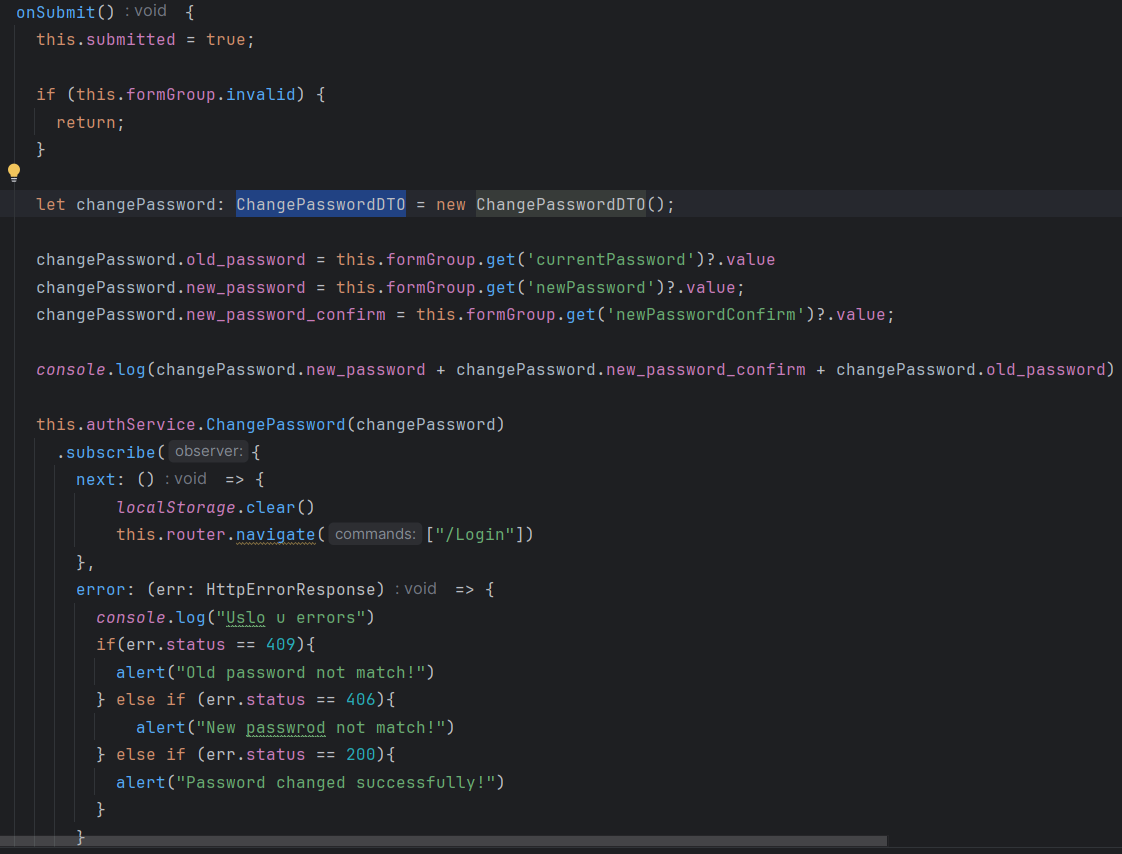
**Листинг 5 – Верификовање корисничког налога**

Корисници који су регистровани на систем, након пријављивања на систем, имају могућност да приступе својој почетној страници на којој му се излситавају објаве свих људи које прати и своје објаве. На листингу 6 је приказана метода OnInit() која преузима све објаве пријављеног корисника као што и преузима све препоруке корисника које пријављени корисник може запратити.



**Листинг 6 – Генерисање почетне странице пријављеног корисника**

Пријављени корисник може променити своју лозинку приступању ChangePasswordComponent, при чему мора унету своју тренутну лозинку, и два пута треба да унесе нову лозинку. На листингу 7 је приказана метода onSubmit(), унутар које се лозинке прикупљају и додељују објекту ChangePasswordDTO.



**Листинг 7 – Промена лозинке**

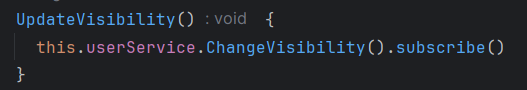
Након успешно добијеног одговора од серверске стране, на основу статусног кода одговора се приказује обавештење кориснику да би имао увид у извршену акцију, при чему статусни код 409 обавештава корисника да се тренутна лозинка не поклапа са унетом, статусни код 406 обавештава корисника да два пута унете лозинке не поклапају, док се за статусни код 200 корисник обавештава да је успешно променио лозинку. За приказивање обавештења је коришћена уграђена метода alert()која приказује обавештење.

Корисник може прегледати персоналне информације приступањем МyProfileComponent . Приликом генерисања компоненте, позива се метода OnInit() која унутар себе позива методу GetMe()чији је задатак да окине позив ка серверској страни и да добави персоналне информације пријављеног корисника на основу корисничког имена унутар JWT токена. Информације које су добијене унутар одговора су његово корисничко име, број пратилаца, број људи које пријављени корисник прати и све његове објаве. На листингу 8 је приказана МyProfileComponent и позивање методе GetMe().



**Листинг 8 – Приказивање персоналних информација пријављеном кориснику**

Кориснику је омогућено да сакрије своје објаве од људи који га не прате, а такође може омогућити да све његове објаве буду видљиве свим корисницима апликације. На листингу 9 је приказана метода ChangeVisibility() за промену видљивости профила пријављеног корисника.



**Листинг 9 – Мењање видљивости профила**

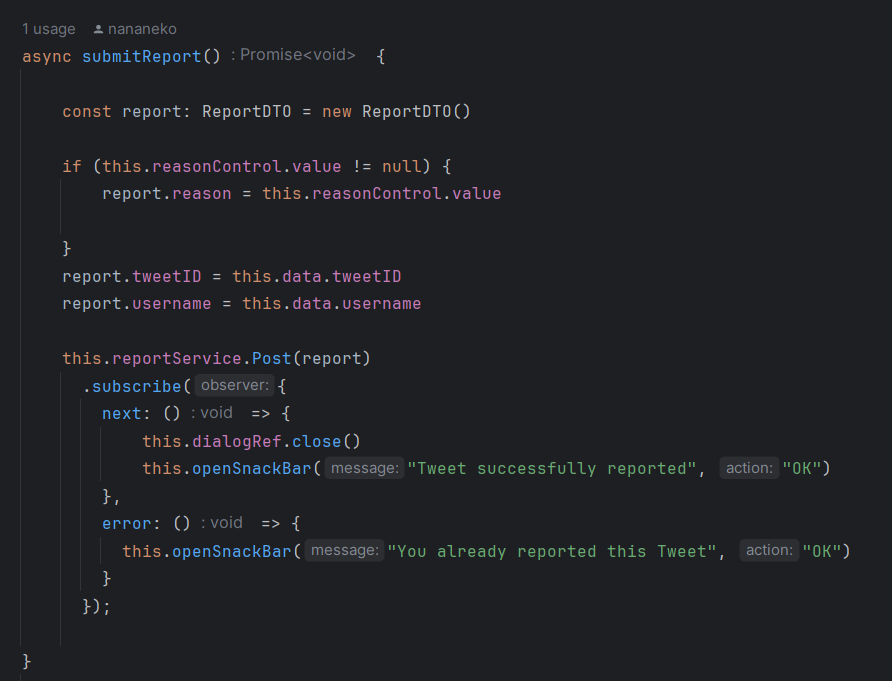
Корисници имају опцију претраге других корисника унутар апликације. На листингу 10 је приказана компонента SearchPeopleComponent помоћу које је могуће извршити претрагу корисника.



**Листинг 10 – Метода Search()за претрагу корисника**

Приликом позивања методе, прикупљају се тип претраге и унета вредности по којој претрага треба да буде извршена. Као одговор на захтев, серверски део апликације ће вратити низ пронађених корисника. Уколико је листа празна, корисник ће бити обавештен да не постоји ниједан корисник са траженом вредношћу.

Објаве појединих корисника могу бити неприкалдне природе, па је корисницима омогућена опција пријављивања неприкладне објаве. На листингу 11 је приказана компонента ReportDialogComponent која у себи садржи методу SubmitReport()помоћу које се остварује подношење пријаве неприкладног садржаја.



**Листинг 11 – Пријављивање неприкладног садржаја**

При пријављивању садржаја, најпре се прво прикупе вредности као што су идентификатор објаве, корисничко име које пријављује садржај и разлог пријављивања садржаја. Разлог пријављивања садржаја представља предефинисане вредности при чему је корисник дужан да одабере један од понуђених разлога.

Пријављивање садржаја мора бити прегледано од стране Админа система, при чему постоји могућност да се пријава на објаву прихвати или одбије. На листингу 12 је приказана компонента ReportListComponent која у себи садржи низ објеката типа ReportItemComponent.



**Листинг 12 – Довавњање свих пријава неприкладног садржаја**

Унутар ове компоненте се помоћи методе Get() добављају све пријаве неприкладног садржаја и затим се врши излиставање свих пријава и свака пријава се прослеђује ReportItemComponent ради генерисања одговарајућег приказа. На листингу 13 је приказано примање прослеђене пријаве компоненти.



**Листинг 13 – Приказивање прослеђене пријаве садржаја**

Метода GetOneTweetById()као параметар прима прослеђени идентификатор објаве која је пријављена. Затим се позива серверски део апликације који као одговор има задатак да врати комплетну објаву са послатим идентификатором.

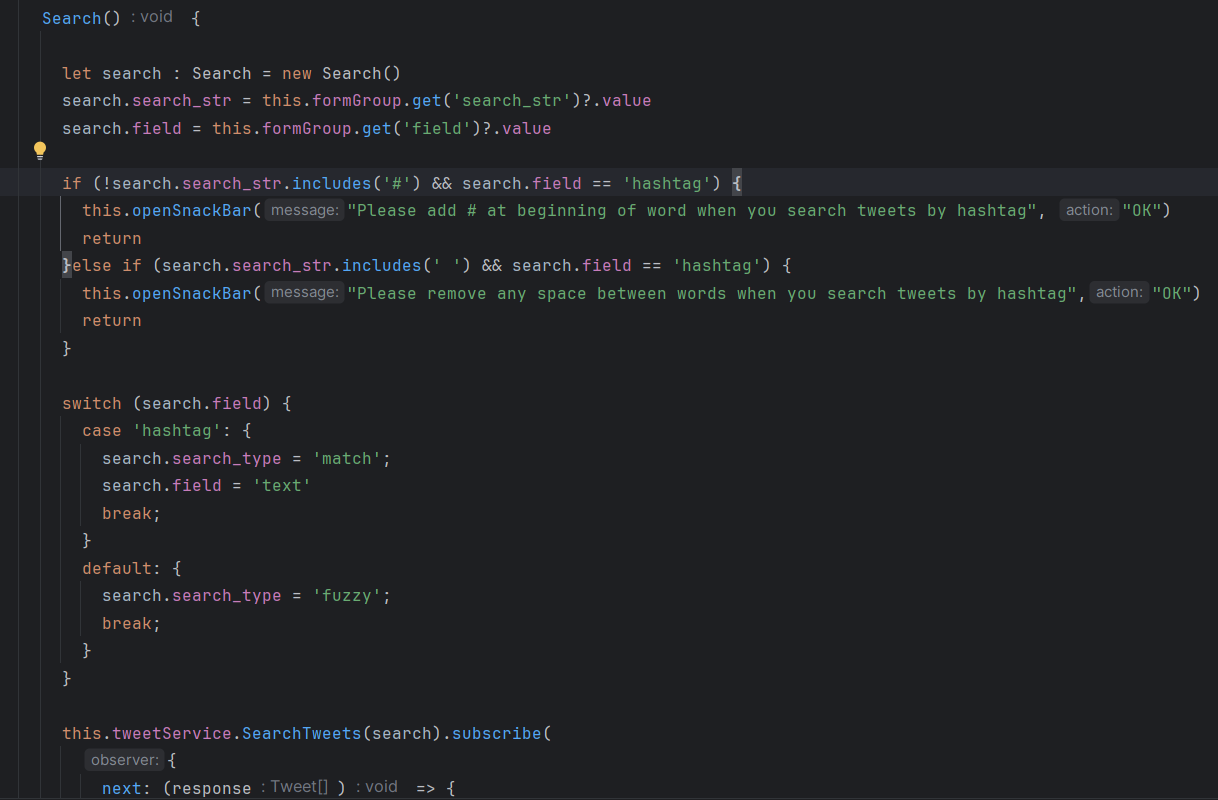
На листингу 14 је приказана метода ChangeStatus() помоћу које Админ прихвата или одбија пријаву.



**Листинг 14 – Прихватање/Одбијање пријаве садржаја**

Након успешног извршавања методе, као одговор Админ ће добити потврду да је успешно променио статус пријаве.

Објаве унутар апликације могу бити претражене од стране пријављеног корисника. Приступањем компоненти SearchTweetsComponent кориснику се отвара могућност да претражи објаве. На листингу 15 је приказана метода Search()помоћу које корисник врши претрагу објава.



**Листинг 15 – Проверавање начина претраге**

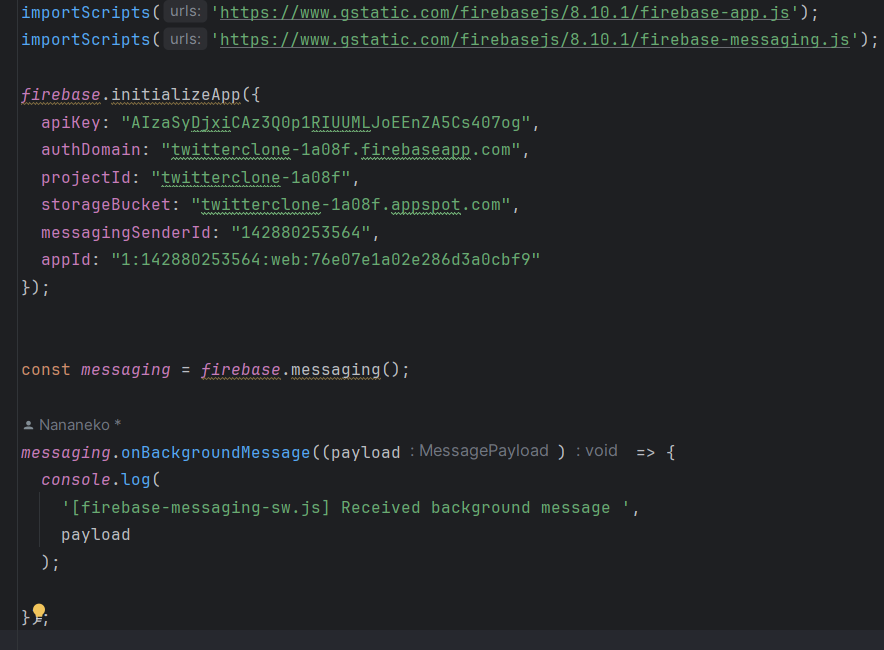
У самом улазу апликације имамо проверу да ли се на правилан начин врши претрага неке објаве. Уколико услови нису испуњени корисник, ће бити обавештен да прилагоди унету вредност према смерницама које су му приказане.На листингу 16 је приказан наставак методе за претрагу објава.



**Листинг 16 – Претрађивање објава унутар апликације Twitter Clone**

Уколико серверски део апликације врати празну листу, корисник ће бити обавештен да нису пронађене објаве са траженим вредностима. У сваком другом сличају ће бити приказане пронађене објаве.

Имплементација Service Workera је приказано на листингу 17.



**Листинг 17 – Имплементација Service Workera**

Идеја овог механизма јесте да ослушкује све захтеве пристигле са Firebase Cloud Messaging услуге. Уколико обавештење пристигне на уређај на коме је покренута апликација, обавештење ће бити приказано у виду пуш нотификације. Функционисање овог механизма је објашњено у серверском делу апликације.

## Серверски део апликације

Серверски део аплкације имплементиран је помоћу програмског језика Го (енг. Go [35]). Серверски део представља микросервисну архитектуру где је сваки сервис задужен за један домен проблема. У овом делу рада је објашњено функционисање серверског дела апликације, на основу пристиглих захтева са клијентског дела апликације. Функционалности су објашњене сукцесивно као и унутар клијентског дела апликације.

Сваки пристигли захтев пролази кроз део система који се назива API Gateway [36], при чему се на основу пристиглог URL [37], захтев се аутоматски преусмери на један од микросервиса. API Gateway je implementiran pomoću NGINX [38] алата. На листингу 18 је приказана NGNIX конфигурација за API Gateway која пристигле захтеве прослеђује аутх сервису.



**Лиситнг 18 – NGNIX конфигурација за аутх сервис**

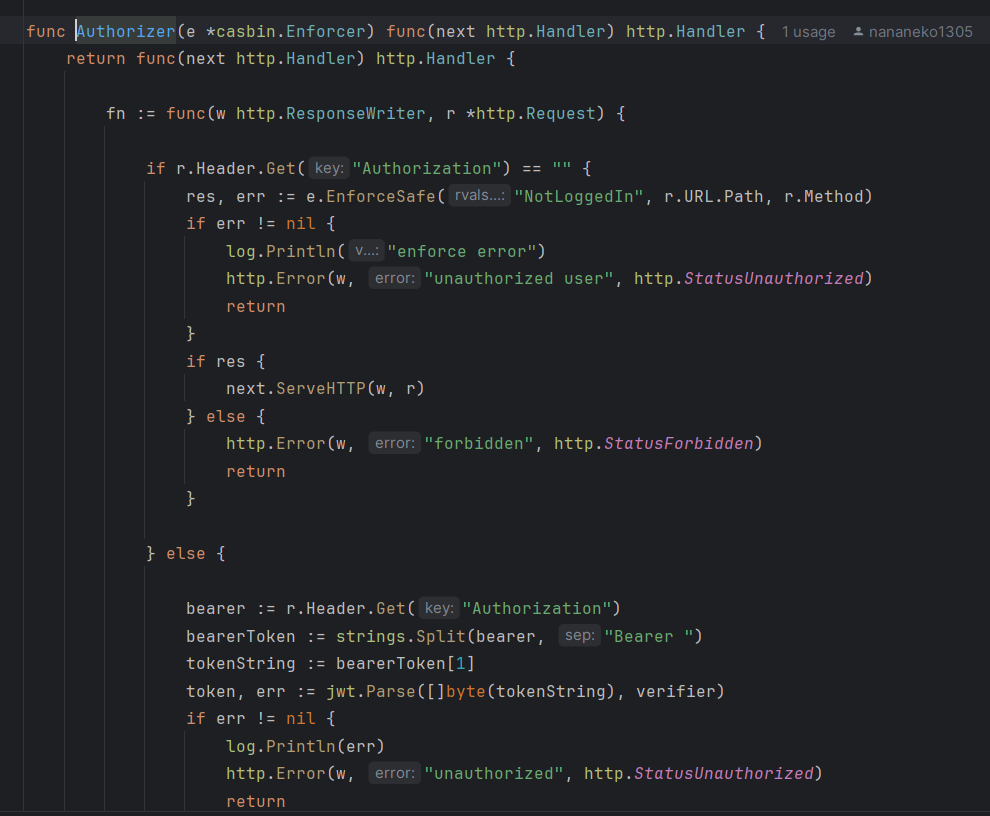
На основу овог листинга можемо закључити да сви пристигли захтеви бивају прослеђени на порт 8003, при чему се прослеђују у JSON [39] формату.

На листингу 19 су приказане све путање до серверског дела унутар аутх сервиса.



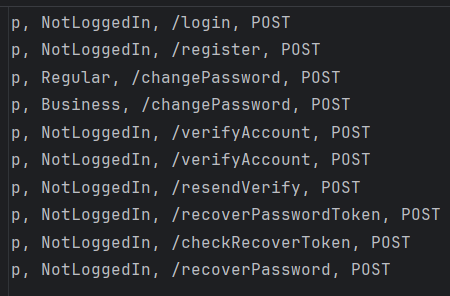
**Листинг 19 – Путање унутар Аутх сервиса**

Сваки сервис у себи садржи проверу улоге. Метода Authorizer()врши проверу улога које су пристигле унутар JWT токена. На листингу 20 је приказан пресретач захтева за проверу пристигле улоге.



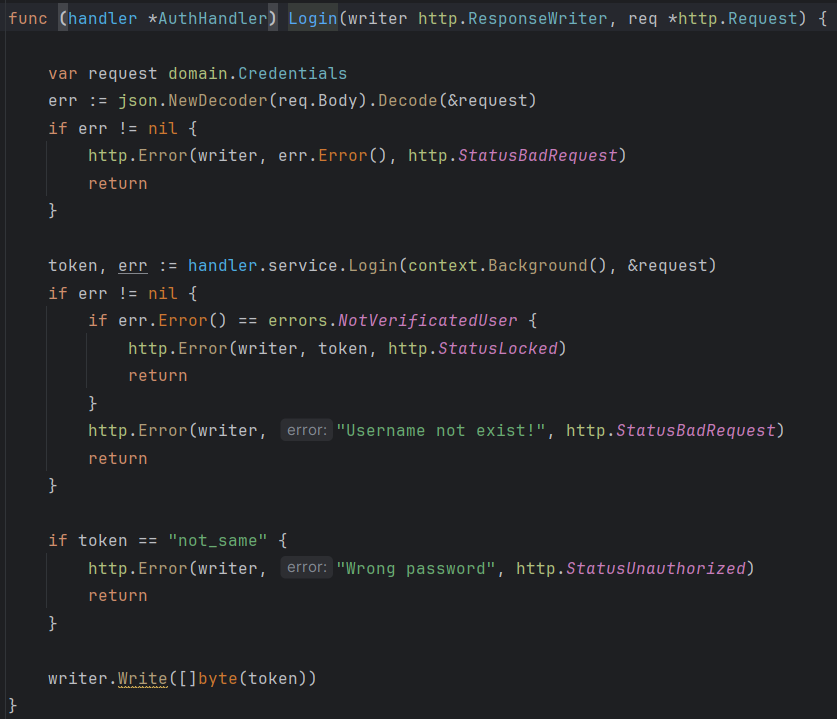
**Листинг 20 – Проверавање улоге унутар пресретача захтева**

Приликом пристизања захтева, прикупља се токен из заглавља захтева при чему се прво проверава да ли је заглавље под кључем „Authorization“ празно. Уколико није празно, врши се провера улоге за URL коме корисник жели да приступи. Свака улога има дефинисан свој URL коме може да се приступи и са којом врстом HTTP методе, као што је приказано на листингу 21.



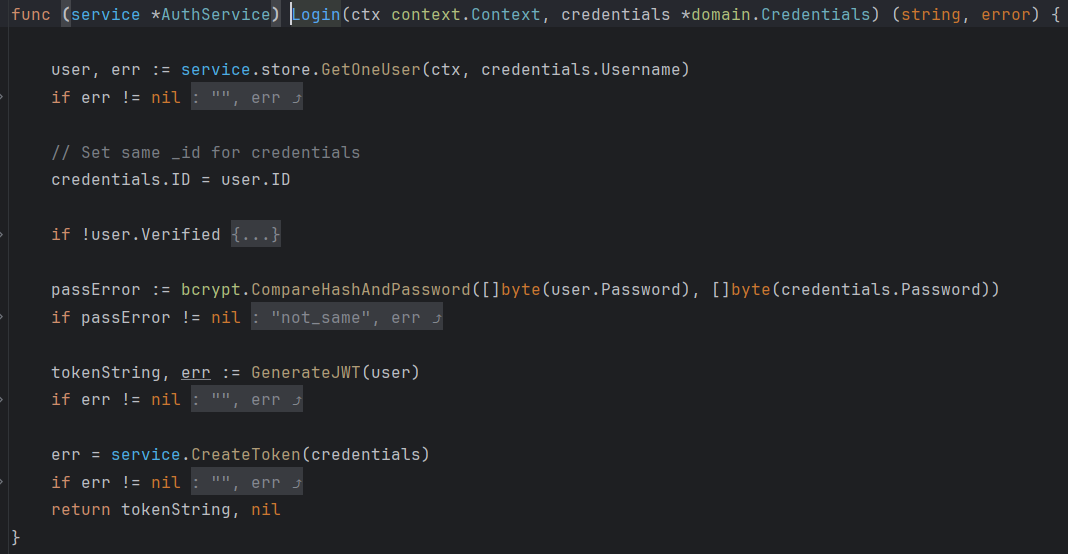
**Листинг 21 – Дефинисање улога за URL**

На листингу 22 је приказана метода за обраду пристиглог захтева за пријављивање корисника.



**Листинг 22 – Руковање пристиглим захтевом за пријављивање корисника**

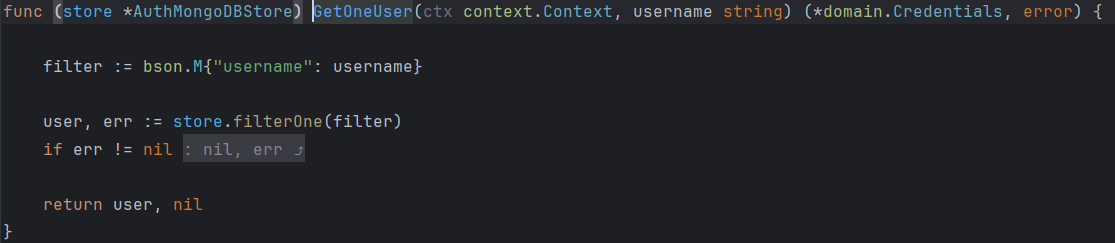
Из пристиглог захтева се преузимају вредности креденцијала које су послате у телу захтева и мапирају се на структуру Credentials. Структура се прослеђује сервисном слоју где се врши позивање базе података као што је приказано на листингу 23.



**Лситинг 23 – Прихватање струкутре унутар сервисног слоја**

На основу пристиглих креденцијала се претражује да ли корисник постоји у систему. Уколико корисник постоји у систему, проверава се да ли је претходно верификовао кориснички налог и уколико јесте, врши се провера лозинке. За проверу лозинке користи се Bcrypt [41] библиотека, која упоређује Hash [42] вредности лозинки.

На листингу 24 је приказана комуникација са базом података за презимање једног корисника.



**Листинг 24 – Претраживање корисника унутар базе**

База података која је коришћена за чување креденцијала је MongoDB [43].

# ДЕМОНСТРАЦИЈА

# ЗАКЉУЧАК

# ЛИТЕРАТУРА

1. World Wide Web, “What is World Wide Web (WWW)?”, доступно на: <https://www.techtarget.com/whatis/definition/World-Wide-Web> ,  
   (посећено 12.09.2023.)
2. Reddit, “Dive Into Anything”, доступно на: <https://www.redditinc.com> ,  
   (посећено 12.09.2023.)
3. Facebook, “About Facebook”,   
   доступно на: <https://about.meta.com/technologies/facebook-app/> ,  
   (посећено 12.09.2023.)
4. LinkedIn, “About LinkedIn”, доступно на: <https://about.linkedin.com> ,  
   (посећено 12.09.2023.)
5. Instagram, “About Instagram”, доступно на: <https://about.instagram.com> ,  
   (посећено 12.09.2023.)
6. Docker, “Docker overview”, доступно на: <https://docs.docker.com/get-started/overview/>   
   (посећено 15.09.2023.)
7. Docker Compose, „Docker Compose overview“, доступно на: <https://docs.docker.com/get-started/overview/>   
   (посећено 15.09.2023.)
8. REST API, „What is REST API?“, доступно на: <https://www.ibm.com/topics/rest-apis>   
   (посећено 15.09.2023.)
9. Node.js, „About Node.js“, доступно на: <https://nodejs.org/en/about>   
   (посећено 15.09.2023.)
10. npm „What is NPM“, доступно на: <https://builtin.com/software-engineering-perspectives/npm>   
    (посећено 15.09.2023.)
11. Go Modules, “Using Go Modules”, доступно на: <https://www.digitalocean.com/community/tutorials/how-to-use-go-modules>  
    (посећено 15.09.2023.)
12. Service Worker, “What is a Service Worker?”, доступно на: <https://medium.com/commencis/what-is-service-worker-4f8dc478f0b9>   
    (посећено 15.09.2023.)
13. Firebase Cloud Messaging, „About FCM Messages“, доступно на: <https://firebase.google.com/docs/cloud-messaging/concept-options>  
    (посећено 15.09.2023)
14. Elastic Search, „Starting with the Elasticsearch Platform and its Solutions“, доступно на: <https://www.elastic.co/guide/index.html>   
    (посећено 15.09.2023.)
15. NATS Message Broker, „What is NATS“, доступно на: <https://docs.nats.io/nats-concepts/what-is-nats>   
    (посећено 15.09.2023.)
16. Container, “Use containers to Build, Share and Run your applications”, доступно на: <https://www.docker.com/resources/what-container/>   
    (посећено 15.09.2023.)
17. YAML, „What is YAML“, доступно на: <https://www.redhat.com/en/topics/automation/what-is-yaml>   
    (посећено 15.09.2023.)
18. HTTP, „What is HTTP“, доступно на: <https://www.cloudflare.com/learning/ddos/glossary/hypertext-transfer-protocol-http/>   
    (посећено 15.09.2023.)
19. CRUD, “What is CRUD?”, доступно на: <https://www.codecademy.com/article/what-is-crud>   
    (посећено 15.09.2023.)
20. package.json, „What is package.json?“, доступно на: <https://docs.npmjs.com/cli/v10/configuring-npm/package-json>   
    (посећено 16.09.2023.)
21. Go, „Build simple, secure, scalable systems with Go“, доступно на: <https://go.dev/>   
    (посећено 16.09.2023.)
22. Google, „About Google“, доступно на: <https://about.google/>   
    (посећено 16.09.2023.)
23. Use-case diagram, „“, доступно на: <https://www.ibm.com/docs/en/rational-soft-arch/9.6.1?topic=diagrams-use-case>   
    (посећено 17.09.2023.)
24. Toggle Switch, „Toggle Switch“, доступно на: <https://mdbootstrap.com/docs/standard/forms/switch/>   
    (посећено 17.09.2023.)
25. Hashtag, „How to use hashtags“, доступно на: <https://help.twitter.com/en/using-twitter/how-to-use-hashtags>   
    (посећено 17.09.2023.)
26. Microservices, „What are microservices?“, доступно на: <https://microservices.io/>   
    (посећено 17.09.2023.)
27. Saga pattern, „Saga Pattern for Microservices Distributed Transactions“, доступно на: <https://medium.com/design-microservices-architecture-with-patterns/saga-pattern-for-microservices-distributed-transactions-7e95d0613345>   
    (посећено 17.09.2023.)
28. Circuit Breaker, „Pattern: Circuit Breaker“, доступно на: <https://microservices.io/patterns/reliability/circuit-breaker.html>   
    (посећено 17.09.2023.)
29. Push Notifications, „Push Notifications“, доступно на: <https://vwo.com/push-notifications/>   
    (посећено 17.09.2023.)
30. Class Diagram, „What is Class Diagram?“, доступно на: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/>   
    (посећено 17.09.2023.)
31. Angular, „Introduction to the Angular docs“, доступно на: <https://angular.io/docs>   
    (посећено 17.09.2023.)
32. HTTP, „HTTP vs. HTTPS: What are the differences?“, доступно на: <https://www.cloudflare.com/learning/ssl/why-is-http-not-secure/>   
    (посећено 17.09.2023.)
33. Database, „What is database?“, доступно на: <https://www.oracle.com/database/what-is-database/>   
    (посећено 17.09.2023.)
34. JWT, „What is JSON Web Tokens?“, доступно на: <https://jwt.io/introduction>   
    (посећено 17.09.2023.)
35. Go, „Build simple, secure, scalable systems with Go“, доступно на: <https://go.dev>   
    (посећено 18.09.2023.)
36. API Gatweay, „What is an API Gateway?“, доступно на: <https://www.nginx.com/learn/api-gateway/>   
    (посећено 18.09.2023.)
37. URL, „What is URL?“, доступно на: <https://developer.mozilla.org/en-US/docs/Learn/Common_questions/Web_mechanics/What_is_a_URL>   
    (посећено 18.09.2023.)
38. NGNIX, доступно на: <https://www.nginx.com/>   
    (посећено 18.09.2023.)
39. JSON, „Introducing JSON“, доступно на: <https://www.json.org/json-en.html>   
    (посећено 18.09.2023.)
40. Endpoint, „What is a Web Service Endpoint?“ доступно на: <https://www.biztalk360.com/blog/web-endpoint-monitoring-biztalk360/>   
    (посећено 18.09.2023.)
41. Bcrypt, доступно на: <https://pkg.go.dev/golang.org/x/crypto/bcrypt>   
    (посећено 18.09.2023.)
42. Hash, доступно на: <https://stytch.com/blog/what-is-password-hashing/>   
    (посећено 18.09.2023.)
43. МongoDB, доступно на: <https://www.mongodb.com/>   
    (посећено 18.09.2023.)

БИОГРАФИЈА