**NearCircle Web Application**

**Submitted By**

**Priyansha Singh**

**Student ID: 885945329**

**A picture containing emblem, text, symbol, trademark

Description automatically generated**

**CPSC-597**

**Spring 2023**

**Advisor: Dr. Rong Jin**

**Department of Computer Science**

**California State University-Fullerton**

A picture containing text, font, graphics, graphic design

Description automatically generated

**Department of Computer Science**

### This project has been satisfactorily demonstrated and is of suitable form.

### This project report is acceptable in partial completion of the requirements for the Master of Science degree in Computer Science.

|  |  |
| --- | --- |
| **NearCircle Web Application** | |
| Project Title |  |
| **Priyansha Singh** |  |
| Student Name |  |
| **Dr. Rong Jin** |  |
| Advisor's Name |  |
| Advisor's signature | Date |
| **Dr. Rong Jin** |  |
| Reviewer's name |  |
| Reviewer's signature | Date |

**Abstract**

Despite living in close proximity to one another, lack of social connection with neighbors can lead to feelings of loneliness and isolation. Social participation in the community is essential to support the quality of social life and a sense of place attachment. The proposed platform aims to provide information about the interests, hobbies, and nationality of people living in the same apartments and enable residents to make connections with them. The platform will serve as a tool for social interaction and help in forming neighborhood communities, which can ultimately contribute to enhancing the quality of life for residents. The project will involve the development of a user-friendly interface, data collection and management, and implementation of algorithms to provide relevant information to users. The objective is to create a practical solution to promote social interaction and help people living in the same apartments get to know each other better.

**Keywords** - Microservices, Spring Boot, Angular, Social connection.

**Contents**

[1.Introduction 6](#_Toc134567783)

[2. Requirements Description 7](#_Toc134567784)

[3. Design Description 8](#_Toc134567785)

[4. Implementation 10](#_Toc134567786)

[5. Test and Integration 20](#_Toc134567787)

[6. Installation Instructions 27](#_Toc134567788)

[7. Operating Instructions 30](#_Toc134567789)

[8. Recommendations for Enhancement 30](#_Toc134567790)

[9. Bibliography 31](#_Toc134567791)

**List of Figures**

[Figure 1: Component based architecture 8](#_Toc134649771)

[Figure 2: App components 9](#_Toc134649772)

[Figure 3: Microservices based architecture 9](#_Toc134649773)

[Figure 4: Application Package Structure 10](#_Toc134649774)

[Figure 5: Application Client side Structure 10](#_Toc134649775)

[Figure 6: Tenants Post Html File-1 11](#_Toc134649776)

[Figure 7: Tenants Post Html File-2 11](#_Toc134649777)

[Figure 8: Tenants Post Html File-3 12](#_Toc134649778)

[Figure 9: Tenants Post Html File-4 12](#_Toc134649779)

[Figure 10: Tenants Post CSS File 13](#_Toc134649780)

[Figure 11: Tenants Post TS File-1 13](#_Toc134649781)

[Figure 12: Tenants Post TS File-2 14](#_Toc134649782)

[Figure 13: Tenants Post TS File-3 15](#_Toc134649783)

[Figure 14: App Configuration File 15](#_Toc134649784)

[Figure 15: ApartmentController Code-1 16](#_Toc134649785)

[Figure 16: ApartmentController Code-2 17](#_Toc134649786)

[Figure 17: ApartmentController Code-3 17](#_Toc134649787)

[Figure 18: ApartmentService Code 18](#_Toc134649788)

[Figure 19: ApartmentServiceImpl Code-1 18](#_Toc134649789)

[Figure 20: ApartmentServiceImpl Code-2 19](#_Toc134649790)

[Figure 21: ApartmentServiceImpl Code-3 19](#_Toc134649791)

[Figure 22: Application.properties Code 20](#_Toc134649792)

[Figure 23: Apartment Manager Sign up Page 20](#_Toc134649793)

[Figure 24: Login Page 21](#_Toc134649794)

[Figure 25: Post Apartment Page 21](#_Toc134649795)

[Figure 26: Apartment Activation Request Status Page 22](#_Toc134649796)

[Figure 27: Admin Dashboard for Apartment Activation Request Approval 22](#_Toc134649797)

[Figure 28: Tenant Registration 23](#_Toc134649798)

[Figure 29: Tenant Profile Creation 23](#_Toc134649799)

[Figure 30: Tenant Profile Creation-1 24](#_Toc134649800)

[Figure 31: Tenant Dashboard 24](#_Toc134649801)

[Figure 32: Manager Dashboard 25](#_Toc134649802)

[Figure 33: Profile Page 25](#_Toc134649803)

[Figure 34: Post Page 26](#_Toc134649804)

[Figure 35: Tenants Search Page 26](#_Toc134649805)

[Figure 36: Friends Page 27](#_Toc134649806)

[Figure 37: Post Feed 27](#_Toc134649807)

[Figure 38: Starting App Client 28](#_Toc134649808)

[Figure 39: Starting Eureka Server 29](#_Toc134649809)

[Figure 40: Starting Application Server 29](#_Toc134649810)

[Figure 41: Starting Application Server-1 30](#_Toc134649811)

[Figure 42: Application Main Page 30](#_Toc134649812)

## **1.Introduction**

1. **Description of the Problem**

Apartment living is a way of life for about 39 million Americans, or nearly 1 in 8 people. Lack of social connection with neighbors is the main factor contributing to social isolation among residents of high-rise flats in inner cities. Even though it is generally believed that neighborhood ties are weaker than those between friends and family, social participation in the community is recognized as a significant indicator to support the quality of social life and a sense of place attachment. The fundamental process of forming neighborhood communities is encouraged by social interaction since it teaches people about their neighbors and the local social structure. The problem is to develop a platform that can help people living in the same apartments get to know the people living around them by providing information such as what are their interests and hobbies, what nationality they belong to and help in making connections with them.

1. **Project Objectives**

The objective of this master project is to develop a web-application called “**NearCircle**” specifically created for the people staying in the rented communities or apartments to strengthen their social connection within their own neighborhood. As tenants, its challenging to meet new people or find people from your own community and make friends when you have just moved into a new apartment specially in a new country and making a move virtually is much easier than knocking the doors physically.

NearCircle is an application that apartment residents can use to find out who lives nearby, what community or age group they belong to, what their hobbies are, add them as friends and share posts with them. This application will allow users to register as apartment property managers and tenants. In essence, it’s a tool to explore your immediate circle.

The learning objective behind building this application is to learn how to make use of Spring Boot Microservices for creating back-end functionality and use Angular for the front-end part. Microservices is an on-going buzzword in the industry where entire functionality of the product is broken down into smaller independent units and as each unit is independent it gives the developer enough flexibility to program units in different technologies with newer stack without having to change anything with the existing code

The skills learnt as part of developing this project would be beneficial for my future job as well as my area of interest lies in web backend engineering.

1. **Development Environment**

**Software Tools**

Coding Platforms-Visual Studio Code -version 1.77.3, IntelliJ IDEA Community-version 2022.3.2

Libraries-Node.js -version18.13.0, Npm-version 9.4.2, JDK 11

Command Line Tool-Angular CLI-version 15.1.6

Database-MySQL-version 8.0

Frameworks-JAVA Spring Boot Microservices, Angular

Browser-Google Chrome, Mozilla Firefox

**Hardware**

Operating System-Windows 10 (64 bit)

Processor-Intel Core i7 or Higher

RAM-8GB

Hard Disk-20 GB or above

1. **Operational Environment**

**Software Libraries**

Node.js -version18.13.0

Npm-version 9.4.2

JDK 11

Database-MySQL-version 8.0

Browser-Google Chrome, Mozilla Firefox

**Hardware**

Operating System-Windows 10 (64 bit)

Processor-Intel Core i7 or Higher

RAM-16GB

Hard Disk-320 GB or above

## **2. Requirements Description**

The application will have the following external functions and the interfaces:

* **Sign Up-**Users such as Apartment Managers and Tenants should be able to register to the NearCircle App by providing information such as email id, password, name and role. The role field in the screen should be a dropdown with two options-MANAGER and TENANT. An ADMIN role should be created in the back end but would not be available in the front-end for the user selection.
* **Login-** Users such as Apartment Managers and Tenants should be able to login into the app by providing details such email and password. Login Service in the back end should be able to verify the user credentials provided at the time of login and once authenticated login should be successful otherwise an error message should be displayed “bad credentials”.
* **Apartment Profile Activation**- Apartment Managers should be able to submit the apartment activation request to admin through the Post Apartment screen. In the screen, they should be able to enter details such as Manager name, apartment name, apartment address, city, state, country, work phone number and work id proof. Once the request is submitted, they should be able to view their request in the My Apartments screen in a “PENDING” status. Once the admin verifies and approves the request, their apartment should become active on the app and be available for selection in tenants profile screen.
* **Tenants Profile Creation-**Tenants should be able to create their profile on the application by providing details such as first name, last name, apartment name, apartment unit no, age category, mobile no, bio, nationality, and interests. Age category should be a dropdown with three options-Young (15-30),Middle Age(30-50),Senior(50+) and apartment name should display the list of active apartments on the NearCircle app. Apartment unit no should be an optional field in the screen. Tenants should also be able to upload their picture in the profile creation section. Once created a profile verification request should be sent to the apartment managers to verify the legitimate tenants and the request status should be available in the tenant’s dashboard.
* **Tenants Profile Verification** -Apartment Managers should be able to view the profile verification request in Manager’s dashboard screen with all the profile details and with an option to approve or reject the request. Only after the profile is verified, the tenants should be able to fully access the app features such as search tenants, friends and post section.
* **Edit Profile**-Tenants should be able to edit their profile details by clicking on the edit button in the post section page.
* **Share Posts**-Tenants should be able to share posts on their feed and they should be able to view their friends post in their post feed too.
* **Add Friends**-Tenants should be able to add other tenants into their friend list by clicking on the friend button and their friend count should be updated. Once added as nearcircle friend, users should be able to view their profile and profile picture.
* **Search Tenants**- Users should be able to view the list of the people staying within their apartments using the search option provided in the dashboard section based on various search filters like nationality, age category and interests. Users should be able to view the profiles of filtered results with the option to add as a friend or send message to connect.
* **Logout**-Users should be able to logout from the application by clicking on the logout button.

## **3. Design Description**

**Component-based application architecture**

The Front end of the application has been designed using component-based application architecture that focuses on creating software systems out of reusable, modular components. The goal is to divide large systems into smaller, easier-to-manage parts, where each part stands for a unique functional unit.

Component-based architecture has the advantage of encouraging code reuse and modularization, which can speed up development, lower the chance of errors, and make it simpler to grow and maintain the program over time. Since components can be added and removed as needed without affecting the system as a whole, it can also aid in increasing the application's flexibility.

A black screen with white text

Description automatically generated with low confidence

Figure 1: Component based architecture

Each component consists of four files:

* HTML file- defines the layout, structure and content of how the component will be viewed.
* CSS file-defines the style of the component.
* Typescript file-acts as the component class which defines the component’s behaviour and interaction with the view
* Spec TS files-defines the units tests for component.

For the NearCircle app, following components have been created.



Figure 2: App components

**Microservices architecture**

The backend of the application has been designed using Spring Boot Microservices. With the help of the Java programming language, web applications and microservices can be created using the Spring Boot framework. Large-scale applications can be created using the architectural style known as microservices, which consists of several smaller, independent services that interact to deliver the functionality of the bigger program. We have used Eureka Server which is a service registry provided by the Spring Boot to allow microservices to discover and interact with each other. Developers can take advantage of the vast ecosystem of tools and libraries offered by the Spring Framework by using Spring Boot to create microservices, as well as the numerous frameworks and tools created especially for Spring Boot.

A diagram of a service

Description automatically generated with low confidence

Figure 3: Microservices based architecture

Following services have been created for the application:

* **UserService**-This service is used to create the admin account at the server startup and is used to register and authenticate application users such as Apartment Managers and tenants and update their details in the users’ table.
* **AdminService**-This service is used to approve and reject the apartment requests coming from the apartment managers and get the details of all apartments in the app from the apartment table.
* **ApartmentService**-This service is used to post apartment request from the managers and approve or reject tenants profile creation request. This service mainly deals with apartment and tenant table in the database.
* **TenantService-**This service is used to post tenants profile creation request, upload tenants profile picture, get all the tenants posts, search other tenants and add tenants as friend in the friend list. This service deals with tenant, posts, friends and profile\_img table in the database.

**4. Implementation**

NearCircle application package comes with two folders as shown below:

A screenshot of a computer

Description automatically generated with medium confidence

Figure 4: Application Package Structure

**Client-side code**

Near\_Circle\_Web contains the client-side code of application written using Angular framework and TypeScript language. Near\_Circle\_Web\src\app contains the source code files for the implementation of different modules such as admin, login, manager, tenant etc. Each module has 3 main files an html file containing the html code for the page, CSS file containing the visual style code and a typescript file containing the code handling the interactions between the presentation layer and the backend layer.

A screenshot of a computer

Description automatically generated with medium confidence

Figure 5: Application Client side Structure

Code snippet for the tenants post page html file

A screen shot of a computer program

Description automatically generated with low confidence

Figure 6: Tenants Post Html File-1

A picture containing text, screenshot, menu

Description automatically generated

Figure 7: Tenants Post Html File-2

A screen shot of a computer program

Description automatically generated with low confidence

Figure 8: Tenants Post Html File-3

A screen shot of a computer screen

Description automatically generated with low confidence

Figure 9: Tenants Post Html File-4

Code snippet for the tenants post page css file

A picture containing text, screenshot, software, multimedia software

Description automatically generated

Figure 10: Tenants Post CSS File

Code snippet for the tenants post page typescript file

A screen shot of a computer program

Description automatically generated with low confidence

Figure 11: Tenants Post TS File-1

A screen shot of a computer program

Description automatically generated with low confidence

A screen shot of a computer program

Description automatically generated with low confidence

Figure 12: Tenants Post TS File-2

A screen shot of a computer program

Description automatically generated with low confidence

Figure 13: Tenants Post TS File-3

**Configuration File**-environment.ts file contains the details of the microservices url.

A screen shot of a computer program

Description automatically generated with low confidence

Figure 14: App Configuration File

**Server-side code**

Near\_circle\_server contains the code of the back-end services created using Spring Boot framework and java programming language. All the main services for admin, user, manager and tenant are present in the src/main/java/com/nearcircle/services folder.

A screenshot of a computer

Description automatically generated

Figure 15: App server side code structure

Code snippet for the ApartmentController.java

A screen shot of a computer

Description automatically generated with medium confidence

Figure 15: ApartmentController Code-1

A screen shot of a computer screen

Description automatically generated with low confidence

Figure 16: ApartmentController Code-2

A screen shot of a computer program

Description automatically generated with low confidence

Figure 17: ApartmentController Code-3

Code Snippet for ApartmentService.java

A screen shot of a computer program

Description automatically generated with low confidence

Figure 18: ApartmentService Code

Code Snippet for ApartmentServiceImpl.java

A screen shot of a computer

Description automatically generated with medium confidence

Figure 19: ApartmentServiceImpl Code-1

A screen shot of a computer program

Description automatically generated with medium confidence

Figure 20: ApartmentServiceImpl Code-2

A screen shot of a computer program

Description automatically generated with low confidence

Figure 21: ApartmentServiceImpl Code-3

Application.properties contains the application url and database connectivity properties.

A screenshot of a computer

Description automatically generated with medium confidence

Figure 22: Application.properties Code

## **5. Test and Integration**

Unit Testing for each module and Integration testing was performed for the NearCircle application to ensure that all the modules work correctly as per the requirement. Below are the test screenshots taken as part of testing phase.

**Test Screenshots**

A screenshot of a computer

Description automatically generated with medium confidence

Figure 23: Apartment Manager Sign up Page

A screenshot of a computer

Description automatically generated

Figure 24: Login Page

A screenshot of a computer

Description automatically generated

Figure 25: Post Apartment Page

A screenshot of a computer

Description automatically generated with medium confidence

Figure 26: Apartment Activation Request Status Page

A screenshot of a computer

Description automatically generated

Figure 27: Admin Dashboard for Apartment Activation Request Approval

A screenshot of a computer

Description automatically generated with medium confidence

Figure 28: Tenant Registration

A screenshot of a computer

Description automatically generated

Figure 29: Tenant Profile Creation

A screenshot of a computer

Description automatically generated

Figure 30: Tenant Profile Creation-1

A screenshot of a computer

Description automatically generated

Figure 31: Tenant Dashboard

A screenshot of a computer

Description automatically generated

Figure 32: Manager Dashboard

A screenshot of a computer

Description automatically generated

Figure 33: Profile Page

A screenshot of a computer

Description automatically generated

Figure 34: Post Page

A screenshot of a computer

Description automatically generated

Figure 35: Tenants Search Page

A screenshot of a computer

Description automatically generated

Figure 36: Friends Page

A screenshot of a computer

Description automatically generated

Figure 37: Post Feed

## **6. Installation Instructions**

To install the software system, follow these steps:

1. Install Node.js version18.13.0.
2. Install Npm version 9.4.2.
3. Install the Angular CLI version 15.1.6.
4. Install JDK 11 and ensure the bin path is set as JAVA\_HOME in the system environment variables.
5. Install MySQL 8.0 and create a clean database with name nearcircle\_db and credentials root/root.
6. Download the source code files for the NearCircle Application from below location: https://github.com/P-R-I-16/NearCircleApp.git
7. Open a command prompt and navigate to the directory where the client side code files are located and **run npm install**
8. Go inside the Near\_Circle\_Web root folder and **run ng serve** command to start the application client.

A screen shot of a computer

Description automatically generated with medium confidence

Figure 38: Starting App Client

1. To start the backend services go inside the server side code Near\_Circle\_Server\Near Circle\_Eureka\_Service folder in the command prompt and **run java -jar server-0.0.1-SNAPSHOT.jar** to start the Eureka server.

A screen shot of a computer screen

Description automatically generated with low confidence

A picture containing text, screenshot, font

Description automatically generated

Figure 39: Starting Eureka Server

1. Then go inside the Near\_Circle\_Server\Near Circle\_service in the command prompt and **run** **java -jar Near-Circle\_Service-0.0.1-SNAPSHOT.jar** command to start the NearCircleApplicationService.

A screen shot of a computer screen

Description automatically generated with medium confidence

Figure 40: Starting Application Server

A screenshot of a computer screen

Description automatically generated with medium confidence

Figure 41: Starting Application Server-1

## **7. Operating Instructions**

After the installation is done successfully, application client, eureka server and spring boot microservices are up then open the below application URL in the browser and the app’s login page should be displayed.

**App URL-** [**http://localhost:4200/login**](http://localhost:4200/login)

Use **Admin credentials** to verify login is working fine.

Username-admin@gmail.com

Password-admin

A screenshot of a computer

Description automatically generated

Figure 42: Application Main Page

## **8. Recommendations for Enhancement**

List of features which can be added in the future for this application are:

1. Manager module can be enhanced where they can view the list of tenants staying within their apartments in the dashboard.
2. Tenants can interact with the apartment managers using chat option and can share their issues or complaints directly with them.
3. Rent module can be added where tenants can pay their rents online on the near circle app.
4. Offensive posts shared by tenants can be reported to managers via report post button.
5. Tenants can create events happening in their apartments and send invites to other tenants on the app.

## **9. Bibliography**

1. Chile L.M., Black X.M., Neill C. Experience, and expression of social isolation by inner-city high-rise residents. *Hous. Care Support.*2014;**17**:151–166. doi: 10.1108/HCS-11-2013-0021.
2. Weijs-Perrée M., Berg P.V.D., Arentze T., Kemperman A. Social networks, social satisfaction, and place attachment in the neighborhood. *Region.*2017;**4**:133. doi: 10.18335/region.v4i3.194.
3. Granovetter M.S. The Strength of Weak Ties. *Am. J. Sociol.*1973;**78**:1360–1380. doi: 10.1086/225469.
4. Grannis R. *From the Ground up: Translating Geography into Community through Neighbor Networks.* Princeton University Press; Princeton, NJ, USA: 2009. pp. 1–242.
5. A. Stec, “Database design in a microservices architecture,” *Baeldung on Computer Science*, 09-Nov-2022. [Online]. Available: https://www.baeldung.com/cs/microservices-db-design. [Accessed: 06-Dec-2022].
6. S. R. Goniwada, “Microservices architecture and Design,” *Cloud Native Architecture and Design*, pp. 191–240, 2021.
7. O. Al-Debagy and P. Martinek, “A comparative review of microservices and Monolithic Architectures,” *2018 IEEE 18th International Symposium on Computational Intelligence and Informatics (CINTI)*, 2018.
8. Team, N.-Z. (n.d.). *Components: NG-zorro*. NG. https://ng.ant.design/components/overview/en
9. *Spring Blog*. Microservices with Spring. (n.d.). https://spring.io/blog/2015/07/14/microservices-with-spring
10. Sharma, A., Abdullah, & Java Development Journal. (2020, September 8). *Microservices with Spring Boot - building microservices application using Spring Boot*. Java Development Journal. https://www.javadevjournal.com/spring-boot/microservices-with-spring-boot/