Service Novigrad

Dr. Hussein Al-Osman

SEG 2105

Team Members

Amro Ahmed	7937646
Daniel Krohn-Anthony	8631849
Raveena Grewal	300065683
Irvine Minh Duc Tran	300126016
Cem Gurel	300104480
Katada Freije	300121815

Submission Date: 6 December 2020

Table of Contents

List of Figures

List of Tables

1. Introduction

In this project we were assigned to create an app for “NOVIGRAD” that would function something like Service Ontario. In this app there are 3 roles that the user can have which is an administrator, employee, or customer. In the app, the admin can create, edit, and delete services. The admin account can also delete accounts of branches and customers. An employee account can see the branch service hours and services. That employee can choose to add the services requested by the admin. In the app, it will show the services that the branch provides. The branch can also see the service requests from customers. As a customer account, the user can search branches by address, type of service provided, and working hours and rate that branch. The user will also be able to send requests with the required information to the specific branch.

2. Class Diagram

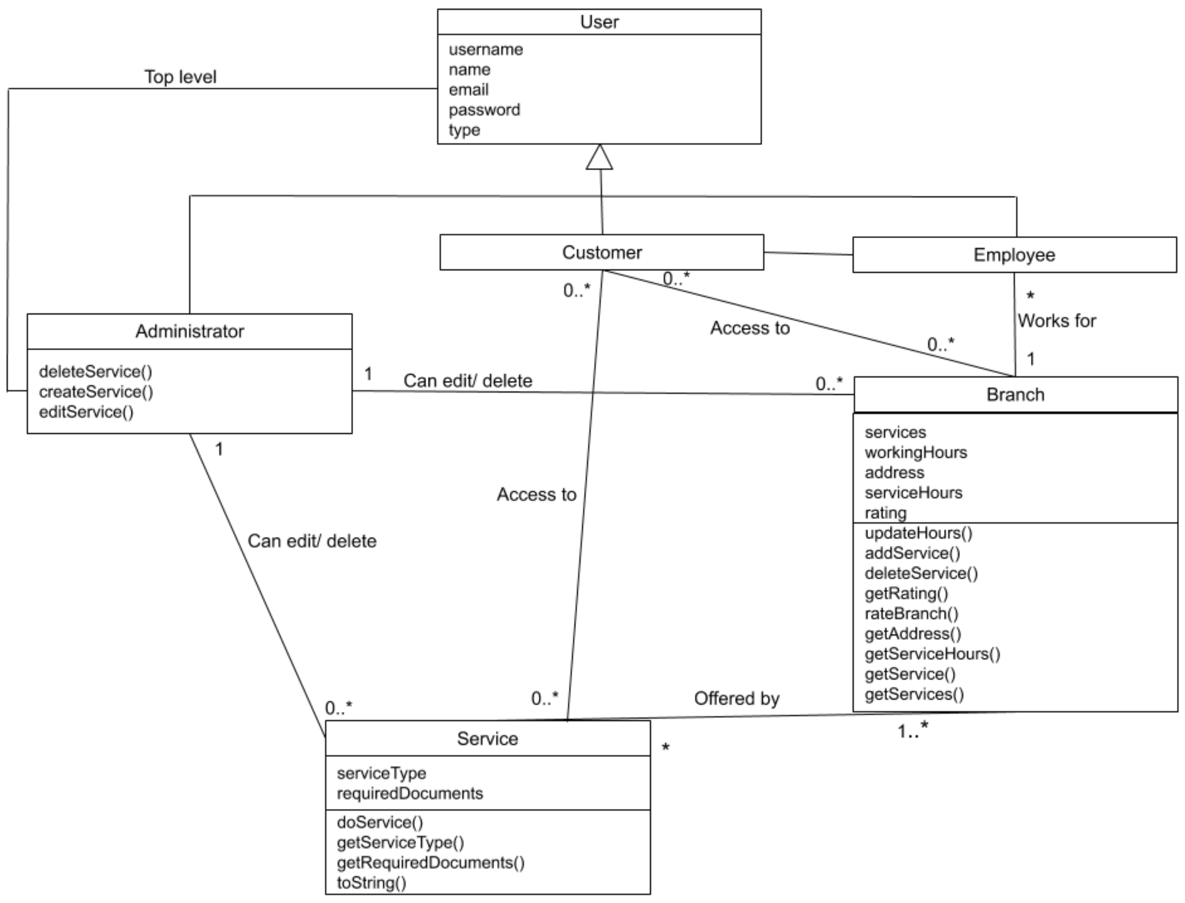


Figure 2.1 Service NOVIGRAD UML Diagram

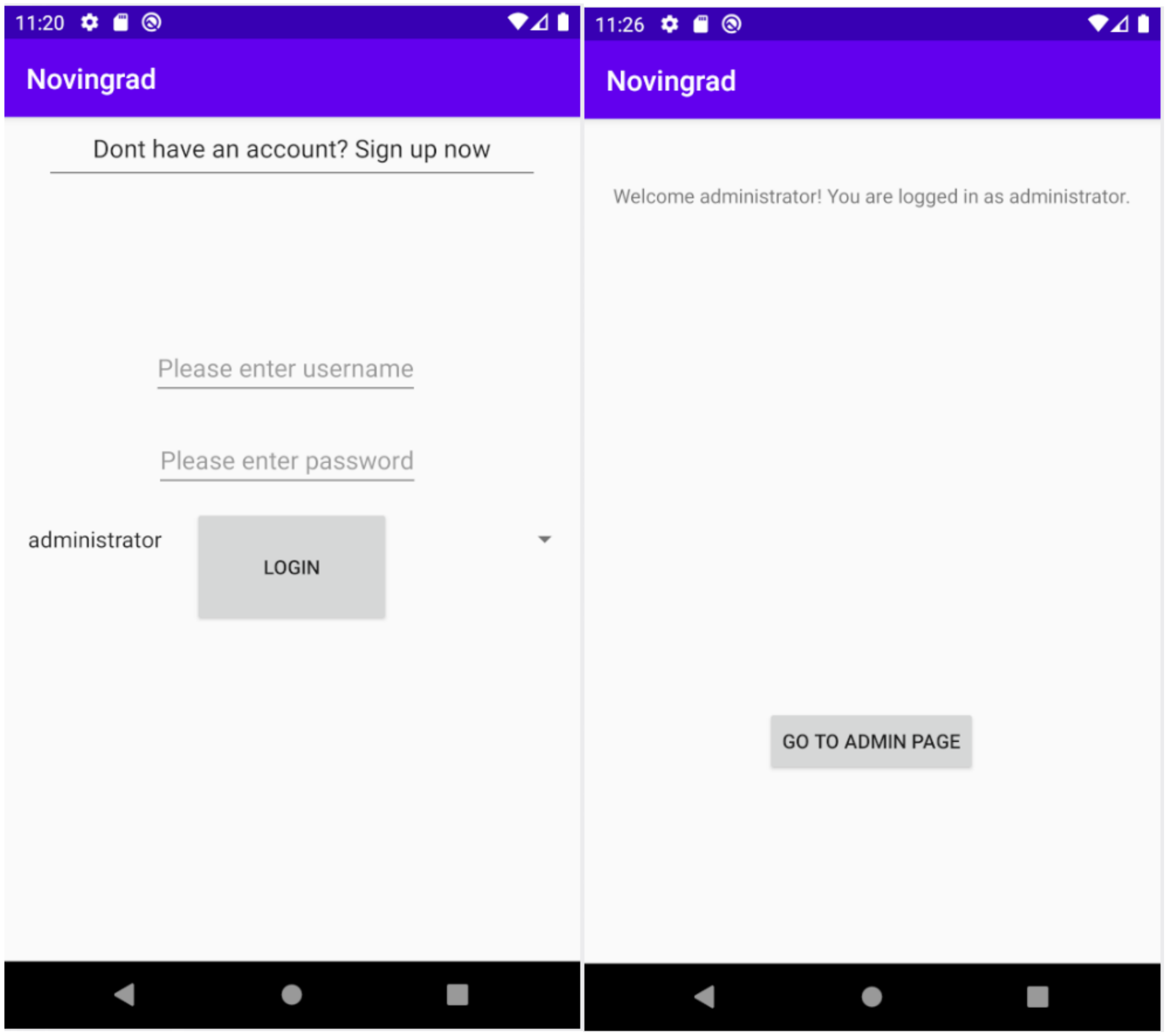
3. Roles

***Note:* Daniel and Amro were part of another group and joined us later in the course**

Table 3.1 Team Roles

Name	Deliverable 1(Oct 19)	Deliverable 2 (Nov 1)	Deliverable 3 (Nov 24)	Deliverable 4
Amro	N/A	Validation for admin functions	Service branch validation	Service request validation.
Cem	UML and Welcome page		UML	Customer UI
Daniel	N/A	Added admin functionality to delete users from the system.	Added service hours upon creation of new services, and validation for it.	Allow customers to make service requests.
Katada	Took care of backend (database setup, login), welcome page functionality	Add, edit, and delete services from database for admin page	Branch account creation, adding and deleting services from branch	Search for branch functionality
Irvine	Sign up and validation functionality	Unit tests	Branch services offered	Branch rating functionality
Raveena	Welcome page UI	UML	Unit tests	Unit tests and UML

4. Screenshots



Welcome administrator! You are logged in as administrator.

Figure 4.1 Main Welcome UI app page

4.1 Admin Functions

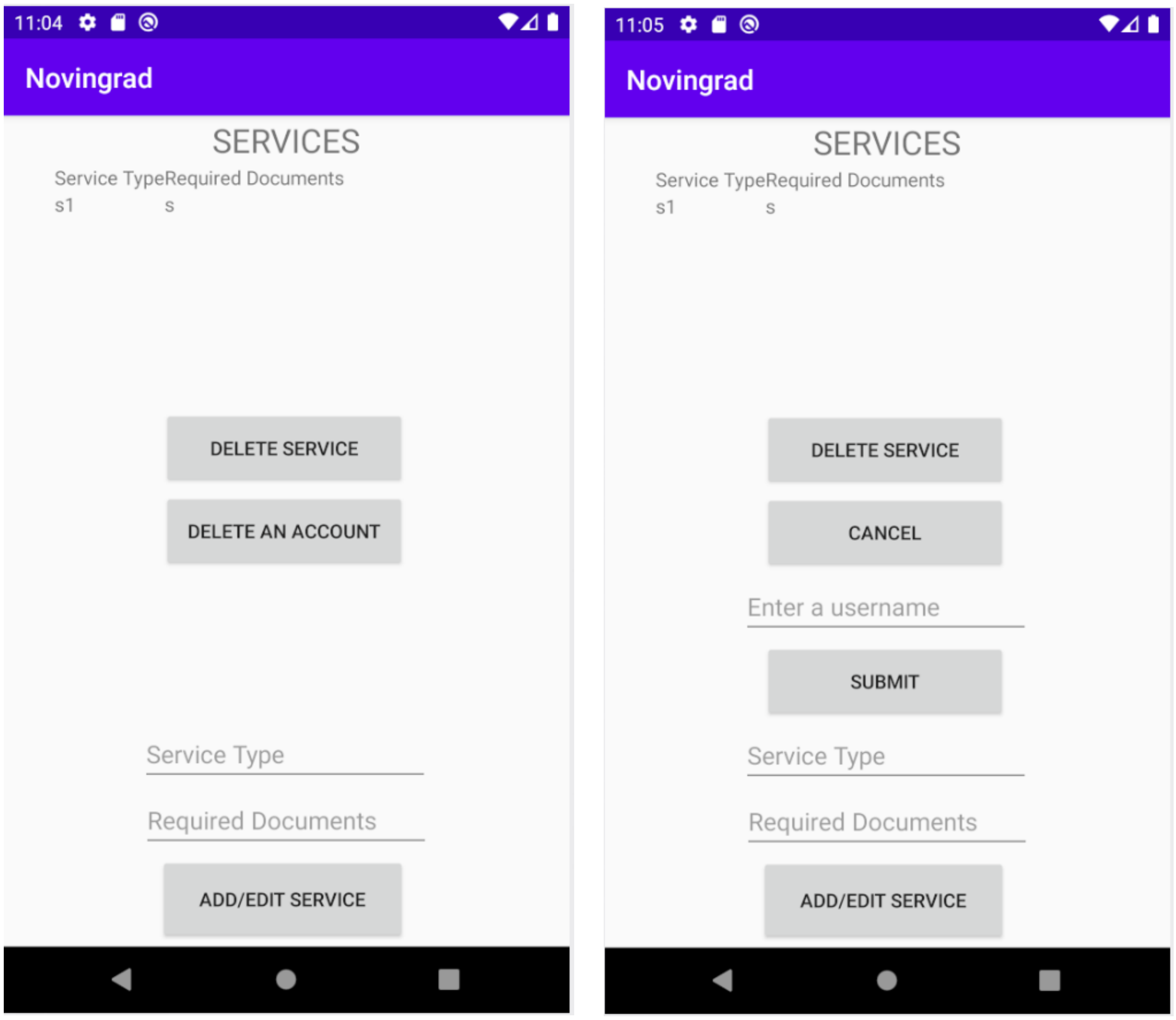
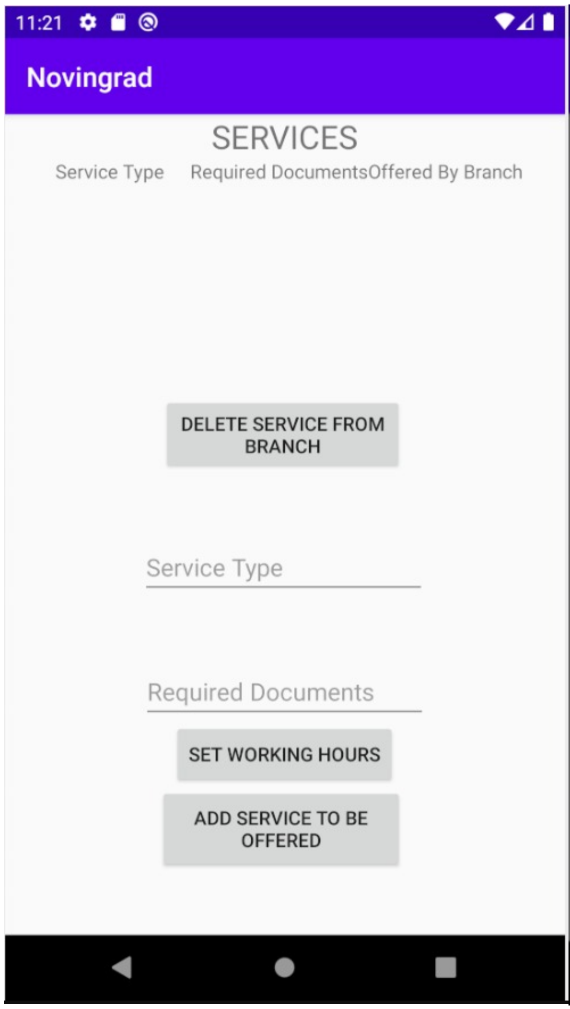


Figure 4.2 Admin Service and Account Functionality

4.2 Employee Functions


11:21
Novingrad
SERVICES
Service Type
Required DocumentsOffered By Branch
DELETE SERVICE FROM BRANCH
Service Type
Required Documents
SET WORKING HOURS
ADD SERVICE TO BE OFFERED

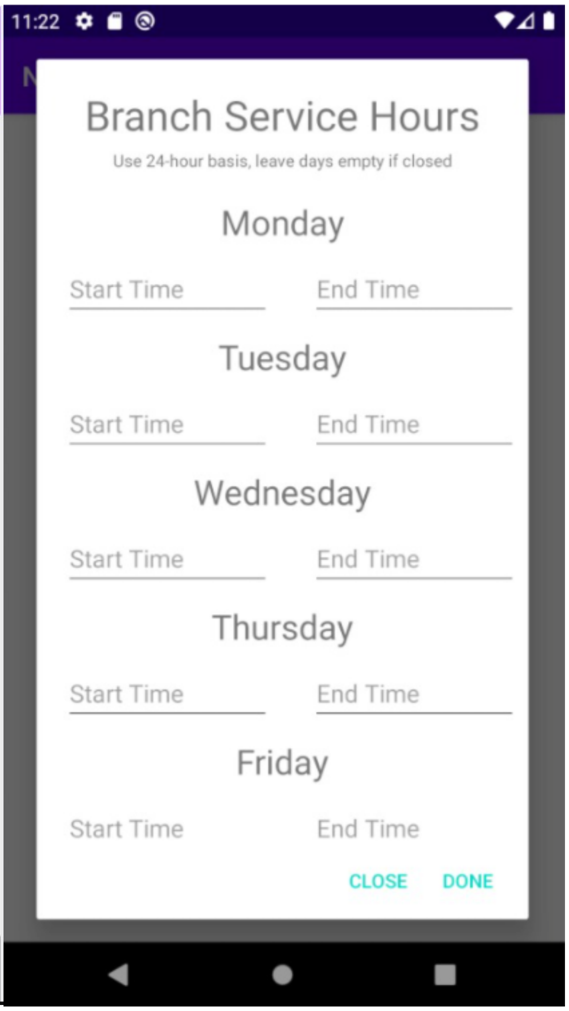
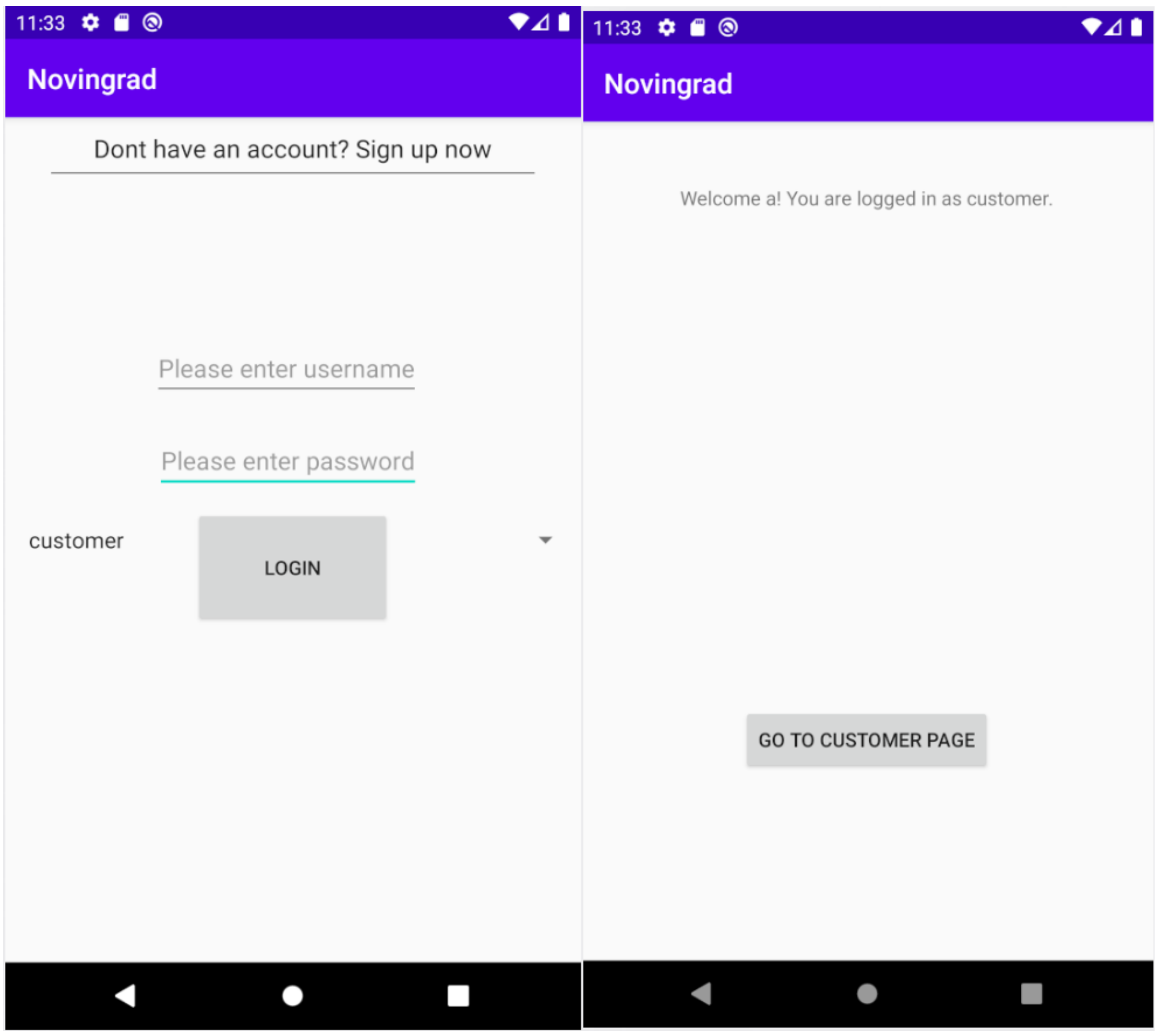


Figure 4.3 Addition of new services and edit of service hours

4.3 Customer Functions



Welcome a! You are logged in as customer.

Figure 4.4 Customer Welcome UI app page

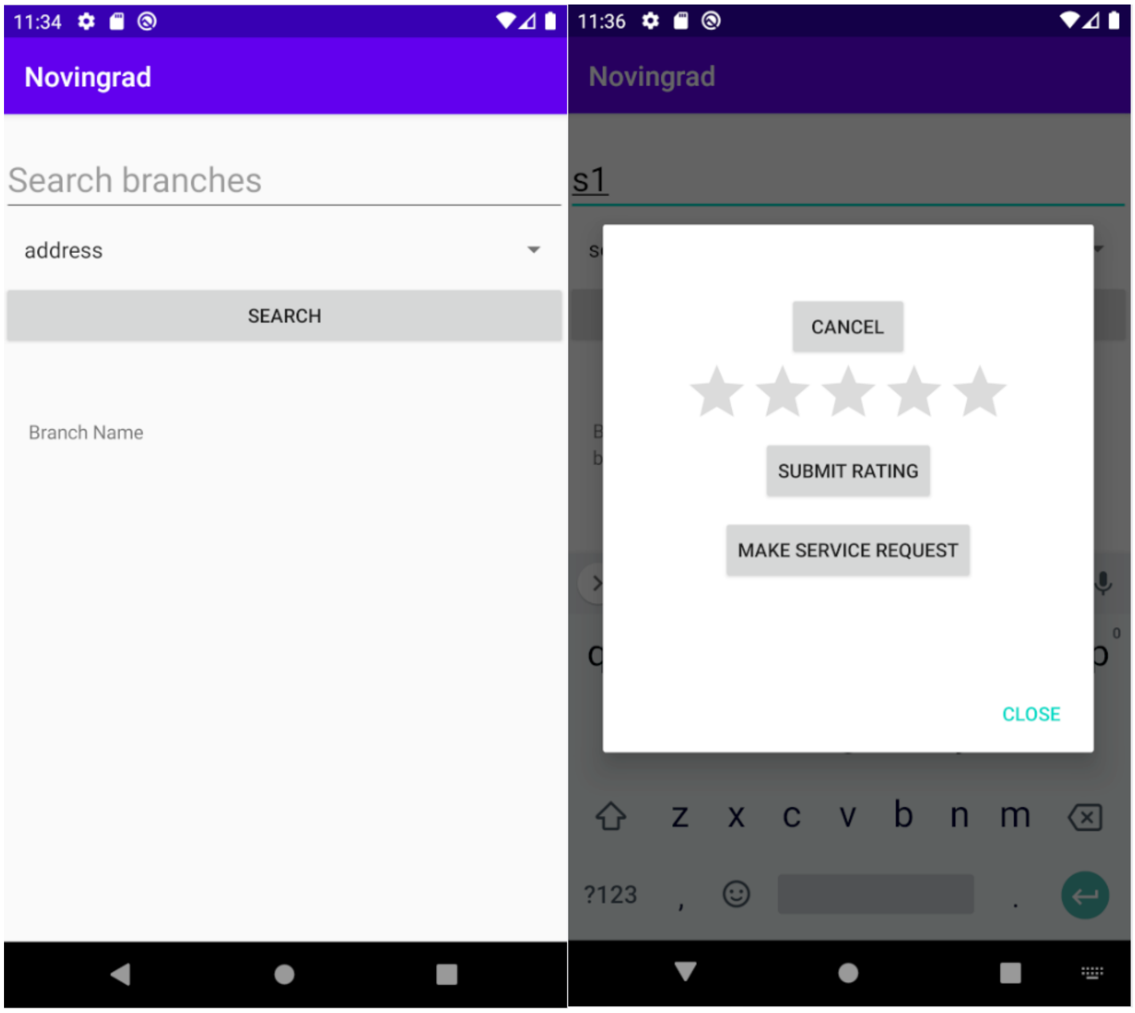


Figure 4.5 Customer branch search and branch rating

5. Lessons Learned

Lesson 1: Building a Graphic User Interface application

Lesson 2: Validate user inputs

Lesson 3: Design the software based on the user's/client's needs properly

Lesson 4: Introduce usability to our UI application by making sure that following are met [1]:

a. Learnability

b. Usage efficiency

c. Error Handling

d. Acceptability

Lesson 5: Make the app as simple as possible without wasting resources and time [1]

Lesson 6: Responsive to user inputs

Lesson 7: Pull members' changes before adding work to avoid version conflicts

Lesson 8: Leaned how to form a UML diagram and link the relationships between the classes

6. References

[1] H. Al-Osman, “Focusing on Users and Their Tasks,” *Object-Oriented Software Engineering Practical Software Development using UML and Java*, chapter. 7, Sep. 2020.