

SEG 2105 Project Report

Gong(Victor) Feng, 300176400

Jenson Wu, 300166874

Quinn Parker-Joyes, 300170098

Kian Ashrafganjouei 300173780

Table of Contents

Topic	Page #
Introduction	2
Contribution to the Project	3
Sign Up and Sign in for customer account	4
Sign up for branch/employee Account	4
Validation of Sign up page	5
Welcome Page for all accounts	6
Admin Page and Manage Account	7
Service Page and Create and Delete Service	7
Display Service for Edit Service and Edit Service	8
Validation of Admin functionalities	9
Cannot delete service that doesn't exist	9
Branch Page before and after adding service and editing profile and working hours	10
Validation profile manager	10
Edit Profile and Working and add/delete Branches services	11
Customer Page	11
Search by address, working hours and services	12
Create Service Request	13
Approve/Reject Request (Employee)	13
View Request's Status (Customer)	14
Rate a branch	14
View Rating	15
Lessons Learned	15

Introduction

We have developed a new car dealership application that allows customers, branches and the admin to interact with each other through this application. This application allows the user to create a customer account and employee account with a built in admin account with the following credentials : username = admin, and password = admin. In this application, the admin has two different types of functionalities that they can perform. The first one is a managed account, where they can delete any type of account. The second one is managed services, where they can add , delete and change existing services that are offered by the company.

Once an employee account is created, the application will ask for the user to enter info about the branch such as location, phone number and working hours before they are allowed to sign into their account. Once an employee is signed in, they can later change these info whenever they want. The employees are also allowed to offer services that were added by the admin and delete services that they no longer want to access. They can also approve and reject service requests that are created by customer accounts and notify the customer of that decision.

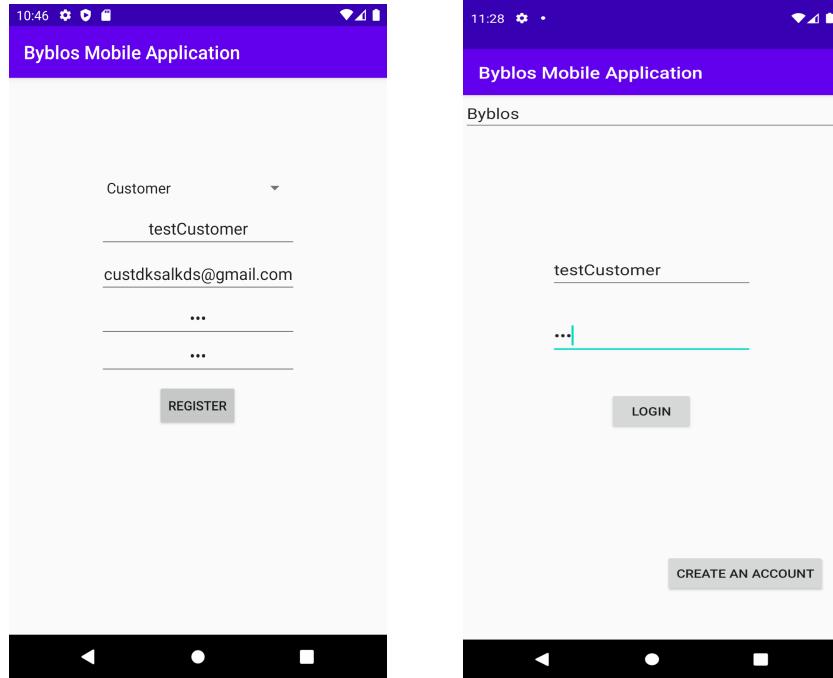
Once a customer logs into their account, they are able to search for branches that they want to do business with and create service requests based on their selection of their branch. They are latered notified when the status of their service request has changed. They can also rate and leave comments to any branch that they desire, while also being able to view the comments that another customer has on each branch.

Contribution to the Project

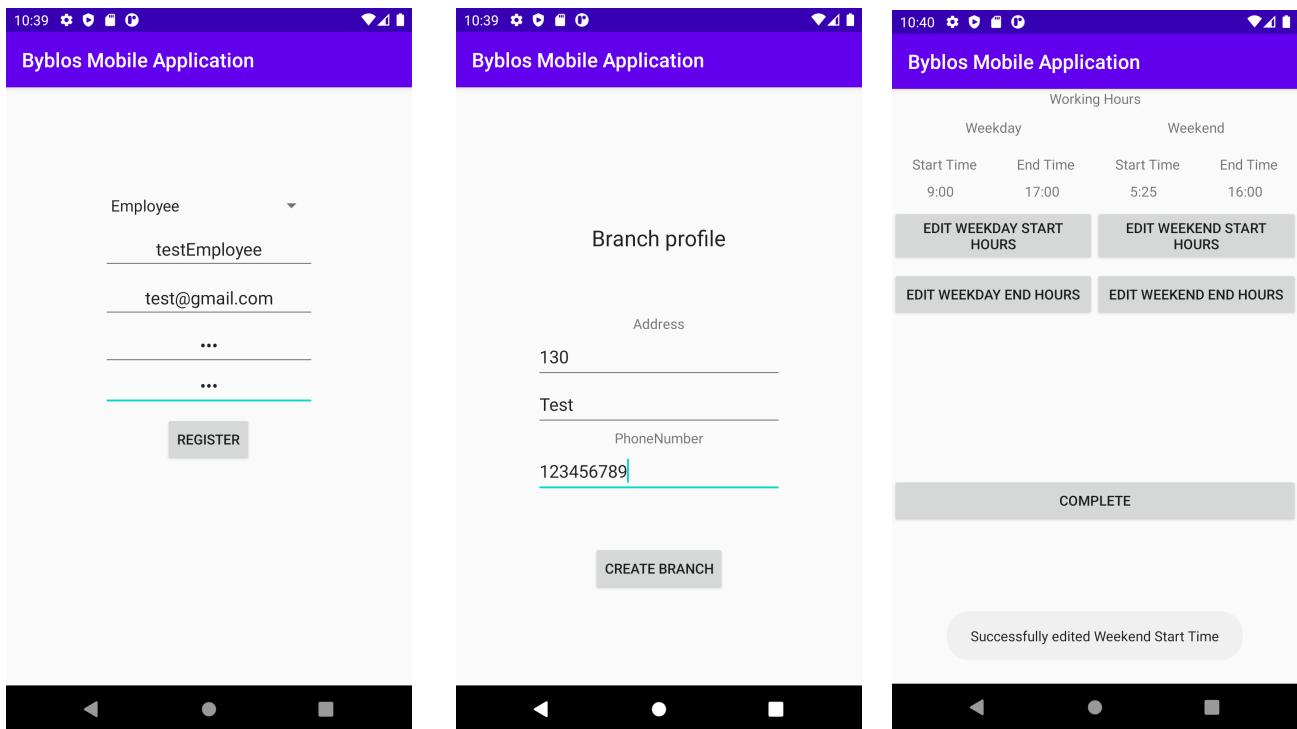
Tasks	Jenson	Victor	Kian	Quinn
Deliverable 1				
Sign Up	X			
Login		X	X	
Setting up firebase		X		
Welcome Page				X
Deliverable 2				
Create admin services	X			X
Edit admin services			X	
Delete admin services		X		X
Delete employee and customer accounts	X			
Admin Page		X		
Deliverable 3				
Create branch services		X		
Delete branch services		X		
Edit profile			X	
Working hours	X			
Approve/Reject Request	X			
Employee/Branch Page		X		
Deliverable 4				
Search for branches		X		
Create requests	X			
Rate branches			X	
Customer Page	X			

Screenshots of our application

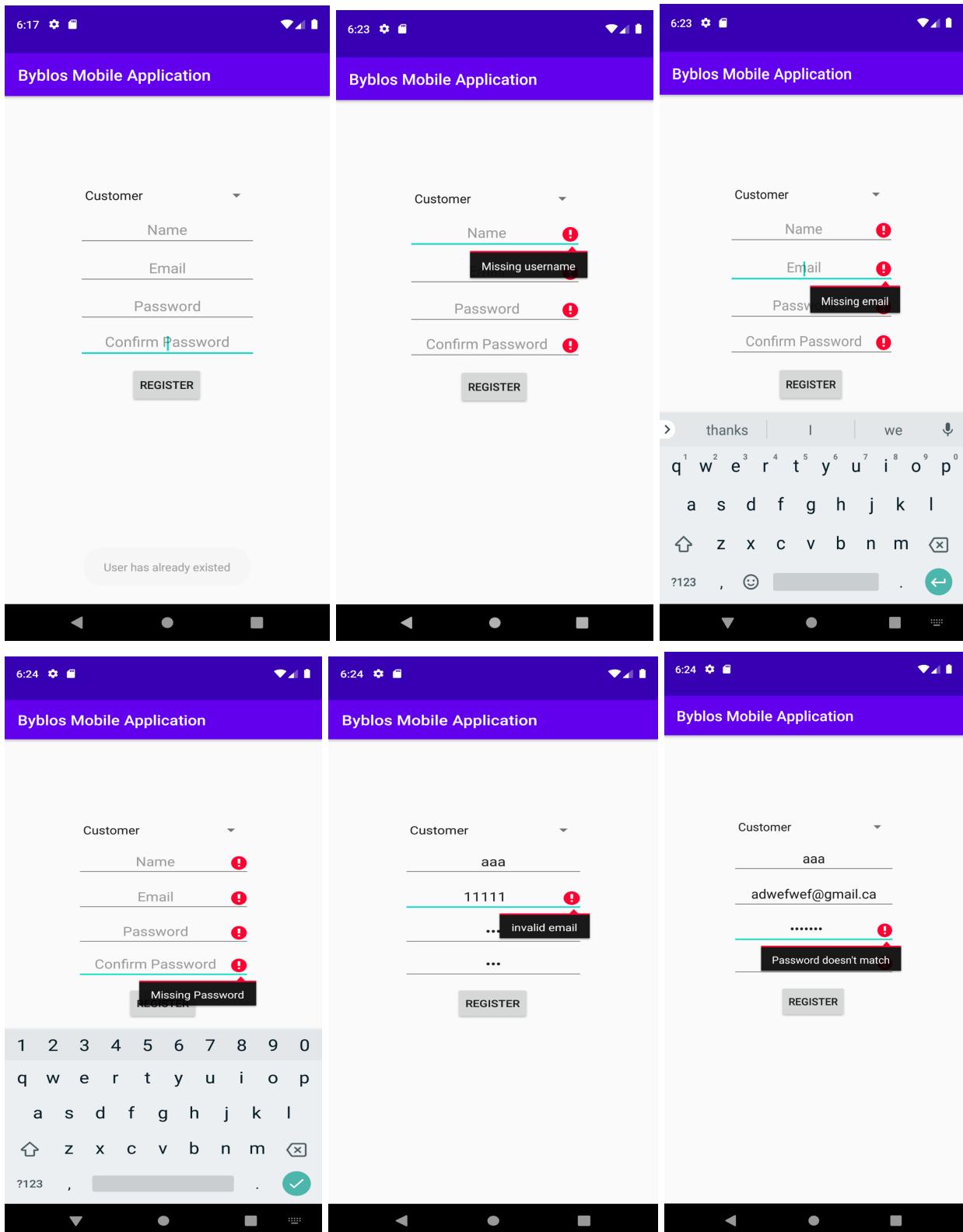
Sign Up and Sign in for customer account



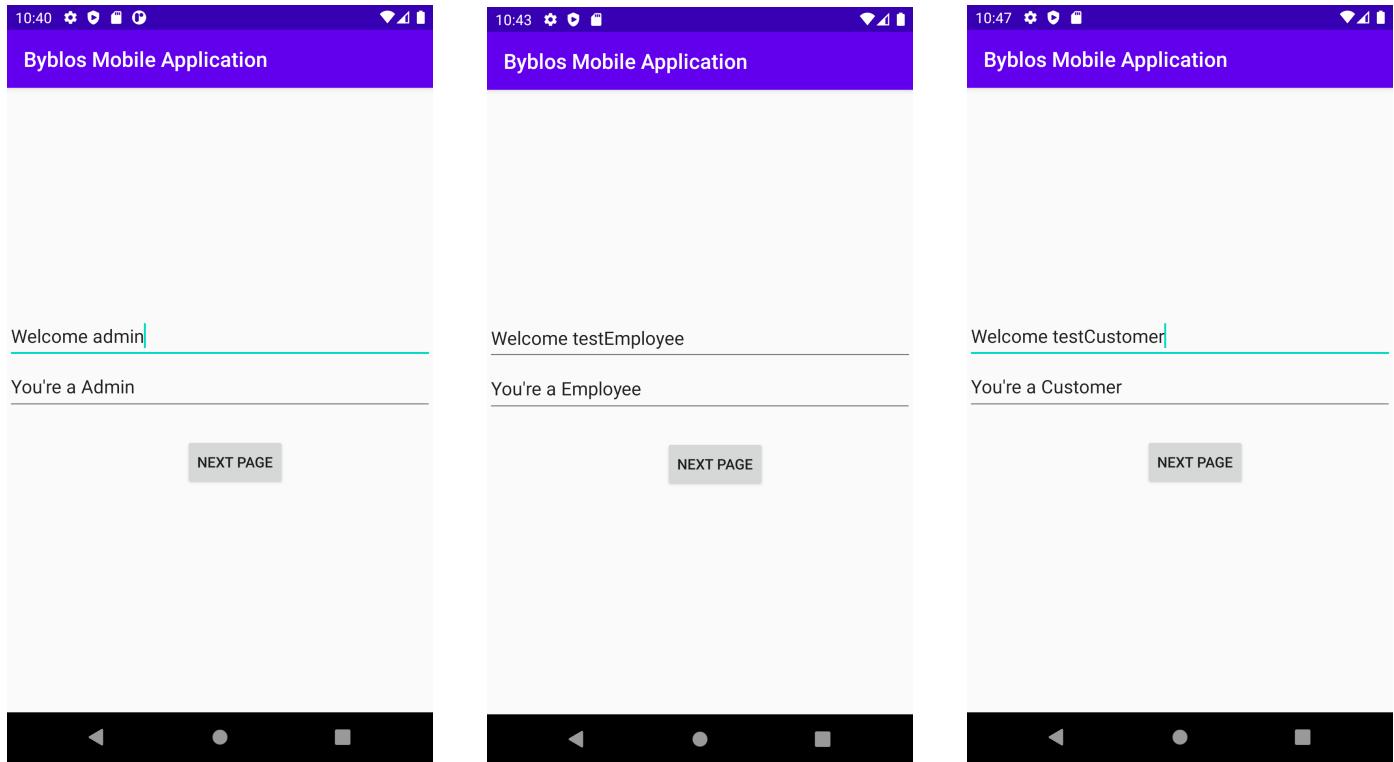
Sign up for branch/employee Account



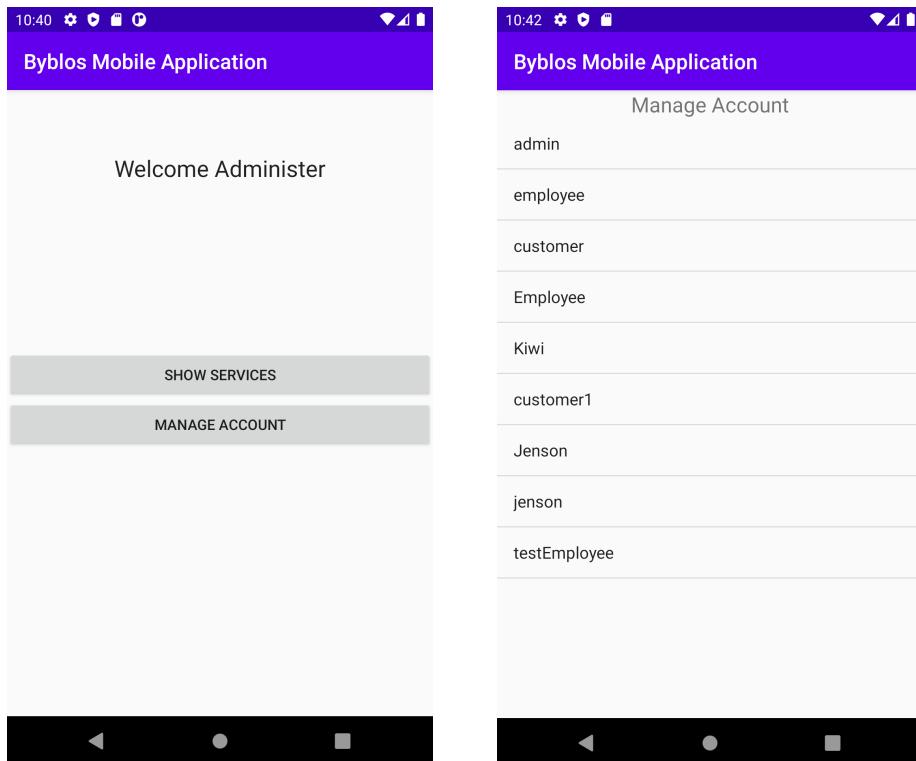
Validation of Sign up page:



Welcome Page for all accounts



Admin Page and Manage Account



Service Page and Create and Delete Service

The image displays three screenshots of the Byblos Mobile Application interface:

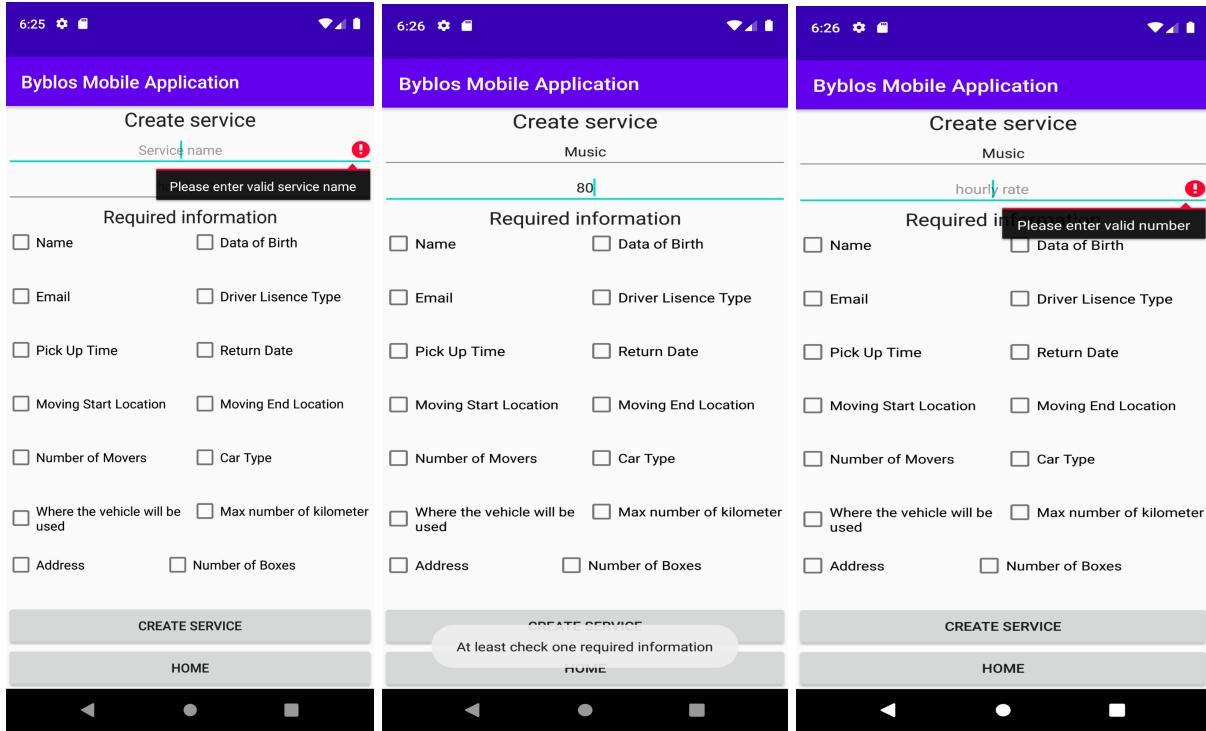
- Left Screenshot:** Shows the "Manage Services" screen with options to "ADD SERVICE", "DELETE SERVICE", and "EDIT SERVICES".
- Middle Screenshot:** Shows the "Create service" screen for a "TEST Service". It includes a section for "Required information" with fields for Name (checked), Data of Birth (checked), Email (unchecked), Driver Liscence Type (checked), Pick Up Time (unchecked), Return Date (unchecked), Moving Start Location (unchecked), Moving End Location (unchecked), Number of Movers (unchecked), Car Type (checked), Where the vehicle will be used (unchecked), Max number of kilometer used (unchecked), Address (unchecked), and Number of Boxes (unchecked). A "CREATE SERVICE" button is at the bottom.
- Right Screenshot:** Shows the "Delete service" screen for a "TEST Service". It includes sections for "carType;" (Moving assitance, Hourly Rate: 15.0, Required infomation: clientName; dateOfBirth; moveStartLocation; moveEndLocation; numMovers; carType; numBoxes), "TEST Service" (Hourly Rate: 12.0, Required infomation: clientName; dateOfBirth; driverLicense; carType), and "Truck rental" (Hourly Rate: 25.0, Required infomation: clientName; dateOfBirth; email; driverLicense; carType; purpose).

Display Service for Edit Service and Edit Service

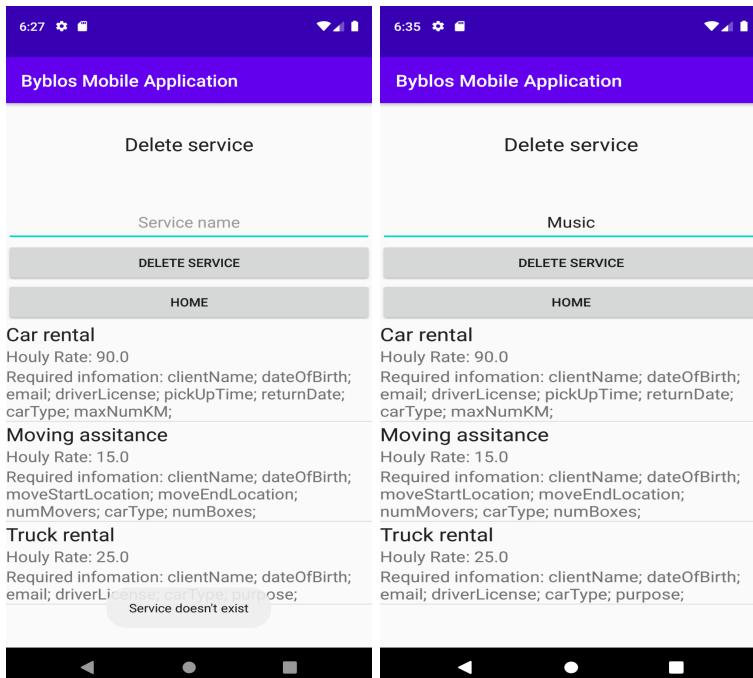
The image displays two screenshots of the Byblos Mobile Application interface:

- Left Screenshot:** Shows the "Display Service" screen with a list of services:
 - Car rental:** Hourly Rate: 100.0, Required infomation: clientName; dateOfBirth; email; driverLicense; pickUpTime; returnDate; carType;
 - Moving assitance:** Hourly Rate: 15.0, Required infomation: clientName; dateOfBirth; moveStartLocation; moveEndLocation; numMovers; carType; numBoxes;
 - Truck rental:** Hourly Rate: 25.0, Required infomation: clientName; dateOfBirth; email; driverLicense; carType; purpose;
- Right Screenshot:** Shows the "Edit service" screen for a "Car rental" service. It includes a section for "Required information" with fields for Name (checked), Data of Birth (checked), Email (checked), Driver Liscence Type (checked), Pick Up Time (checked), Return Date (checked), Moving Start Location (unchecked), Moving End Location (unchecked), Number of Movers (unchecked), Car Type (checked), Where the vehicle will be used (unchecked), Max number of kilometer used (checked), Address (unchecked), and Number of Boxes (unchecked). An "EDIT SERVICE" button is at the bottom.

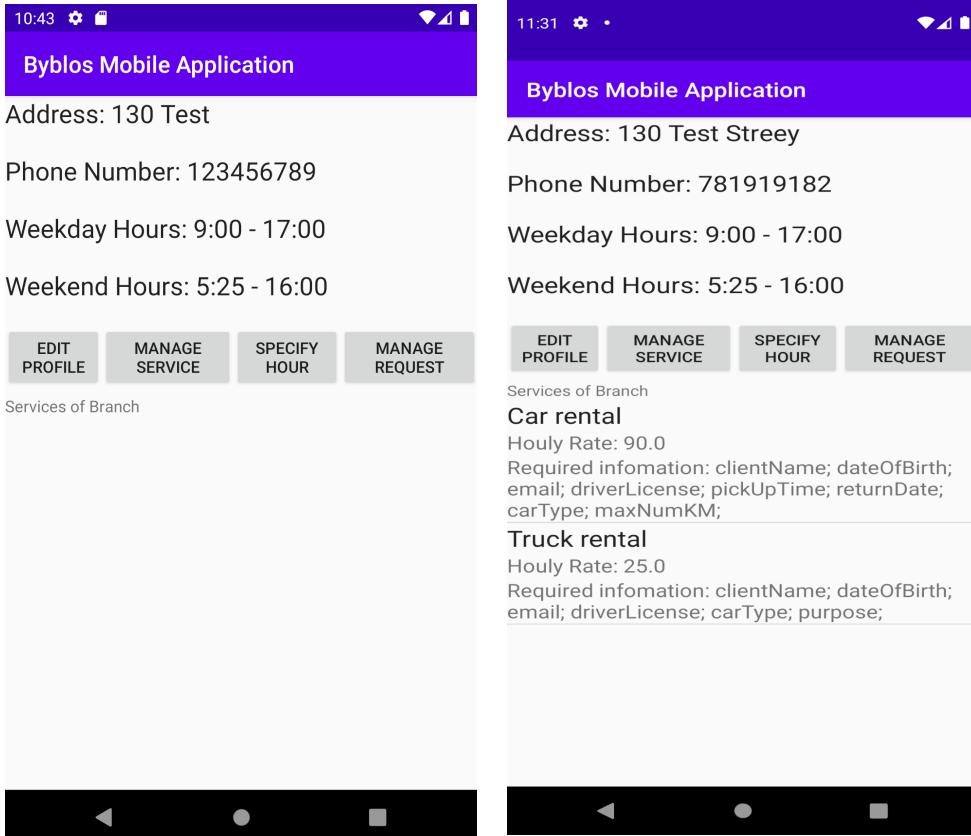
Validation of Admin functionalities:



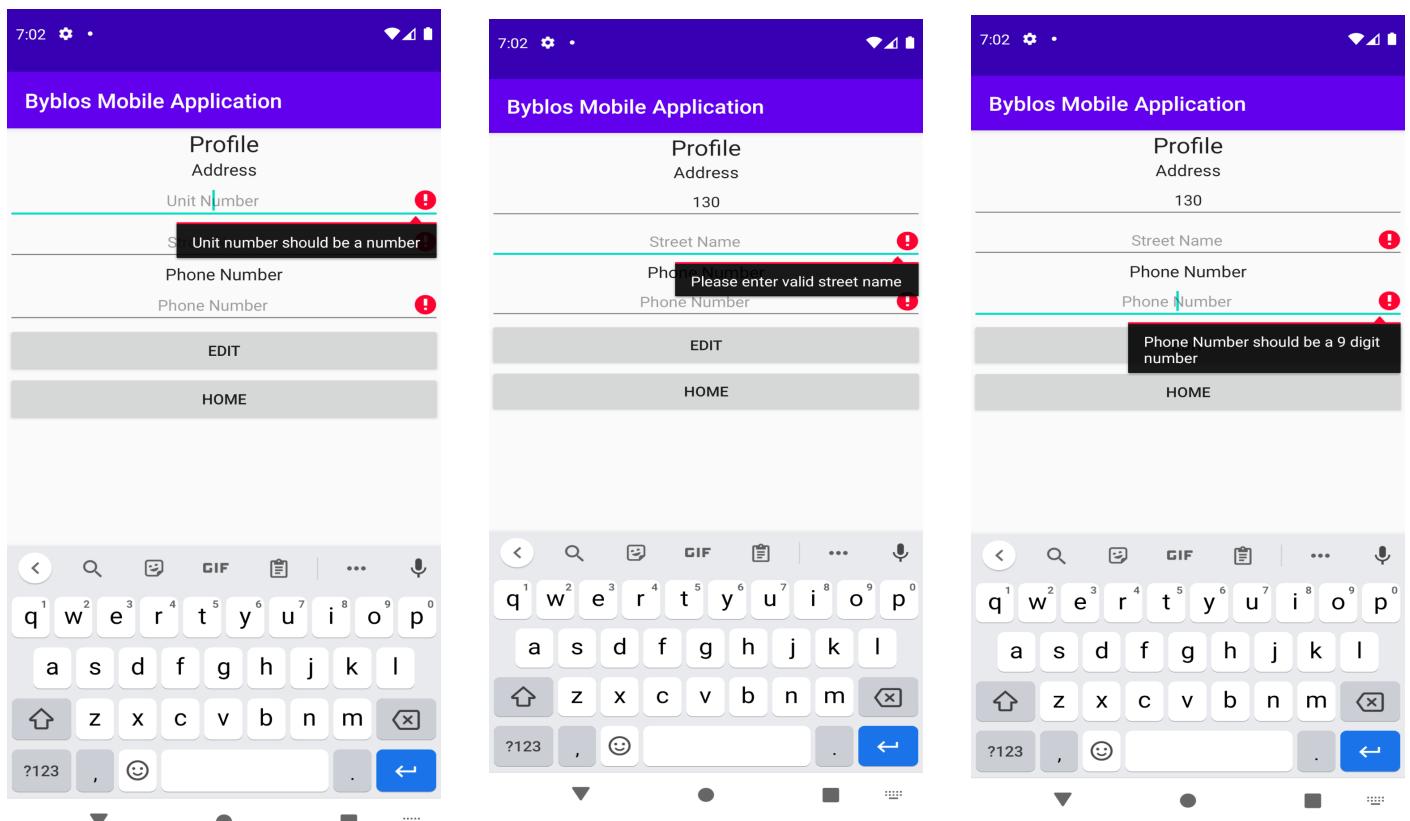
Cannot delete service that doesn't exist.



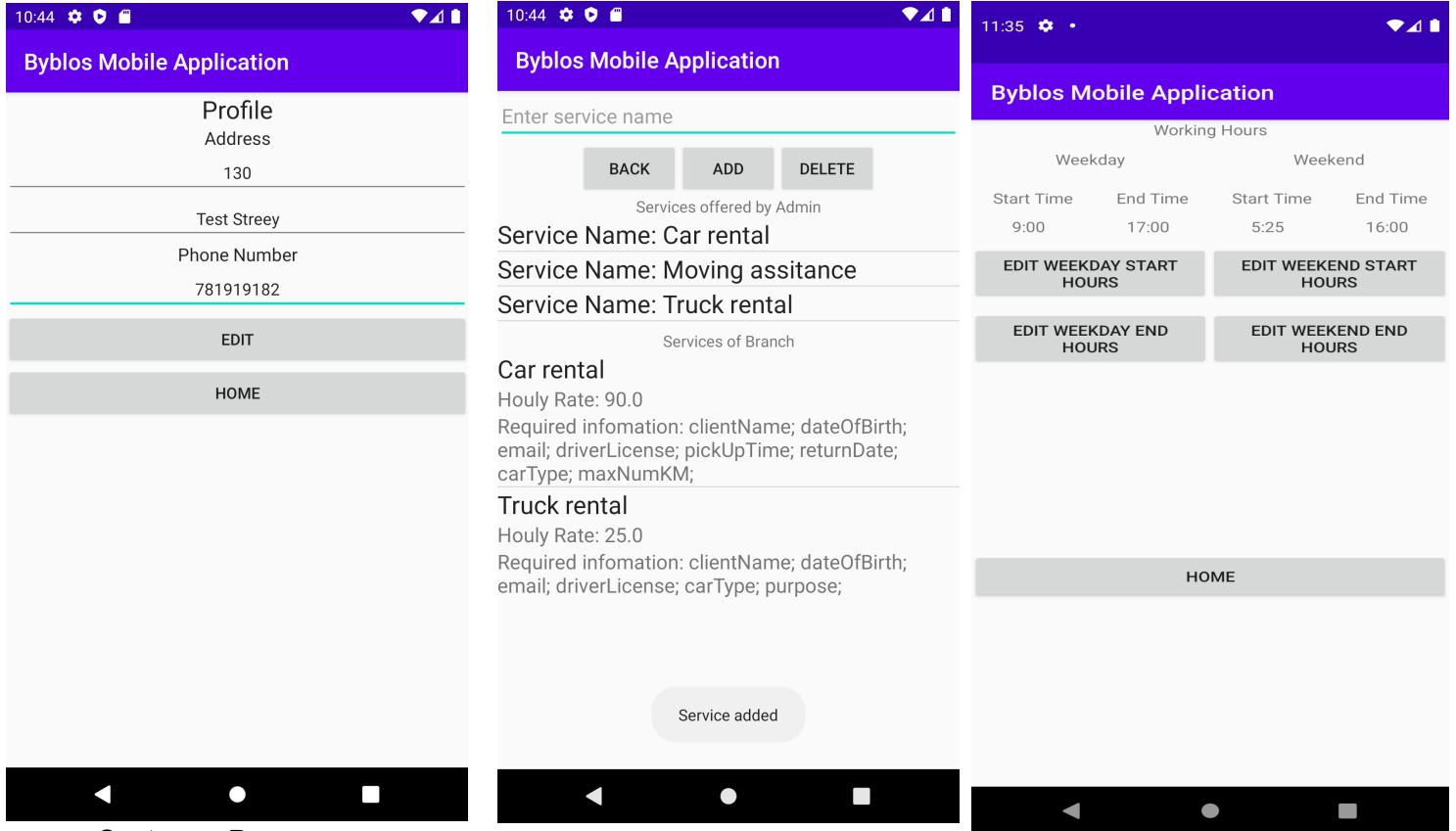
Branch Page before and after adding service and editing profile and working hours



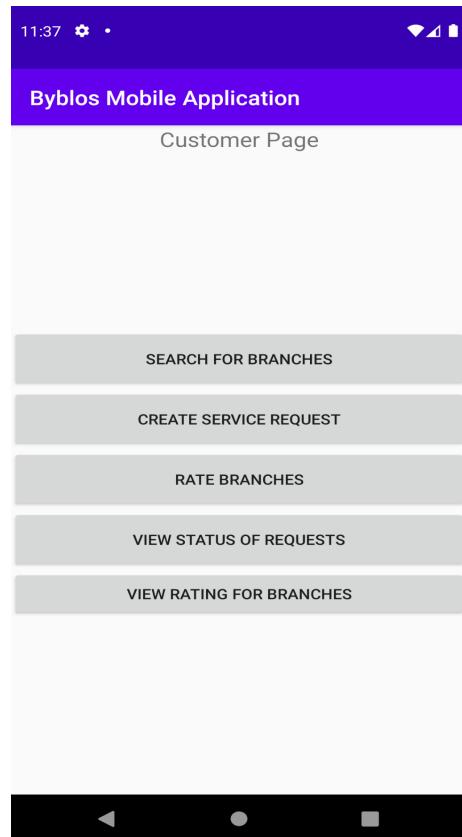
Validation for Profile Manager



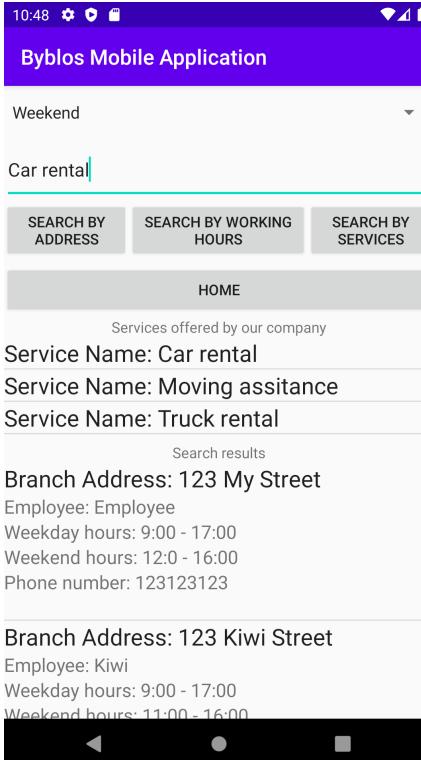
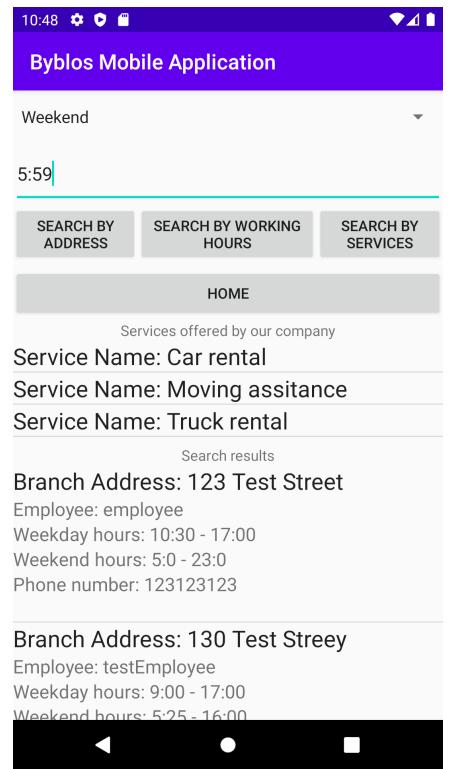
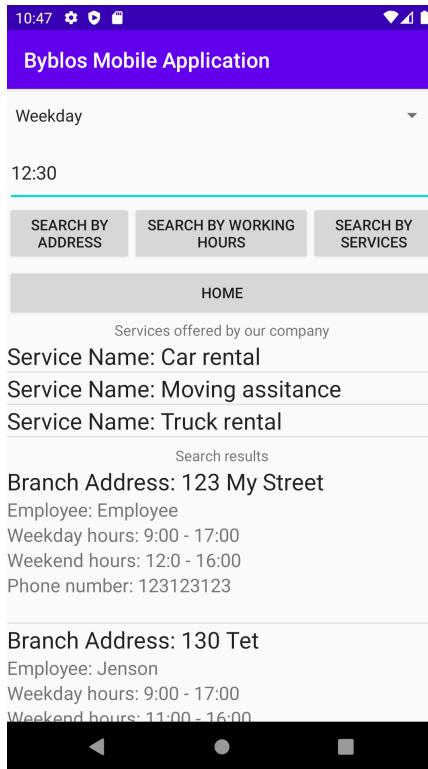
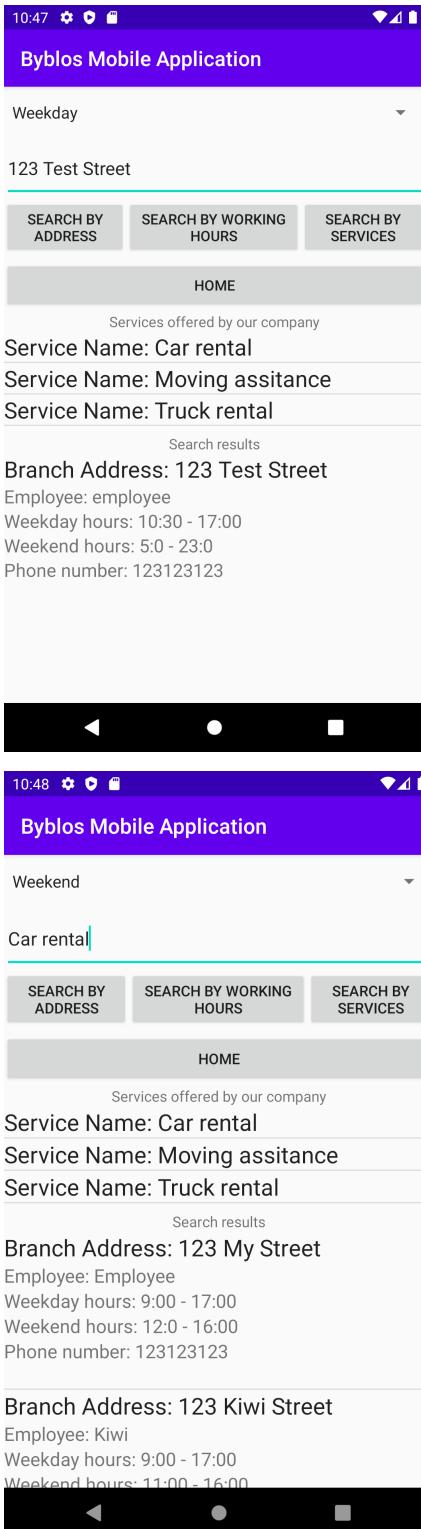
Edit Profile and Working and add/delete Branches services



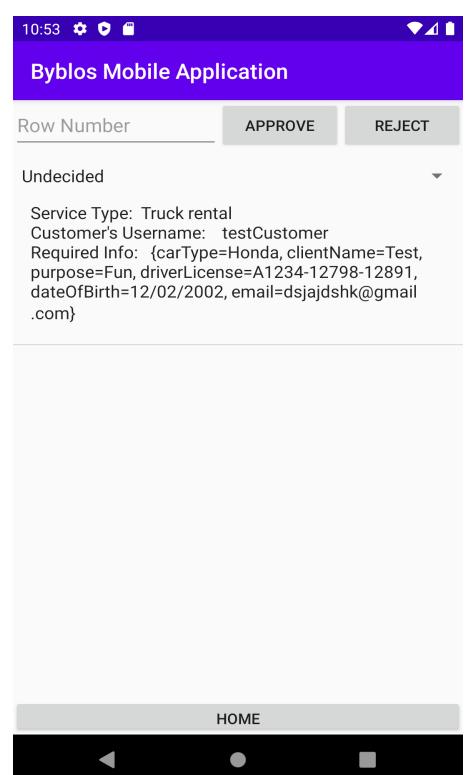
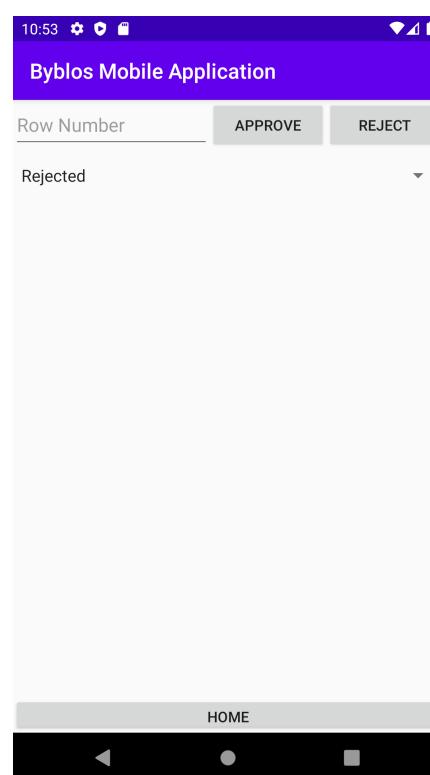
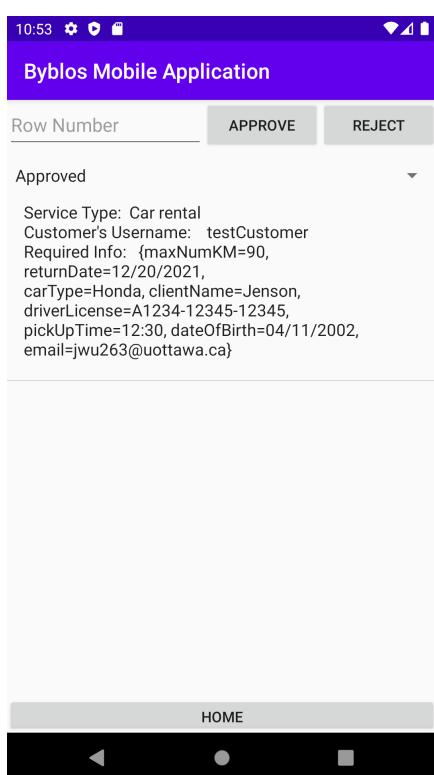
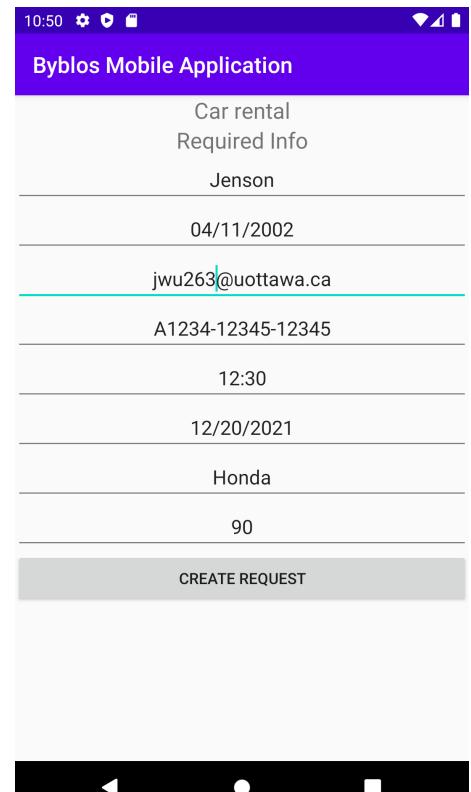
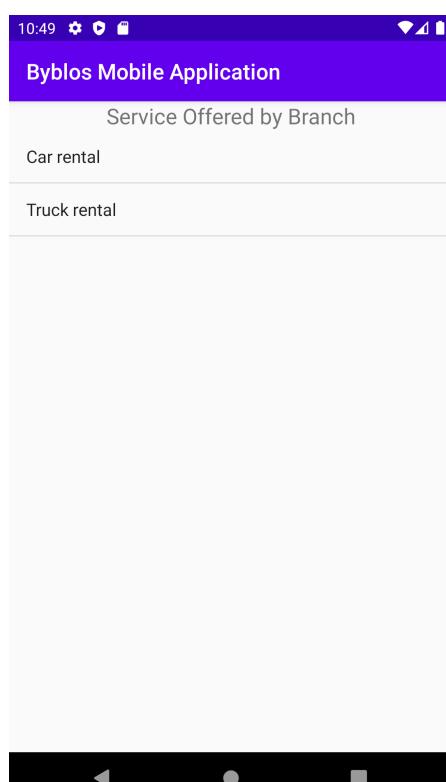
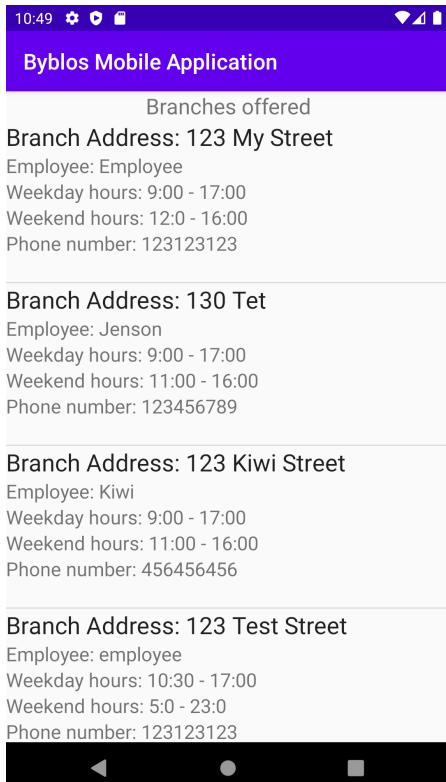
Customer Page



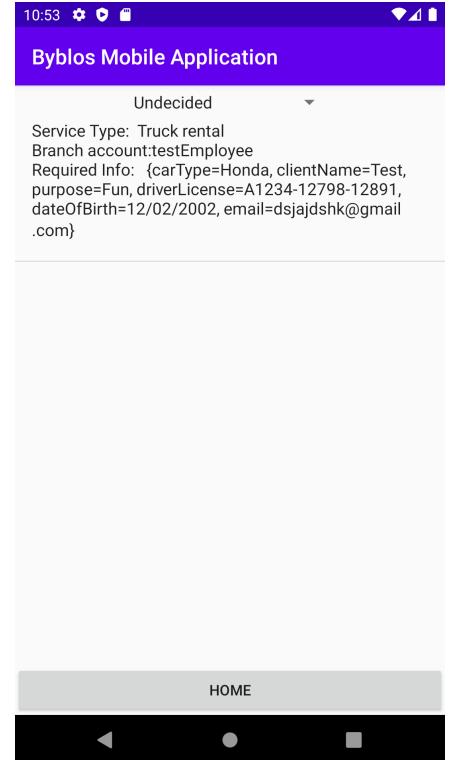
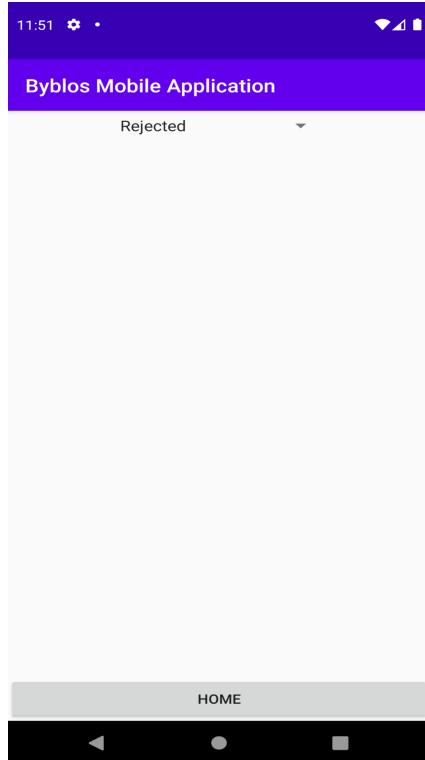
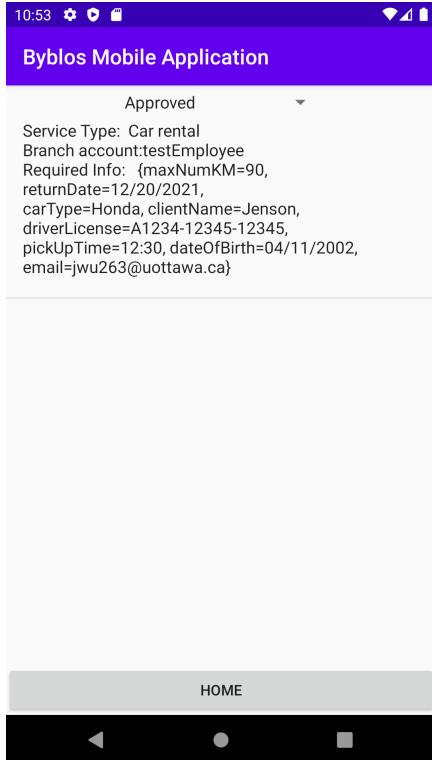
Search by address, working hours and services



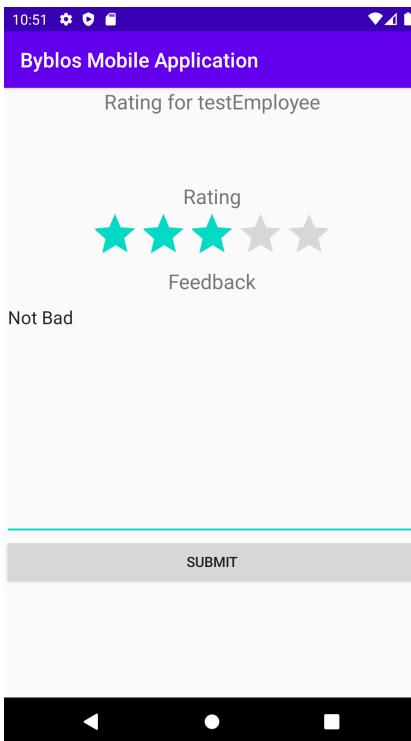
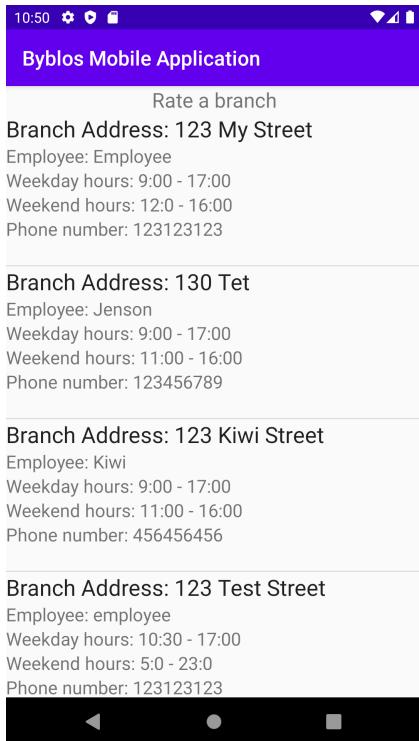
Create Service Request



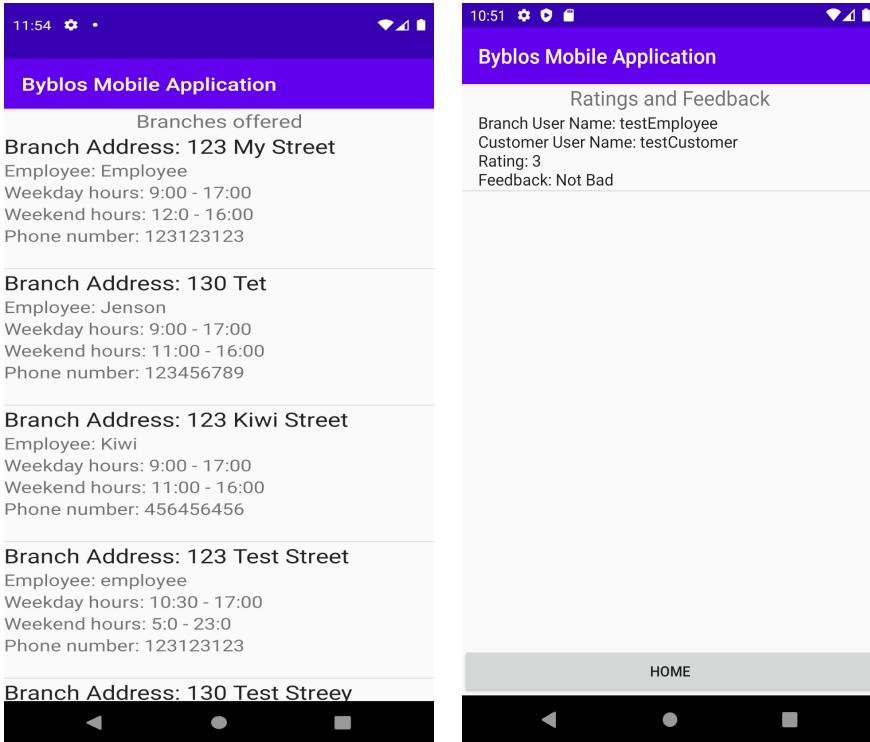
View Request's Status (Customer)



Rate a branch



View Rating



Lesson Learnt

Throughout developing our application, we learned several lessons. Firstly, we learned that we needed to use git properly to avoid merge conflicts and outdated source code issues. Git allowed us to stay updated with one another's changes and helped increase the development rate.

Secondly, we learned that while switching back and forth between different activities in our Android application, it is important to finish the previous activity before starting the next. Doing this allowed us to avoid unwanted behavior from the application and better manage the flow of the program.

Thirdly, we learned how to design UML diagrams before coding our solution. This tremendously helped us in getting a big picture understanding of the application before starting the coding process. Designing the UML diagrams also helped us avoid design flaws as we were solely focused on the design aspect of the application, not the implementation.

Moreover, we also learned the importance of using Firebase properly to store, manage and retrieve the data. Also, maintaining the database correctly simplified our interactions with the database and helped avoid potential problems.

While designing the UI for the application, we learned to leverage the many widgets that come with the Android Studio environment and to create a better user experience. We also learned how to make the UI more responsive and intuitive to the user by gathering input with the appropriate widgets .

While developing the application, we learned to use the design patterns that we had learned to our advantage. Firstly, we used the observer pattern to perform the appropriate event handling in our application. Furthermore, we used the facade pattern by using the sign up page as an entry point to access the other complex components.

Lastly, we learned to implement test cases as a part of our debugging process to ensure that the application behaves as expected. This allowed us to prevent mistakes in the code and made sure the functionalities of the app met our requirements.

