Service Novigrad (A.R.M.Y.S.)

SEG2105A – Introduction to Software Engineering

Fall 2020
School of Electrical Engineering and Computer Science
University of Ottawa

Shaan Atwal (300078472)

Ayman Fakri (300120735)

Rachel Wendling (300007087)

Yassine Moumine (300140139)

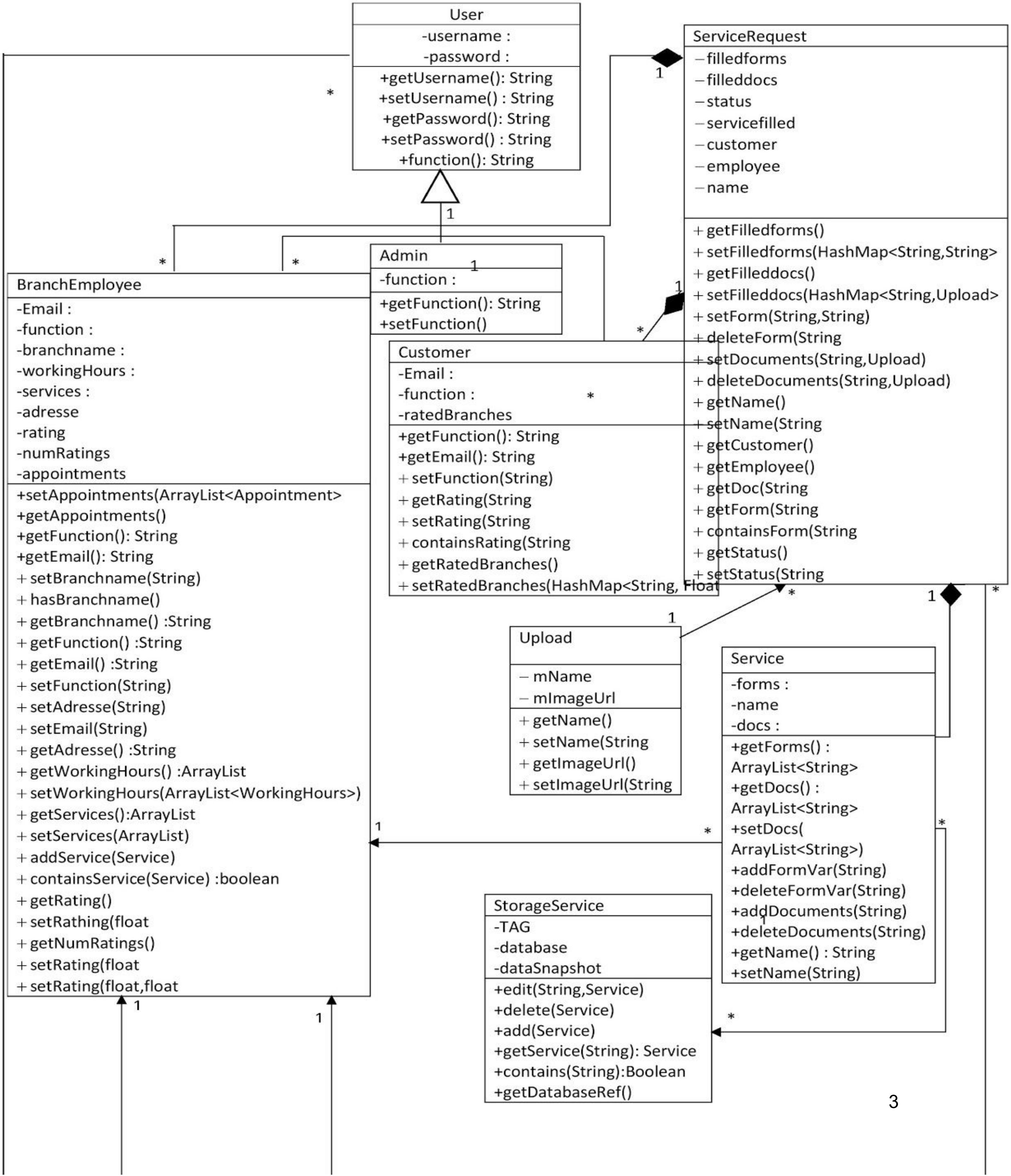
Mohammed Oussalah (300119977)

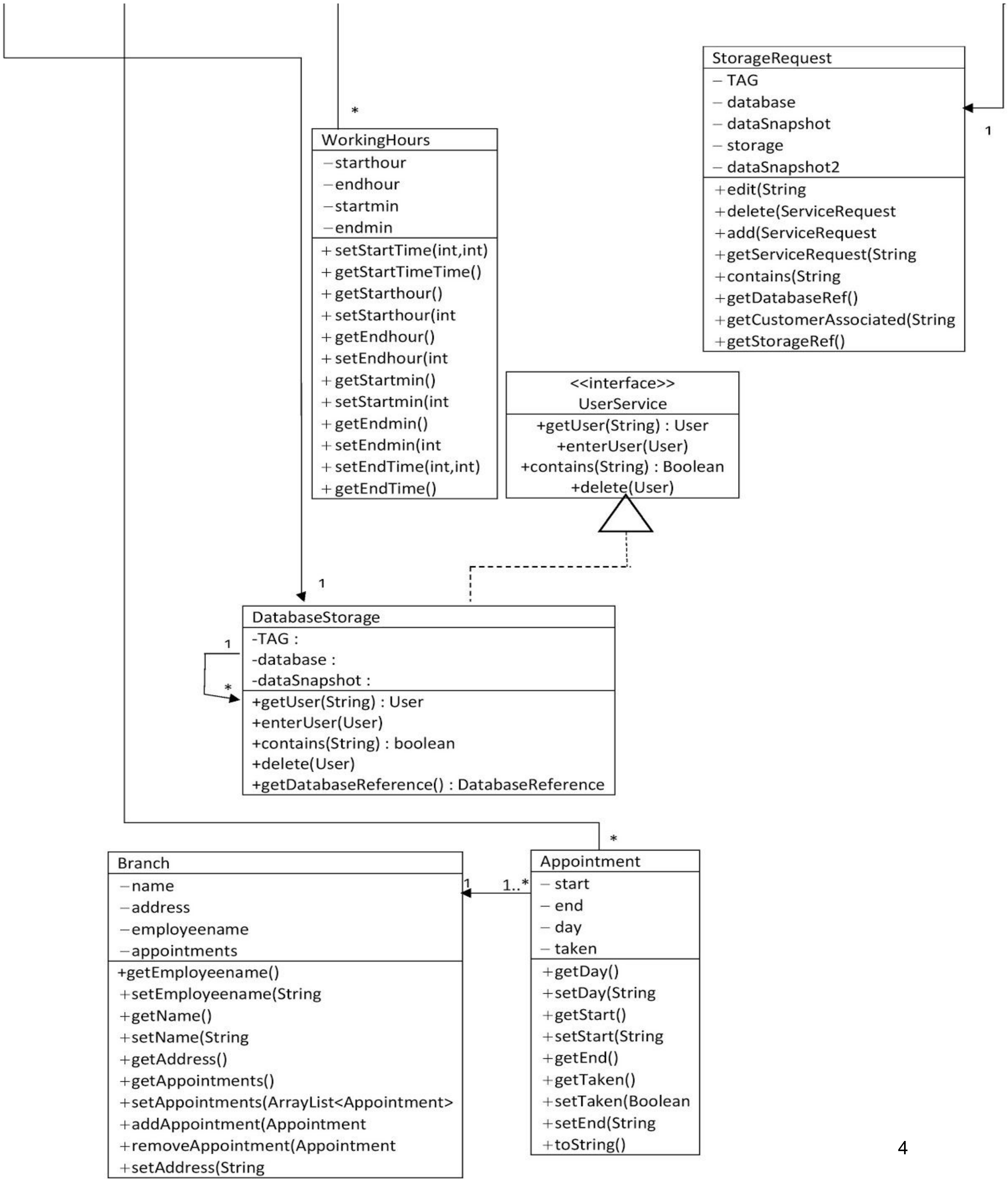
Course Coordinator: Hussein Al Osman
Teaching Assistants: Olivia Borel, Faezeh Halabian, Md. Aminur Rab Ratul

Introduction

The Service Novigrad application implemented offers services to the imaginary citizens of the province Novigrad. The application allows for two types of accounts to be created, branch employee and customer. In addition, three types of accounts can log in including a premade account administrator which uses the username admin and password 123admin456. Each account has different features. The administrator can create and delete services along with deleting users (employees and customers). The employees are associated with a branch and can choose services the administrator created. The employers have additional capabilities including choosing working hours, viewing the branch rating, selecting appointment times, and accepting or declining service requests. Lastly, the customers can choose a branch and submit a completed service request to the employee associated with the branch. In addition, the customer can view the status of their request, rate a branch, and search for a branch under specifications of working hours, services, address, or name. These features were created in android studio and coded in java. For storage firebase was implemented and unit tests were created to test the application's abilities.

UML


User
-username :
-password :
+getUsername(): String
+setUsername() : String
+getPassword(): String
+setPassword() : String
+function(): String
ServiceRequest
−filledforms
−filleddocs
−status
−servicefilled
−customer
−employee
−name
+getFilledforms()
+setFilledforms(HashMap<String,String>
+getFilleddocs()
+setFilleddocs(HashMap<String,Upload>
+setForm(String,String)
+deleteForm(String
+setDocuments(String,Upload)
+deleteDocuments(String,Upload)
+getName()
+setName(String
+getCustomer()
+getEmployee()
+getDoc(String
+getForm(String
+containsForm(String
+getStatus()
+setStatus(String
Admin
-function :
+getFunction(): String
+setFunction()
BranchEmployee
-Email :
-function :
-branchname :
-workingHours :
-services :
-adresse
-rating
-numRatings
-appointments
+setAppointments(ArrayList<Appointment>
+getAppointments()
+getFunction(): String
+getEmail(): String
+setBranchname(String)
+hasBranchname()
+getBranchname() :String
+getFunction() :String
+getEmail() :String
+setFunction(String)
+setAdresse(String)
+setEmail(String)
+getAdresse() :String
+getWorkingHours() :ArrayList
+setWorkingHours(ArrayList<WorkingHours>)
+getServices():ArrayList
+setServices(ArrayList)
+addService(Service)
+containsService(Service) :boolean
+getRating()
+setRathing(float
+getNumRatings()
+setRating(float
+setRating(float,float
Customer
-Email :
-function :
-ratedBranches
+getFunction(): String
+getEmail(): String
+setFunction(String)
+getRating(String
+setRating(String
+containsRating(String
+getRatedBranches()
+setRatedBranches(HashMap<String, Float
Upload
−mName
−mImageUrl
+getName()
+setName(String
+getImageUrl()
+setImageUrl(String
Service
-forms :
-name
-docs :
+getForms() :
ArrayList<String>
+getDocs() :
ArrayList<String>
+setDocs(
ArrayList<String>)
+addFormVar(String)
+deleteFormVar(String)
+addDocuments(String)
+deleteDocuments(String)
+getName() : String
+setName(String)
StorageService
-TAG
-database
-dataSnapshot
+edit(String,Service)
+delete(Service)
+add(Service)
+getService(String): Service
+contains(String):Boolean
+getDatabaseRef()
*
1
1
*
*
1
1
*
*
*
1
*
1
*
1
*
1
*
1
1
1
*


StorageRequest
– TAG
– database
– dataSnapshot
– storage
– dataSnapshot2
+edit(String
+delete(ServiceRequest
+add(ServiceRequest
+getServiceRequest(String
+contains(String
+getDatabaseRef()
+getCustomerAssociated(String
+getStorageRef()
1
*
WorkingHours
–starthour
–endhour
–startmin
–endmin
+ setStartTime(int,int)
+ getStartTimeTime()
+ getStarthour()
+ setStarthour(int
+ getEndhour()
+ setEndhour(int
+ getStartmin()
+ setStartmin(int
+ getEndmin()
+ setEndmin(int
+ setEndTime(int,int)
+ getEndTime()
<<interface>>
UserService
+getUser(String) : User
+enterUser(User)
+contains(String) : Boolean
+delete(User)
1
DatabaseStorage
-TAG :
-database :
-dataSnapshot :
+getUser(String) : User
+enterUser(User)
+contains(String) : boolean
+delete(User)
+getDatabaseReference() : DatabaseReference
1
*
*
Branch
–name
–address
–employeename
–appointments
+getEmployeename()
+setEmployeename(String
+getName()
+setName(String
+getAddress()
+getAppointments()
+setAppointments(ArrayList<Appointment>
+addAppointment(Appointment
+removeAppointment(Appointment
+setAddress(String
1
1..*
Appointment
– start
– end
– day
– taken
+getDay()
+setDay(String
+getStart()
+setStart(String
+getEnd()
+getTaken()
+setTaken(Boolean
+setEnd(String
+toString()

Team Roles

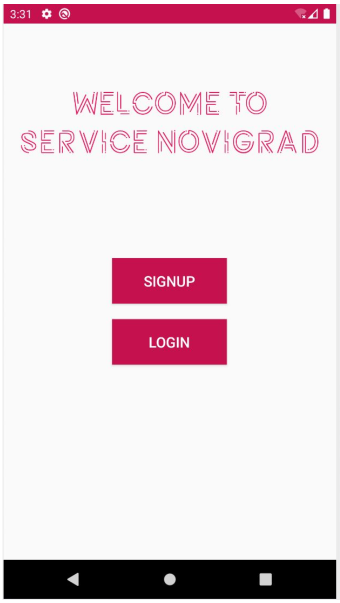
Member	Participation
Ayman Fakri	Worked on: The whole UI design, the sign up for both employee and customer, the login, setting up appointments from the customer side and the branchside, did the branch search feature. Did the working hours search, worked on the app architecture ie : Branch , appointment , branchemployee, service, Workinghoursfilter. I made all of the resource layouts and view holders, did most of the layouts and worked on the database.
Rachel Wendling	Worked on: Designed architecture for backend classes. Codded majority of backend classes seen in UML. Implemented class and layout for the implementation of working hours, rating branch, customers checking status of requests, and the list for employee to see request's made. Also, various layouts, debugging in other areas (such as JUnit and backend classes I didn't code), and written part of the report.
Mohamed Oussalah	Worked on: The signup page, the see users part on the admin, admin able to delete the users. See services available by the BranchEmployee. Employee can add or delete the services. Check forms and documents of requests by Employee. Did the send request by customer. Customer is able to complete the forms upload documents for each Service. Worked on layouts and database for the requests.
Yassine Moumine	Worked on: Updated UML Class diagram for each deliverable.Worked on the implementation of both the class and layout for setting up an appointment as well as the meeting calendar customer side. Had to update the dependencies in our gradle to include the JUnit test which I also worked on.
Shaan Atwal	Worked on: Ensured valid data on signup, login, and a large majority of input fields Completed the Info class and assisted with work on the docs class. Worked on the meeting calendar customer side. Went through almost all of the java files to add comments explaining what each button / method does and returns, and general code explanations where needed. Helped to debug errors and weird glitches. Posted the apps screenshots to the report.

Lessons Learned

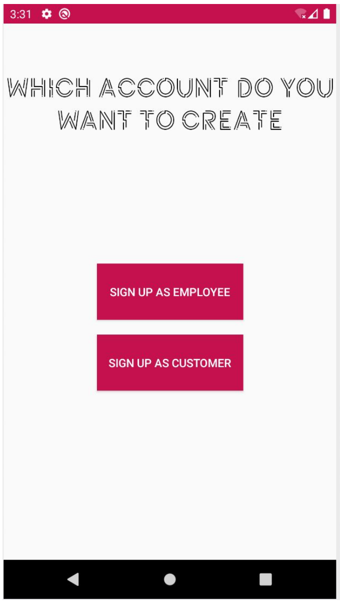
From working on the app the group gained a better understanding of object-oriented analysis and design, software architecture (particularly the client-server architecture), and basic UI design. Firstly, the group learned the ins and outs of android studio and how to create cohesive layouts and connect the classes through the user interface, as well as the standard way of expressing requirements and design in software engineering known as UML. The group also learned firebase and unit tests while implementing them through the app. In addition, during the process of designing the software, the group had to learn to plan ahead and prepare for verifying fields. Lastly, GitHub was used for version control. Working in a large group allowed for a better understanding of GitHub as members had to deal with merger errors and keep up to date on changes made to the repository. The project forced the group to quickly become acquainted with using Git. Overall, the group learned that requirement gathering, specification, testing and most importantly teamwork play a crucial role in the software engineering process.

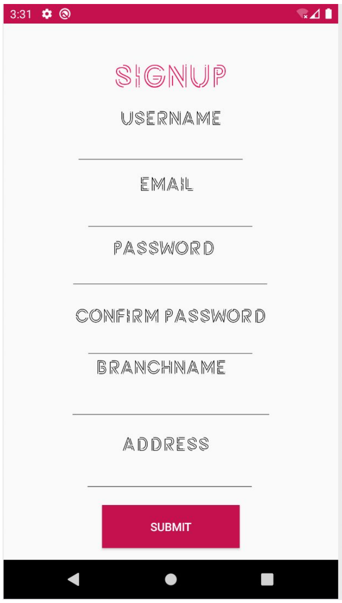
Screenshots

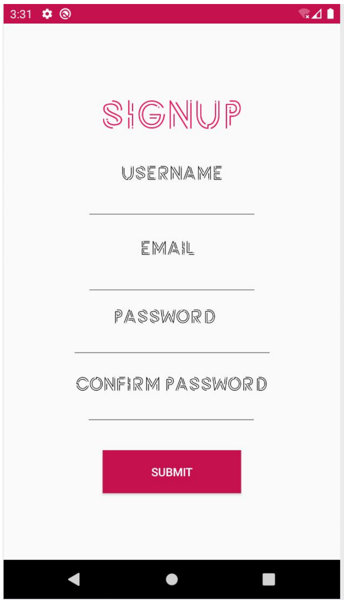
Welcome page :



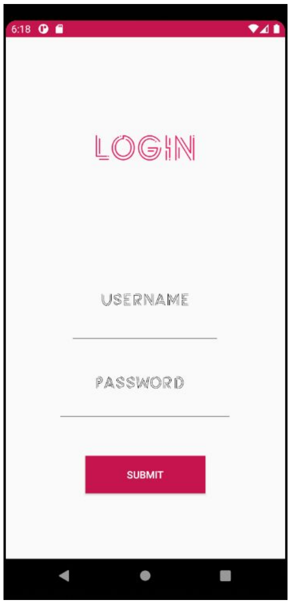
Signup:





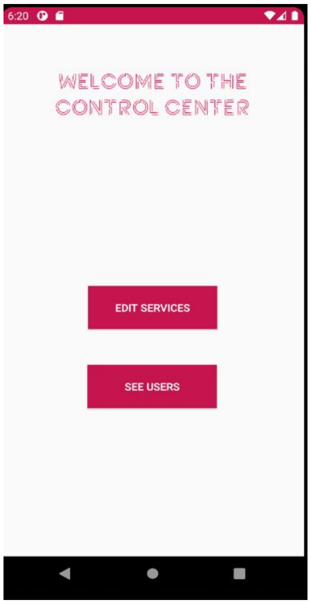


Login:



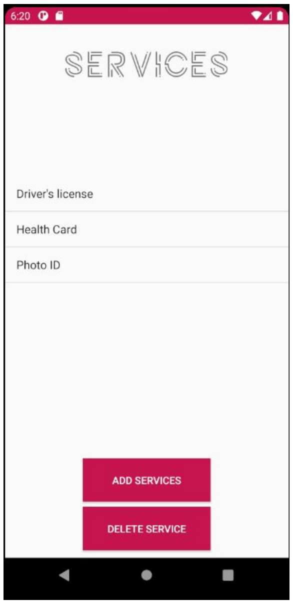
ADMIN:

Control Center



WELCOME TO THE CONTROL CENTER

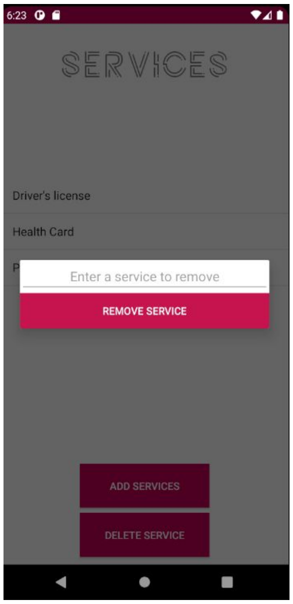
Service

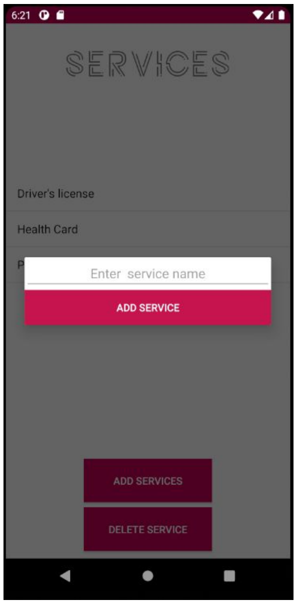


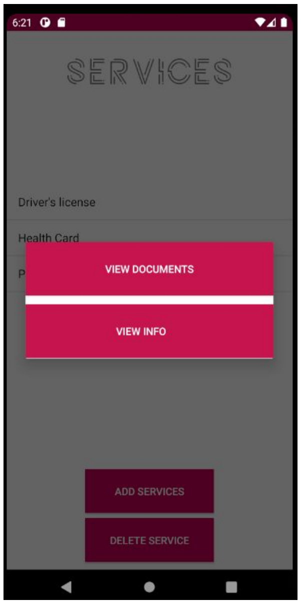
Driver's license

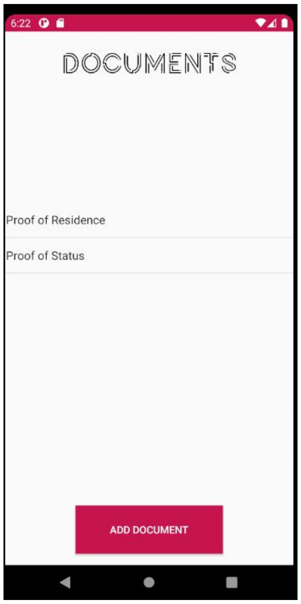
Health Card

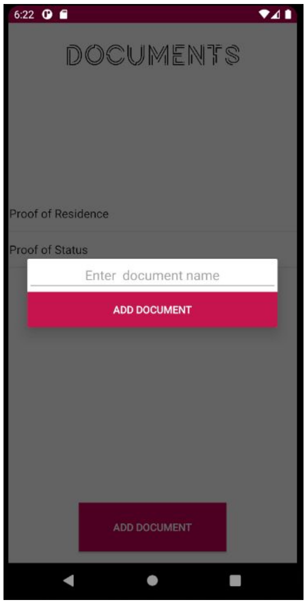
Photo ID

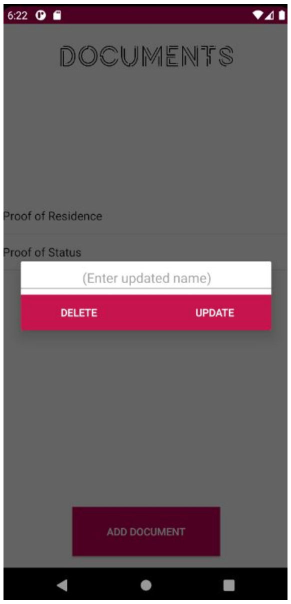




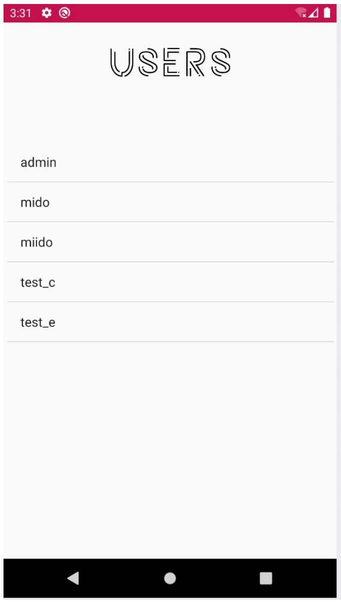

6:21
SERVICES
Driver's license
Health Card
P
VIEW DOCUMENTS
VIEW INFO
ADD SERVICES
DELETE SERVICE


6:22
DOCUMENTS
Proof of Residence
Proof of Status
ADD DOCUMENT


6:22
DOCUMENTS
Proof of Residence
Proof of Status
Enter document name
ADD DOCUMENT
ADD DOCUMENT


6:22
DOCUMENTS
Proof of Residence
Proof of Status
(Enter updated name)
DELETE
UPDATE
ADD DOCUMENT

USERINFO



USERS

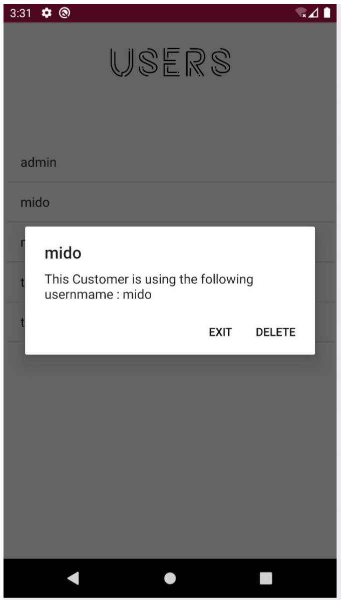
admin

mido

miido

test_c

test_e

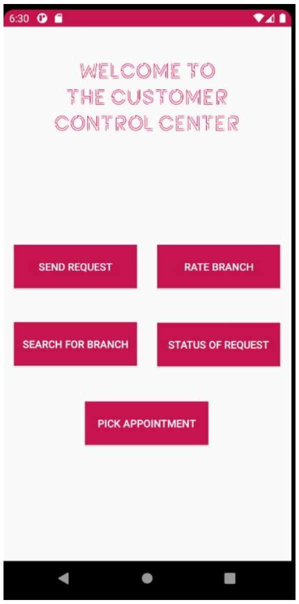


mido

This Customer is using the following usernmame : mido

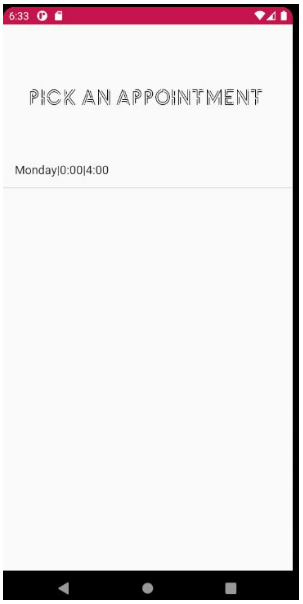
CUSTOMER:

Control Center



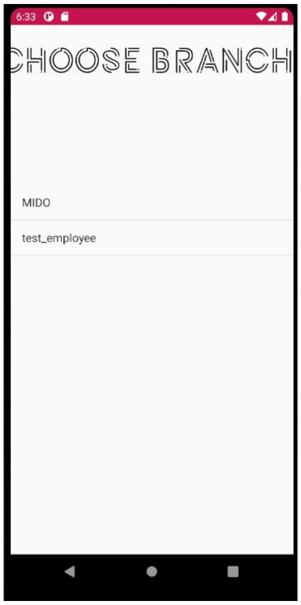
WELCOME TO THE CUSTOMER CONTROL CENTER

APPOINTMENTS



PICK AN APPOINTMENT

Monday|0:00|4:00

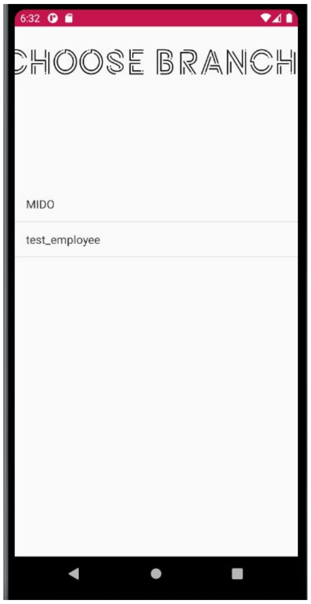


CHOOSE BRANCH

MIDO

test_employee

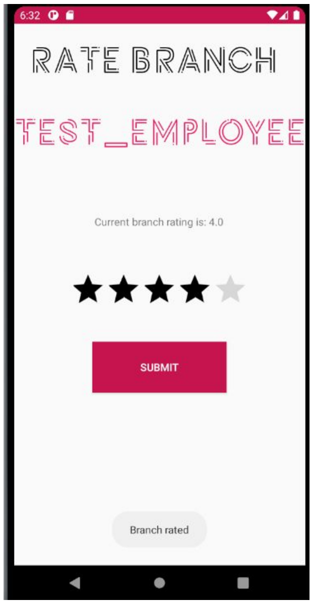
RATE BRANCH



CHOOSE BRANCH

MIDO

test_employee



RATE BRANCH

TEST_EMPLOYEE

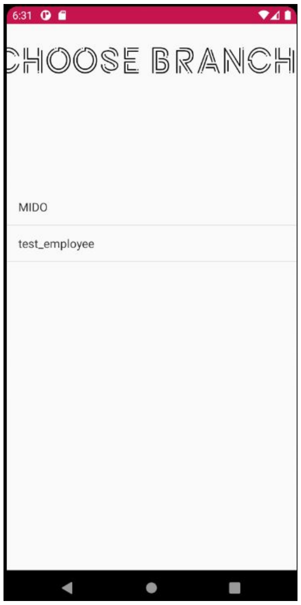
Current branch rating is: 4.0

★★★★☆

SUBMIT

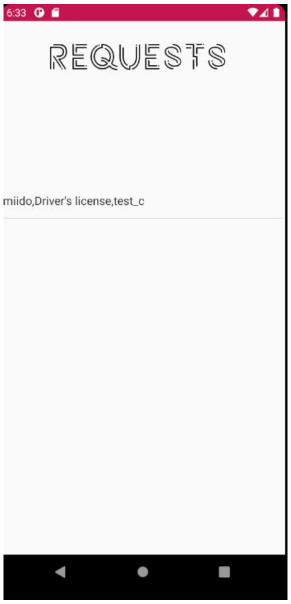
Branch rated

REQUESTS

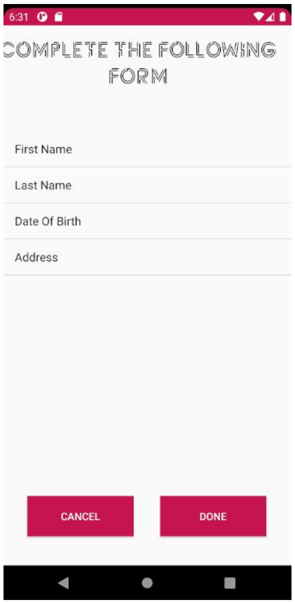

CHOOSE BRANCH

MIDO

test_employee


REQUESTS

miido,Driver's license,test_c

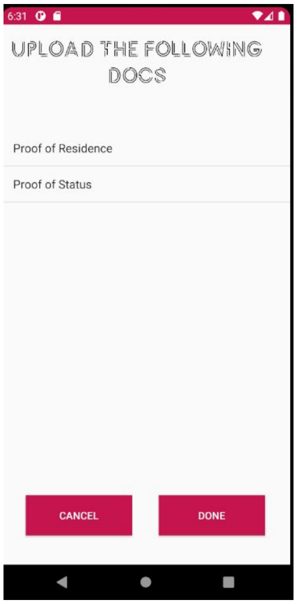

COMPLETE THE FOLLOWING FORM

First Name

Last Name

Date Of Birth

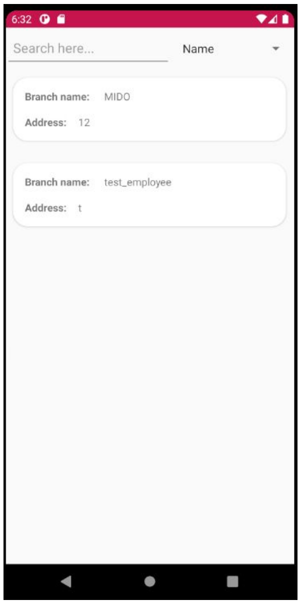
Address


UPLOAD THE FOLLOWING DOCS

Proof of Residence

Proof of Status

Search


Search here...

Name

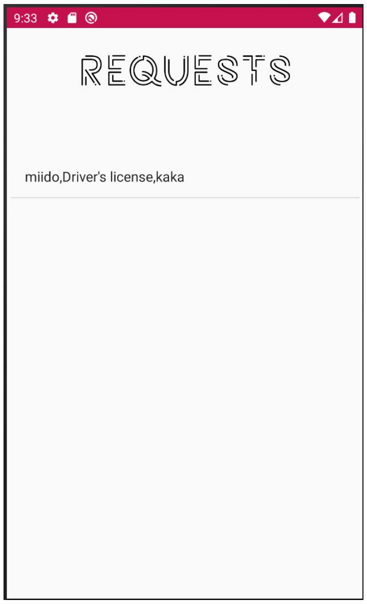
Branch name: MIDO

Address: 12

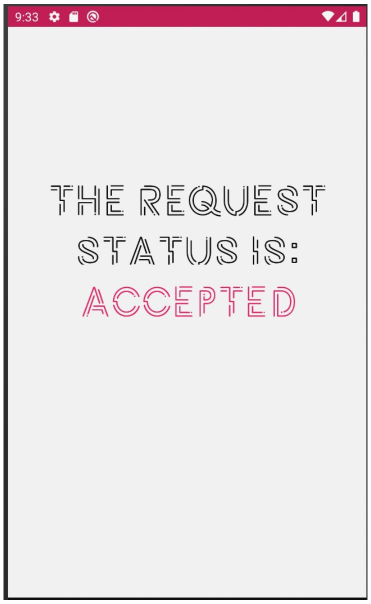
Branch name: test_employee

Address: t

Request Status

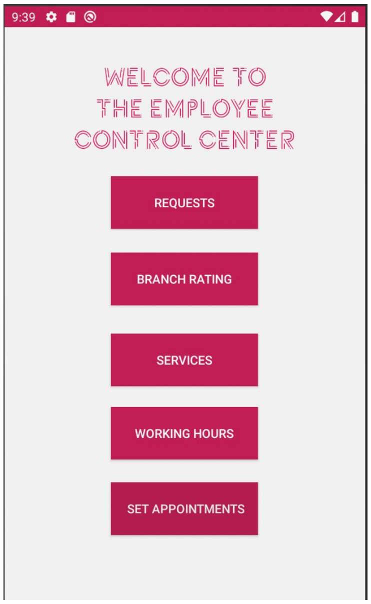

REQUESTS

miido,Driver's license,kaka


THE REQUEST STATUS IS: ACCEPTED

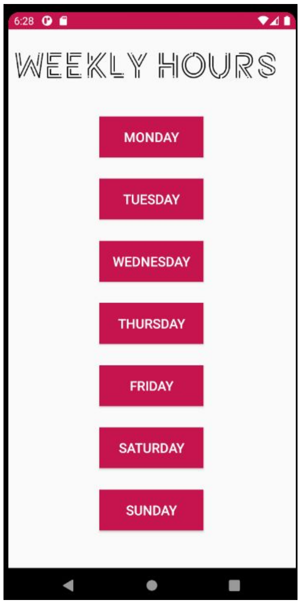
EMPLOYEE:

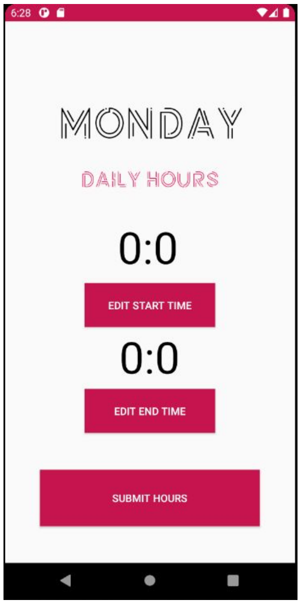
Control Center

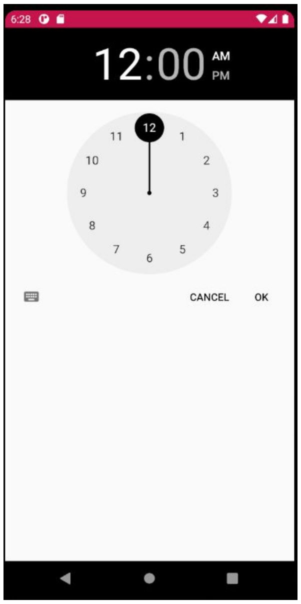


WELCOME TO THE EMPLOYEE CONTROL CENTER

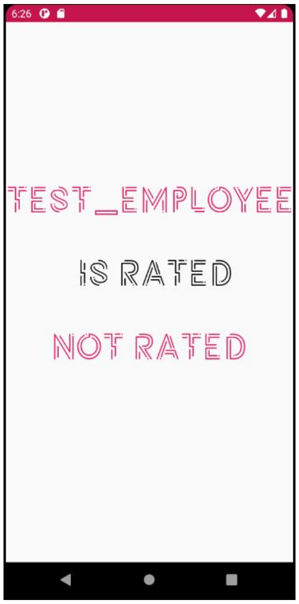
WORKING HOURS







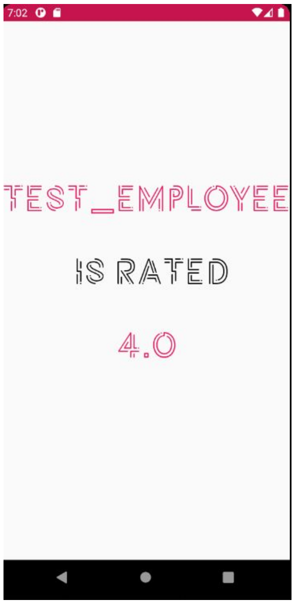
RATINGS



TEST_EMPLOYEE

IS RATED

NOT RATED

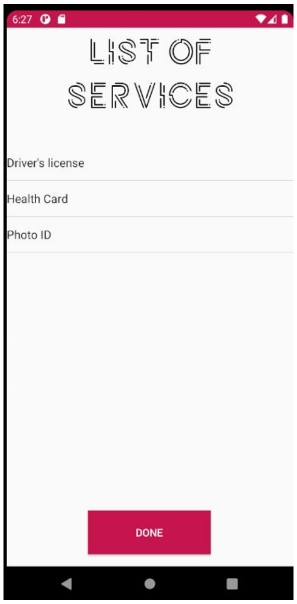


TEST_EMPLOYEE

IS RATED

4.0

SERVICES

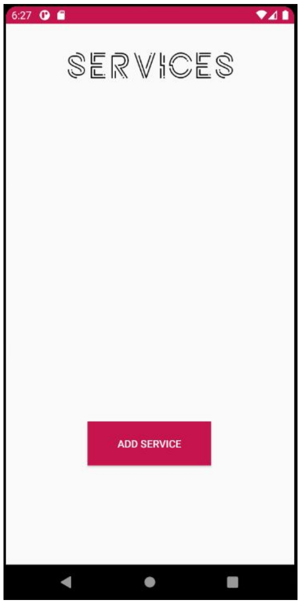


LIST OF SERVICES

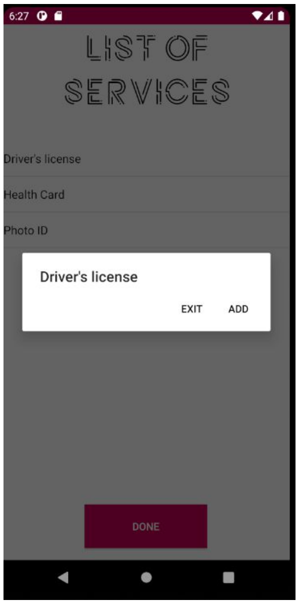
Driver's license

Health Card

Photo ID



SERVICES



LIST OF SERVICES

Driver's license

Health Card

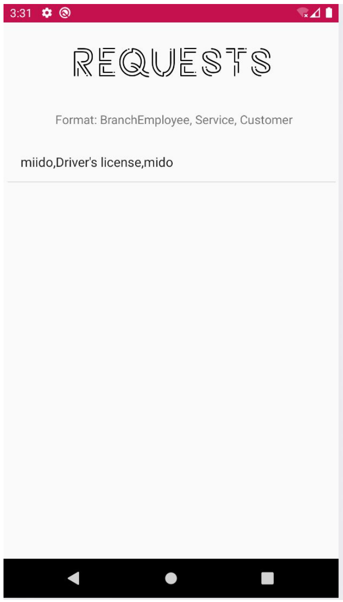
Photo ID

Driver's license

APPOINTMENT

REQUESTS

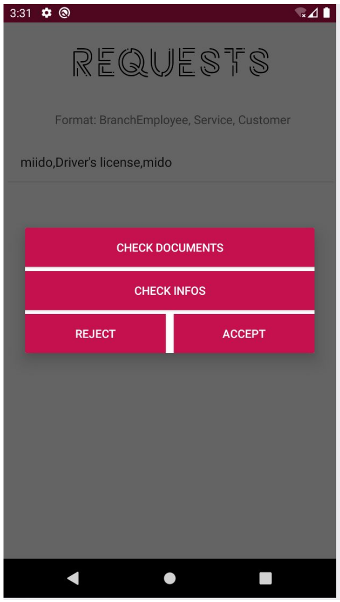
Forms:



REQUESTS

Format: BranchEmployee, Service, Customer

miido,Driver's license,mido

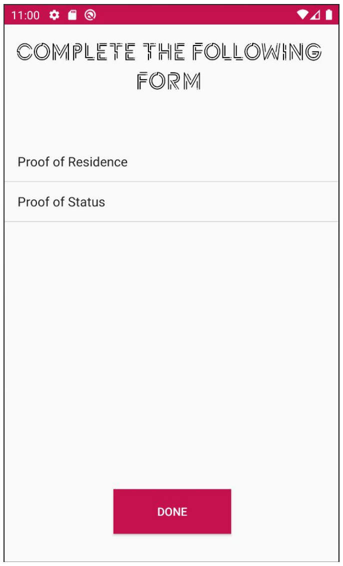


REQUESTS

Format: BranchEmployee, Service, Customer

miido,Driver's license,mido

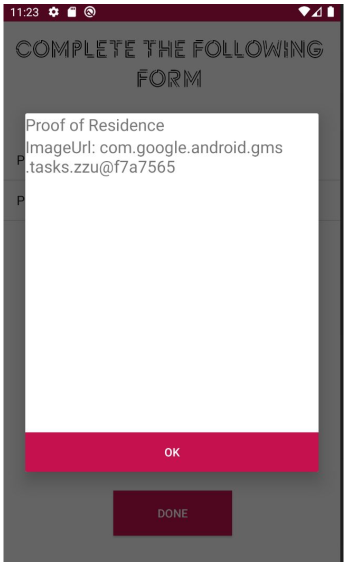
Documents:



COMPLETE THE FOLLOWING FORM

Proof of Residence

Proof of Status



COMPLETE THE FOLLOWING FORM

Proof of Residence
ImageUrl: com.google.android.gms
.tasks.zzu@f7a7565