**DrScanner**

Luis Gonzalez Morell - luis.gonzalez76@upr.edu

Luis Sala Ortiz - luis.sala@upr.edu

**Introduction**

DrScanner is an application to manage patient’s medical files in a more efficient, organized way, while implementing a fast and simple process. All of the patient’s file will be assigned an unique QR Code. When the QR code is scanned, the application will look, inside a SQL server, for the patient’s file associated with said QR code. If the file does not exist, then new information could be entered to be assigned to the scanned QR Code. If it exists, then the user will have the option to look at the patient’s information and edit it. This app is meant to make the patient registration and the retrieval of the patient’s files easier for hospitals and medical offices in general. Since the world has become more “mobile” in the recent years with the boom of smartphones and tablets, it is fitting that DrScanner should be a mobile application, because of the simplicity these provide. The platform to be used for the creation of the application will be Android.  Android is an open source operating system that is distributed on a variety of mobile systems which are accessible and affordable, making the implementation a breeze compared to other conventional technologies. Also, the fact that android runs on most of smartphones today would assure the product a bigger reach.

**Client App Description**

The client side will be based on Android, using native libraries for the user interface and Heroku on NodeJS for web services. There will be two types of users, the Nurse which can only view his own information, and the Admin which will be able to add/view/edit the patients files. It will have a user-friendly interface, having a main screen with options to either scan a file or to access the files on the database. Once a QR Code is scanned, the application will communicate to the database on the background, using AyncTask, and if the file associated with the QR Code is empty or does not exist, the application will ask to inform the user and ask if a new file will be created or not. If the QR Code is already associated with a file with information, then the application will show the user the patient’s file and provide edit options.

**Server Side Description**

The server side will be based on PostgreSQL Database Server. The server will contain various tables to store different information.  One table for each type of user will then give access to the respective tables containing the patient's information. However, a fee will be charged for the server and app maintenance. The email verification of the user is supposed to be implemented using the FireBase service, which provides email verification and other great services. The Databases were implemented and managed using a combination of PGAdmin, where they were originally created and then linked to Heroku, where they were modified via command line. The fee will be charged when the administrator is registering, and the information will then be stored in another table. To give an in depth look at how the tables are arranged and their connections, here we provide the schemas (create table), for each one of them:

create table doctor(

did serial primary key,

dfirst varchar(10),

dlast varchar(10),

specialty varchar(20)

);

create table patient(

pid serial primary key,

pfirst varchar(10),

plast varchar(10),

ssn char(9) unique

);

create table condition(

cid serial primary key,

cname varchar(20) unique,

severity varchar(10)

);

create table address(

aid serial primary key,

address varchar(50)

);

create table healthcare(

hcid serial primary key,

hcname varchar(10),

hcnum varchar(15) unique,

hcexp date

);

create table users(

usid serial primary key,

email varchar(10) unique,

password varchar(20)

);

create table personal\_info(

birth date,

email varchar(50),

marital varchar(10),

gender varchar(6),

phone char(9),

weight int,

height int,

blood varchar(3),

pid int unique references patient (pid),

aid int references address (aid),

hcid int references healthcare (hcid),

primary key(pid)

);

create table visits(

vdate date,

pid int references patient (pid),

did int references doctor (did),

primary key(vdate,pid,did)

);

create table diagnostic(

didate date,

pid int references patient(pid),

did int references doctor(did),

cid int references condition(cid),

primary key(didate,pid,cid,did)

);

create table credit\_card(

usid int references users(usid),

ufirst varchar(10),

ulast varchar(10),

cnumber char(16) unique primary key,

cvv char(3),

ccexp date,

address varchar(100),

city varchar(100),

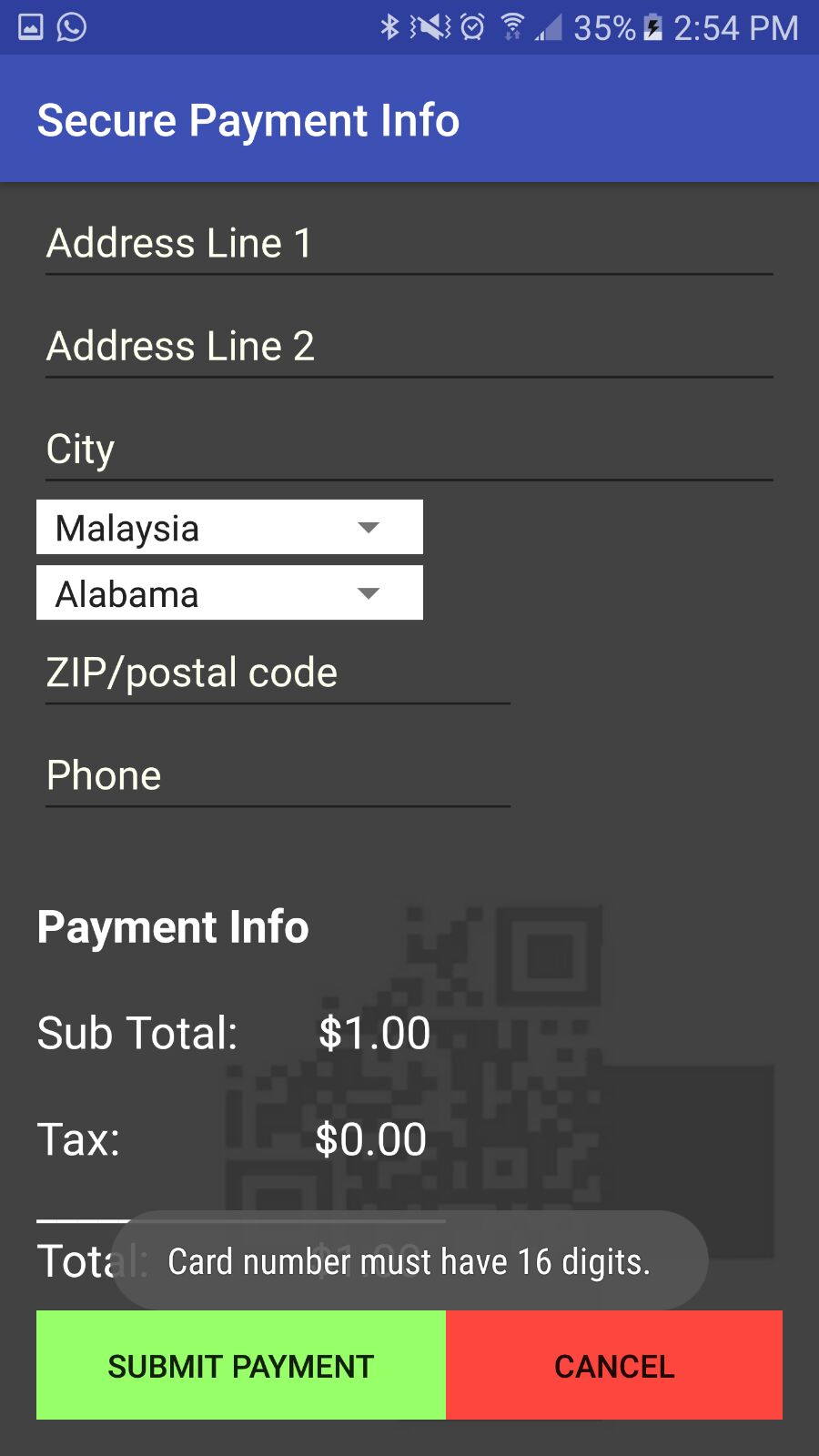
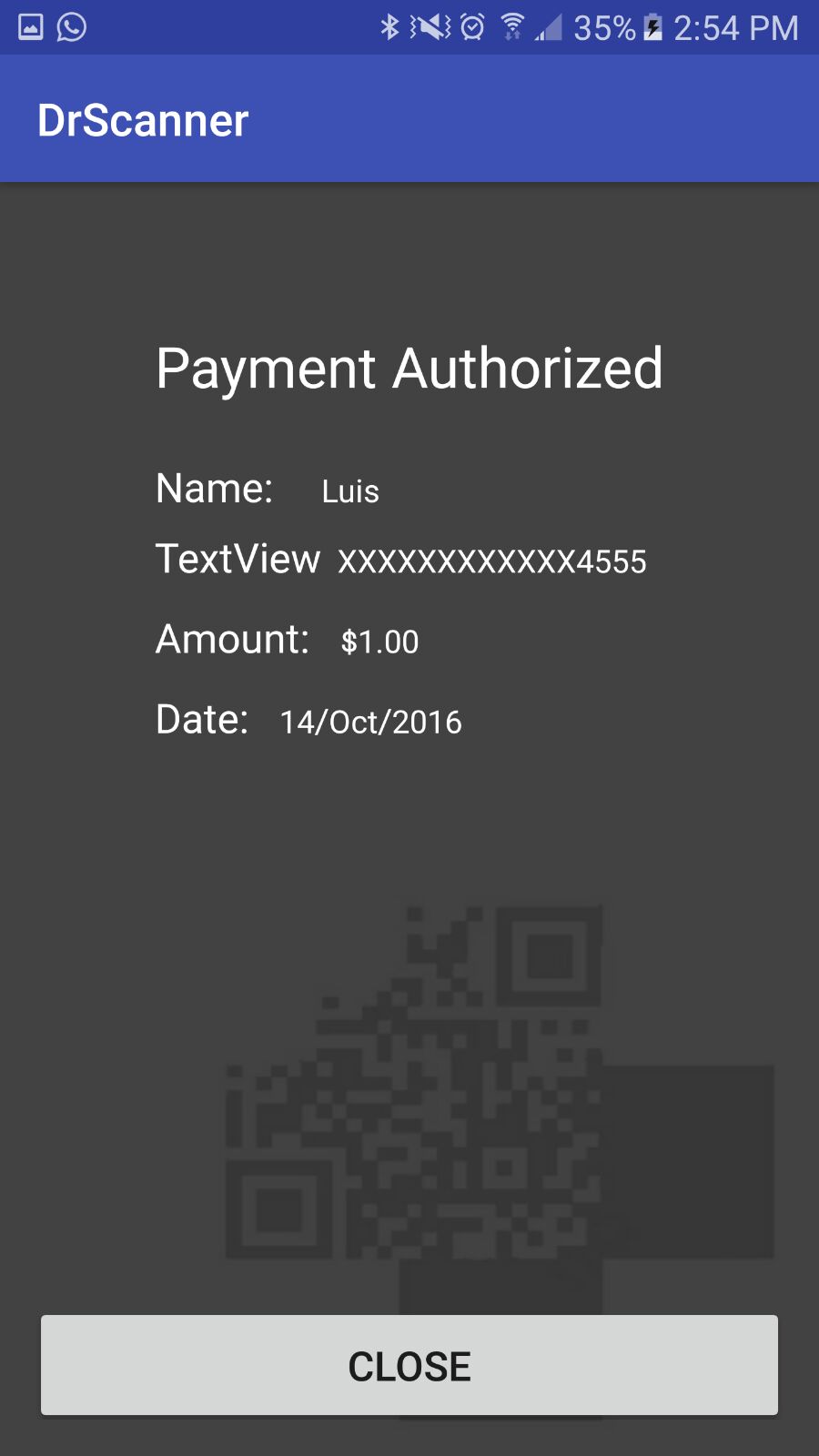
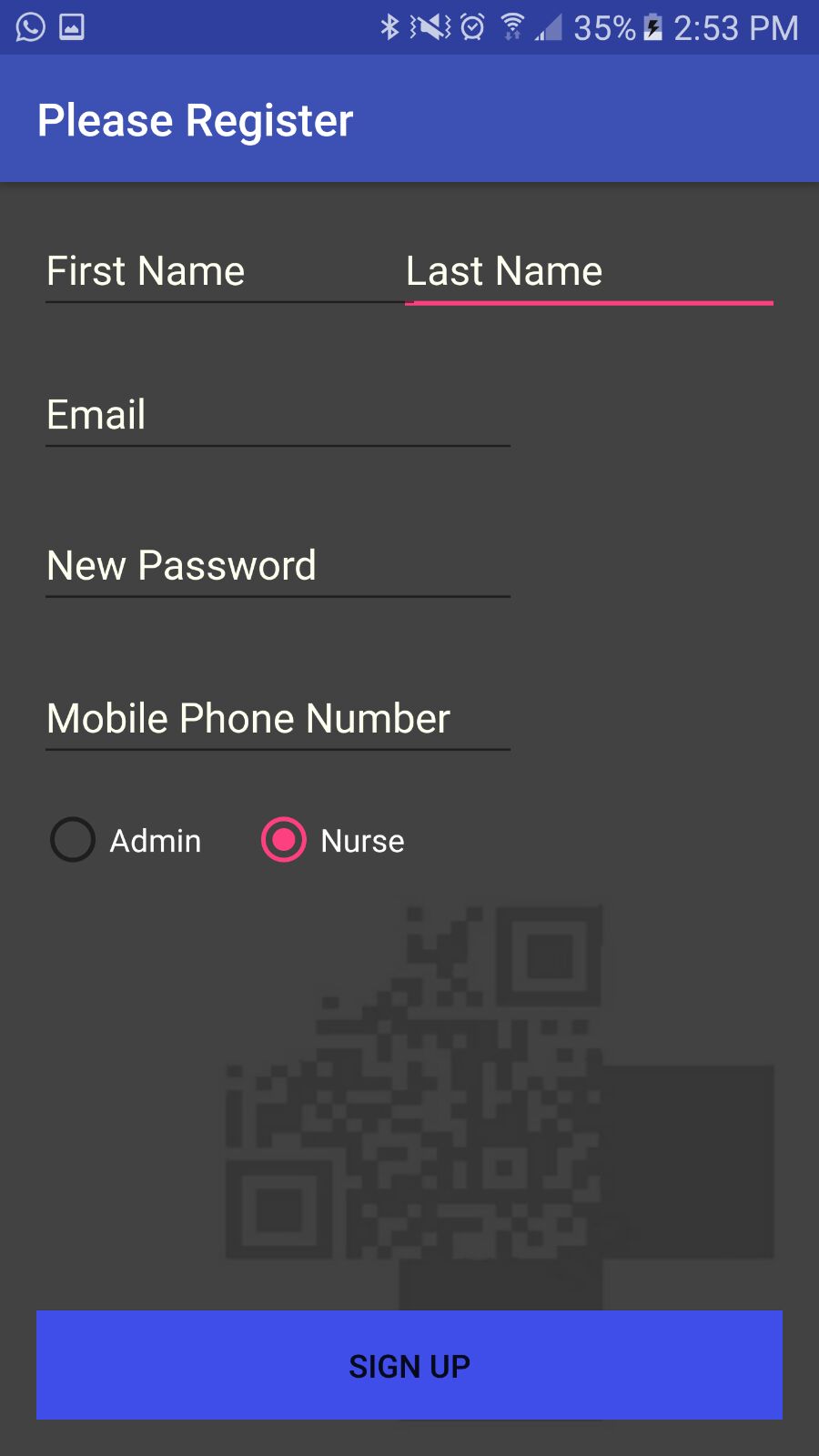
zipcode varchar(5)

);

**ER Diagram**

****

**Client Side Screenshots:**

****

