Baby In Car

Olga Mazo
Daniel Zilpa

Software Design Document

Date: (17/03/2020)

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Purpose

This software design document describes the architecture and system design of "Baby In Car".
This document is intended for developers and project managers.

1.2 Scope

A system that helps preventing children from being forgotten in the car.
The project consists of 2 main sections: the first is a mobile app and the second is a physical system being installed in the car.

1.3 Overview

The SDD document contains details about our products and system design information. See table of contents1.4 Reference Material

1.4 Reference Material

Arduino website: https://www.arduino.cc/

1.5 Definitions and Acronyms

Not relevant.

2. SYSTEM OVERVIEW

The app alerts the user when it detects a child forgetfulness situation in the car.
How does the app recognize this kind of situation?
The system installed in the car consists of 2 weight sensors: the first is installed under the child's safety seat and the second under the driver's seat.
When the system detects a child's presence in the car without a driver's presence for more than 3 minutes, the app alerts the user.

Another mechanism that exists in the system is a real-time test - when the user receives the alert, he can give a feedback whether it is a false or true alert, the system collects the data and displays statistics for the system's veracity.

3. SYSTEM ARCHITECTURE

3.1 Architectural Design

The app is installed on the user's mobile device
The sensor system is installed in the user vehicle
The app receives information from the sensors, processes it and according to the

algorithm decides when to send an alert to the user.

3.2 Decomposition Description

MainActivity: Home page of the app.
StatisticActivity: The statistic page.
AlertActivity: The alert page, the user gets to this page when he click on the alert.
Clicks object: count the true and the false alerts of all the system users.

3.3 Design Rationale

At the beginning the design was to use a magnetic sensor in the car for the seat belts to know if there is a person in the seat.
We did not choose this way because:
1. Not all cars have such a sensor.
2. Baby up to age 3 sits in a safety chair and not on the car seat directly (the seat itself is always strapped in the car's seat belt).

4. DATA DESIGN

4.1 Data Description

The data-base will hold a single object that is common to all users of the system. The object will contain 2 numeric fields, the first representing the amount of truth alerts and the second the amount of false alerts. This object will be updated according to the user feedback and through which the statistics will be displayed.

4.2 Data Dictionary

The data-base will hold a single object that is common to all users of the system. The object will contain 2 numeric fields, the first representing the amount of truth alerts and the second the amount of false alerts. This object will be updated according to the user feedback and through which the statistics will be displayed.

5. COMPONENT DESIGN

HomeActivity:

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {

    FirebaseAuth mAuth;
    DatabaseReference reff;
    private NotificationManagerCompat notificationManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
```

```
        findViewById(R.id.ma_button).setOnClickListener(this);

        mAuth = FirebaseAuth.getInstance();
        reff = FirebaseDatabase.getInstance().getReference().child("Clicks");

        notificationManager = NotificationManagerCompat.from(this);
    }

    public void onClick(View view) {

        switch (view.getId()){

            case R.id.ma_button:
                startActivity(new Intent(this, StatisticActivity.class));
                break;
            default:

        }
    }

    @Override
    public void onPointerCaptureChanged(boolean hasCapture) {

    }

    public void sendNotification() {
        Notification notification = new NotificationCompat.Builder(this, App.CHANNEL).

setSmallIcon(R.drawable.ic_android_black_24dp).setContentTitle("allert").setContentText
("kids in the car").

setPriority(NotificationCompat.PRIORITY_HIGH).setCategory(NotificationCompat.CATEGORY_M
ESSAGE).build();
        notificationManager.notify(1, notification);
    }
}



StatisticActivity:
public class StatisticActivity extends AppCompatActivity {

    AnyChartView anyChartView;
    int trueAlert = 80;
    int falseAlert = 20;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_statistic);
        anyChartView = findViewById(R.id.any_chart_view);
        setUpPieChart();
    }
```

```
    public void setUpPieChart() {
        Pie pie = AnyChart.pie();
        pie.background(String.valueOf(Color.BLACK));
        List<DataEntry> list = new ArrayList<>();
        list.add(new ValueDataEntry("True Alert", trueAlert));
        list.add(new ValueDataEntry("False Alert", falseAlert));
        pie.data(list);
        anyChartView.setChart(pie);
    }
}


AlertActivity:
public class AlertActivity extends AppCompatActivity {

    DatabaseReference databaseReference;
    Button trueButton, falseButton;
    int trueClicksI, falseClicksI;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_alert);

        trueButton=(Button)findViewById(R.id.a_true_button);
        falseButton=(Button)findViewById(R.id.a_false_button);

        trueButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                databaseReference =
FirebaseDatabase.getInstance().getReference().child("Clicks");
                databaseReference.addListenerForSingleValueEvent(new
ValueEventListener() {
                    @Override
                    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                       String trueClicksS =
dataSnapshot.child("trueClicks").getValue().toString().trim();
                        trueClicksI = Integer.parseInt(trueClicksS);
                        databaseReference.child("trueClicks").setValue(trueClicksI+1);
                    }
                    @Override
                    public void onCancelled(@NonNull DatabaseError databaseError) {

                    }
                });
            }
        });

        falseButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                databaseReference =
FirebaseDatabase.getInstance().getReference().child("Clicks");
                databaseReference.addListenerForSingleValueEvent(new
```

```
ValueEventListener() {
                    @Override
                    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                        String falseClicksS =
dataSnapshot.child("falseClicks").getValue().toString().trim();
                        falseClicksI = Integer.parseInt(falseClicksS);

databaseReference.child("falseClicks").setValue(falseClicksI+1);
                    }
                    @Override
                    public void onCancelled(@NonNull DatabaseError databaseError) {

                    }
                });
            }
        });
    }
}


Clicks:
public class Clicks {
    public int trueClicks = 0;
    public int falseClicks = 0;
}
```

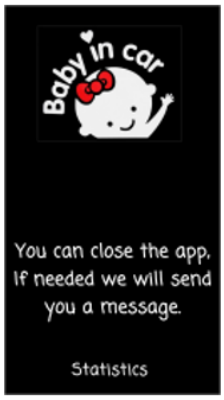
6. HUMAN INTERFACE DESIGN

6.1 Overview of User Interface

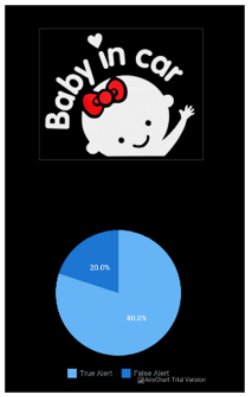
To use the system, the user will need to physically install the sensors in his car and install thr app in his mobile phone. The sensor system will then automatically send information to the app without the user having to take any proactive action, the app will process the data and alert if necessary.

6.2 Screen Images

Home Page:



Statistic Page:



Alert Page:

6.3 Screen Objects and Actions

According to the plan.

7. Requirements Matrix

The project logo at the top of the page	Home page
Displaying the Message in the Center of the Screen You can close the app.	Home page
Statistics Button - Moves to the Statistics View screen	Home page
The project logo at the top of the page	Statistics page
Data diagram for displaying the data - percentages of true alerts and percentages of false alerts	Statistics page
When the user clicks on the alert he receives he arrives on this screen	Alert Page
The project logo at the top of the page	Alert Page
Displaying the message in the center of the screen The system detected a baby in the car alone	Alert Page
True Alert button	Alert Page
False Alert Button	Alert Page

8. Appendices