<u>Baby In car</u>

Olga Mazo
Daniel Zilpa

Instructor : Elad Horev

Software Requirements Specification

Document

Version: (1.0) **Date:(21/01/2020)**

Table of Contents

1. Introduction

1.1 Purpose

The SRS document is intended to explain to the customer about the product and its purpose.

1.2 Scope

1. The product consists of 2 parts: a sensor system for car installation and app for mobile installation.
2. The app will receive data from the sensor system installed in the car.
3. The app will process the data and alert if necessary.

1.3 Definitions, Acronyms, and Abbreviations.

We don't have any Definitions, Acronyms, and Abbreviations.

1.4 References

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

1.5 Overview

The SRS document contains detailed information about the product and the system, both for the client and the programmer.
SECTION 2 is for the customer using the Web Application. It lists all the requirements the customer needs for the product, as well as the best use of the product.
SECTION 3 is for programmers. It lists all the functional and non-functional requirements of the product and explains the system setup.

2. The Overall Description

2.1 Product Perspective

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.

2.1.1 System Interfaces

Our system does not interfere with any external system.

2.1.2 Interfaces

The system interface with the customer:
1. Install the system.

2. Provide feedback on whether this is false or true when receiving the alert (optional).

2.1.3 Hardware Interfaces

The system hardware interface is the sensor system that must be installed in the car.
This system consists of 2 weight sensors, first in the child's safety seat and the second in the driver's seat.
There is support for Android operating system only.

2.1.4 Software Interfaces

The app interfaces to a firebase database.
The customer gives feedback on the alerts, the app updates the database and shows statistics.

2.1.5 Communications Interfaces

Among other things, the sensor system contains a component called Arduino. Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online.
We will use it to send data from the sensors to the app.

2.1.6 Memory Constraints

There are no memory constraints.

2.1.7 Operations

The system interface with the customer:
1. Install the system (sensors and app).
2. Provide feedback on whether this is false or true when receiving the alert (optional).

2.1.8 Site Adaptation Requirements

Using a car and an Android mobile device.

2.2 Product Functions

To use the system the client need to:
1. Physically install the system in the car (2 weight sensors).
2. Install the app on his mobile device.
Once installed, the customer does not need to take any proactive action, but once the system detects a need, it alerts the user of a child's forgetfulness in the car and if he wishes he can give feedback on the alert.

2.3 User Characteristics

The target audience are parents of small children who are concerned about their forgetfulness in the car.
The user does not need to have a technical background beyond installing a mobile app.

2.4 Constraints

1. There must be a lotus connection
2. The alert should reach the customer if a child has been identified in a vehicle without an adult over 3 minutes

2.5 Assumptions and Dependencies

1. The user knows how to physically install the system in the vehicle
2. The user knows how to install the mobile app and run the lotus services

2.6 Apportioning of Requirements.

Version 1.1 - Support for alerting if the system detects child forgetfulness condition in the vehicle
Version 1.2 - The system processes data and displays statistics about the veracity of the system
Version 1.3 - Support for IOS operating system

3. Specific Requirements

The physical system: the driver's seat and the child's seat will have a system consisting of a weight sensor, arduino, HX711 weight sensor amplifier and connection matrix.

Application requirements:
1. Home page:
1.1 The project logo at the top of the page
1.2 Displaying the Message in the Center of the Screen You can close the app.
1.3 Statistics Button - Moves to the Statistics View screen
2. Statistics page:
2.1 The project logo at the top of the page
2.2 Data diagram for displaying the data - percentages of true alerts and percentages of false alerts
3. Alert Page: When the user clicks on the alert he receives he arrives on this screen
3.1 The project logo at the top of the page
3.2 Displaying the message in the center of the screen The system detected a baby in the car alone.
3.3 True Alert button
3.4 False Alert Button

- No need to connect to the app.

- Clicking one of these buttons updates the statistics in the database.

- System Emergency This is a situation where the sensor detects weight in the child seat and no weight in the driver's seat.

- The system will alert the user when an emergency detector is detected for 3 minutes.

3.1 External Interfaces

[SKIP THIS PART]

3.2 Functions

The app can send information to the database.
The app can send an alert to the user.
The app will display the data in a user-friendly way.

3.3 Performance Requirements

[SKIP THIS PART]

3.4 Logical Database Requirements

[HERE KEEP SHORT BUT LIST THE DATA MODEL CLEARLY]

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.

3.5 Design Constraints

There are no design constraints.

3.5.1 Standards Compliance

[SKIP THIS PART]

3.6 Software System Attributes

[SKIP THIS PART – FILL ONLY SECTION 3.6.3 ON Security]

3.6.1 Reliability

3.6.2 Availability

3.6.3 Security

The user will only be able to enter a single feedback when an alert is received.

3.6.4 Maintainability

3.6.5 Portability

3.7 Organizing the Specific Requirements

3.7.1 System Mode

The system has 2 modes:
1. Child Attendance Identification - Continuously checks adult attendance in the car.
2. Does not detect child presence in car.

3.7.2 User Class

Not relevant we have only one user type.

3.7.3 Objects

The only object in the system is the truth of the system.

3.7.4 Feature

1. Alert in forgetful child mode
2. Presenting system truths by statistics

3.7.5 Stimulus

Weight detection in the child seat for 3 minutes without weight in the driver's seat causes the system to issue a user alert.

3. 7.6 Response

Weight detection in the child seat for 3 minutes without weight in the driver's seat causes the system to issue a user alert.

3.7.7 Functional Hierarchy

The system dont have Functional Hierarchy.

3.8 Additional Comments

4. Change Management Process

[SKIP THIS PART]

5. Document Approvals

[YOUR SUPERVISOR] Elad Horev

.

---------------------------------- [END OF RELEVANT PARTS]--------------------------

6. Supporting Information

[SKIP THIS PART]

The supporting information makes the SRS easier to use. It includes:

- *Table of Contents*
- *Index*
- *Appendices*

The Appendices are not always considered part of the actual requirements specification and are not always necessary. They may include:

(a) Sample I/O formats, descriptions of cost analysis studies, results of user surveys
(b) Supporting or background information that can help the readers of the SRS
(c) A description of the problems to be solved by the software
(d) Special packaging instructions for the code and the media to meet security, export, initial loading, or other requirements

When Appendices are included, the SRS should explicitly state whether or not the Appendices are to be considered part of the requirements.

Tables on the following pages provide alternate ways to structure section 3 on the specific requirements. You should pick the best one of these to organize section 3 requirements.

Outline for SRS Section 3
Organized by mode: Version 1

3. Specific Requirements

3.1 External interface requirements

3.1.1 User interfaces

3.1.2 Hardware interfaces

3.1.3 Software interfaces

3.1.4 Communications interfaces

3.2 Functional requirements

3.2.1 Mode 1

3.2.1.1 Functional requirement 1.1

.....

3.2.1.*n* Functional requirement 1.*n*

3.2.2 Mode 2

.....

3.2.*m* Mode *m*

3.2.*m*.1 Functional requirement *m*.1

.....

3.2.*m*.*n* Functional requirement *m*.*n*

3.3 Performance Requirements

3.4 Design Constraints

3.5 Software system attributes

3.6 Other requirements

Outline for SRS Section 3
Organized by mode: Version 2

3. Specific Requirements
    3.1 Functional Requirements
        3.1.1 Mode 1
            3.1.1.1 External interfaces
                3.1.1.1 User interfaces
                3.1.1.2 Hardware interfaces
                3.1.1.3 Software interfaces
                3.1.1.4 Communications interfaces
            3.1.1.2 Functional Requirement
                3.1.1.2.1 Functional requirement 1
                .....
                3.1.1.2.*n* Functional requirement *n*
            3.1.1.3 Performance
        3.1.2 Mode 2
            .....
        3.1.*m* Mode *m*
    3.2 Design constraints
    3.3 Software system attributes
    3.4 Other requirements

Outline for SRS Section 3
Organized by user class (i.e. different types of users ->System Adminstrators, Managers, Clerks, etc.)

3. Specific Requirements
    3.1 External interface requirements
        3.1.1 User interfaces
        3.1.2 Hardware interfaces
        3.1.3 Software interfaces
        3.1.4 Communications interfaces
    3.2 Functional requirements
        3.2.1 User class 1
            3.2.1.1 Functional requirement 1.1
            .....
            3.2.1.*n* Functional requirement 1.*n*
        3.2.2 User class 2
            .....

        3.2.*m* User class *m*
            3.2.*m*.1 Functional requirement *m*.1
            .....
            3.2.*m.n* Functional requirement *m.n*
    3.3 Performance Requirements
    3.4 Design Constraints
    3.5 Software system attributes
    3.6 Other requirements

Outline for SRS Section 3
Organized by object (Good if you did an object-oriented analysis as part of your requirements)

3 Specific Requirements
- 3.1 External interface requirements
  - 3.1.1 User interfaces
  - 3.1.2 Hardware interfaces
  - 3.1.3 Software interfaces
  - 3.1.4 Communications interfaces
- 3.2 Classes/Objects
  - 3.2.1 Class/Object 1
    - 3.2.1.1 Attributes (direct or inherited)
      - 3.2.1.1.1 Attribute 1
      - .....
      - 3.2.1.1.*n* Attribute *n*
    - 3.2.1.2 Functions (services, methods, direct or inherited)
      - 3.2.1.2.1 Functional requirement 1.1
      - .....
      - 3.2.1.2.*m* Functional requirement 1.*m*
    - 3.2.1.3 Messages (communications received or sent)
  - 3.2.2 Class/Object 2
  - .....
  - 3.2.*p* Class/Object *p*
- 3.3 Performance Requirements
- 3.4 Design Constraints
- 3.5 Software system attributes
- 3.6 Other requirements

Outline for SRS Section 3
Organized by feature (Good when there are clearly delimited feature sets.

- 3 Specific Requirements
  - 3.1 External interface requirements
    - 3.1.1 User interfaces
    - 3.1.2 Hardware interfaces
    - 3.1.3 Software interfaces
    - 3.1.4 Communications interfaces
  - 3.2 System features
    - 3.2.1 System Feature 1
      - 3.2.1.1 Introduction/Purpose of feature
      - 3.2.1.2 Stimulus/Response sequence
        - 3.2.1.3 Associated functional requirements
          - 3.2.1.3.1 Functional requirement 1
          - .....
          - 3.2.1.3.*n* Functional requirement *n*
    - 3.2.2 System Feature 2
      - .....
    - 3.2.*m* System Feature *m*
      - .....
  - 3.3 Performance Requirements
  - 3.4 Design Constraints
  - 3.5 Software system attributes
  - 3.6 Other requirements

Outline for SRS Section 3
Organized by stimulus (Good for event driven systems where the events form logical groupings)

- 3 Specific Requirements
  - 3.1 External interface requirements
    - 3.1.1 User interfaces
    - 3.1.2 Hardware interfaces
    - 3.1.3 Software interfaces
    - 3.1.4 Communications interfaces
  - 3.2 Functional requirements
    - 3.2.1 Stimulus 1
      - 3.2.1.1 Functional requirement 1.1
      - .....
      - 3.2.1.*n* Functional requirement 1.*n*
    - 3.2.2 Stimulus 2
    - .....
    - 3.2.*m* Stimulus *m*
      - 3.2.*m*.1 Functional requirement *m*.1
      - .....
      - 3.2.*m*.*n* Functional requirement *m*.*n*
  - 3.3 Performance Requirements
  - 3.4 Design Constraints
  - 3.5 Software system attributes
  - 3.6 Other requirements

Outline for SRS Section 3
Organized by response (Good for event driven systems where the responses form logical groupings)

3 Specific Requirements
 3.1 External interface requirements
  3.1.1 User interfaces
  3.1.2 Hardware interfaces
  3.1.3 Software interfaces
  3.1.4 Communications interfaces
 3.2 Functional requirements
  3.2.1 Response 1
   3.2.1.1 Functional requirement 1.1
   .....
   3.2.1.*n* Functional requirement 1.*n*
  3.2.2 Response 2
  .....
  3.2.*m* Response *m*
   3.2.*m*.1 Functional requirement *m*.1
   .....
   3.2.*m.n* Functional requirement *m.n*
 3.3 Performance Requirements
 3.4 Design Constraints
 3.5 Software system attributes
 3.6 Other requirements

Outline for SRS Section 3
Organized by functional hierarchy (Good if you have done structured analysis as part of your design.)

3 Specific Requirements
3.1 External interface requirements
3.1.1 User interfaces
3.1.2 Hardware interfaces
3.1.3 Software interfaces
3.1.4 Communications interfaces
3.2 Functional requirements
3.2.1 Information flows
3.2.1.1 Data flow diagram 1
3.2.1.1.1 Data entities
3.2.1.1.2 Pertinent processes
3.2.1.1.3 Topology
3.2.1.2 Data flow diagram 2
3.2.1.2.1 Data entities
3.2.1.2.2 Pertinent processes
3.2.1.2.3 Topology
.....
3.2.1.*n* Data flow diagram *n*
3.2.1.*n*.1 Data entities
3.2.1.*n*.2 Pertinent processes
3.2.1.*n*.3 Topology
3.2.2 Process descriptions
3.2.2.1 Process 1
3.2.2.1.1 Input data entities
3.2.2.1.2 Algorithm or formula of process
3.2.2.1.3 Affected data entities
3.2.2.2 Process 2
3.2.2.2.1 Input data entities
3.2.2.2.2 Algorithm or formula of process
3.2.2.2.3 Affected data entities
…..
3.2.2.*m* Process *m*
3.2.2.*m*.1 Input data entities
3.2.2.*m*.2 Algorithm or formula of process
3.2.2.*m*.3 Affected data entities
3.2.3 Data construct specifications
3.2.3.1 Construct 1
3.2.3.1.1 Record type
3.2.3.1.2 Constituent fields
3.2.3.2 Construct 2
3.2.3.2.1 Record type
3.2.3.2.2 Constituent fields
…..
3.2.3.*p* Construct *p*
3.2.3.*p*.1 Record type

3.2.3.*p*.2 Constituent fields
3.2.4 Data dictionary
3.2.4.1 Data element 1
3.2.4.1.1 Name
3.2.4.1.2 Representation
3.2.4.1.3 Units/Format
3.2.4.1.4 Precision/Accuracy
3.2.4.1.5 Range
3.2.4.2 Data element 2
3.2.4.2.1 Name
3.2.4.2.2 Representation
3.2.4.2.3 Units/Format
3.2.4.2.4 Precision/Accuracy
3.2.4.2.5 Range
…..
3.2.4.*q* Data element *q*
3.2.4.*q*.1 Name
3.2.4.*q*.2 Representation
3.2.4.*q*.3 Units/Format
3.2.4.*q*.4 Precision/Accuracy
3.2.4.*q*.5 Range
3.3 Performance Requirements
3.4 Design Constraints
3.5 Software system attributes
3.6 Other requirements

Outline for SRS Section 3
Showing multiple organizations (Can’t decide? Then glob it all together)

3 Specific Requirements

3.1 External interface requirements

3.1.1 User interfaces

3.1.2 Hardware interfaces

3.1.3 Software interfaces

3.1.4 Communications interfaces

3.2 Functional requirements

3.2.1 User class 1

3.2.1.1 Feature 1.1

3.2.1.1.1 Introduction/Purpose of feature

3.2.1.1.2 Stimulus/Response sequence

3.2.1.1.3 Associated functional requirements

3.2.1.2 Feature 1.2

3.2.1.2.1 Introduction/Purpose of feature

3.2.1.2.2 Stimulus/Response sequence

3.2.1.2.3 Associated functional requirements

…..

3.2.1.*m* Feature 1.*m*

3.2.1.*m*.1 Introduction/Purpose of feature

3.2.1.*m*.2 Stimulus/Response sequence

3.2.1.*m*.3 Associated functional requirements

3.2.2 User class 2

.....

3.2.*n* User class *n*

.....

3.3 Performance Requirements

3.4 Design Constraints

3.5 Software system attributes

3.6 Other requirements

Outline for SRS Section 3
Organized by Use Case (Good when following UML development)

3. Specific Requirements

3.1 External Actor Descriptions

3.1.1 Human Actors

3.1.2 Hardware Actors

3.1.3 Software System Actors

3.2 Use Case Descriptions

3.2.1 Use Case 1

3.2.2 Use Case 2

3.2.n Use Case n

3.3 Performance Requirements

3.4 Design Constraints

3.5 Software system attributes

3.6 Other requirements