In Situ Evaluation of Climate Control in Cars via a Mobile App

Master's Thesis
submitted to the
Media Computing Group
Prof. Dr. Jan Borchers
Computer Science Department
RWTH Aachen University

by
Sahar Delavar

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof. Dr. Thomas Rose

Registration date: 15.09.2017
Submission date: 12.03.2018

Eidesstattliche Versicherung

____________________________ ____________________________

Name, Vorname Matrikelnummer

Ich versichere hiermit an Eides Statt, dass ich die vorliegende Arbeit/Bachelorarbeit/ Masterarbeit* mit dem Titel

selbständig und ohne unzulässige fremde Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt. Für den Fall, dass die Arbeit zusätzlich auf einem Datenträger eingereicht wird, erkläre ich, dass die schriftliche und die elektronische Form vollständig übereinstimmen. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Ort, Datum Unterschrift

*Nichtzutreffendes bitte streichen

Belehrung:

§ 156 StGB: Falsche Versicherung an Eides Statt

Wer vor einer zur Abnahme einer Versicherung an Eides Statt zuständigen Behörde eine solche Versicherung falsch abgibt oder unter Berufung auf eine solche Versicherung falsch aussagt, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.

§ 161 StGB: Fahrlässiger Falscheid; fahrlässige falsche Versicherung an Eides Statt

(1) Wenn eine der in den §§ 154 bis 156 bezeichneten Handlungen aus Fahrlässigkeit begangen worden ist, so tritt Freiheitsstrafe bis zu einem Jahr oder Geldstrafe ein.

(2) Straflosigkeit tritt ein, wenn der Täter die falsche Angabe rechtzeitig berichtigt. Die Vorschriften des § 158 Abs. 2 und 3 gelten entsprechend.

Die vorstehende Belehrung habe ich zur Kenntnis genommen:

Ort, Datum Unterschrift

Sperrvermerk

Die vorgelegte Masterarbeit basiert auf internen, vertraulichen Daten und Informationen des Unternehmens Ford-Werke GmbH.

Die Abschlussarbeit darf nur dem Erst- und Zweitgutachter sowie befugten Mitgliedern der Prüfungsorgane zugänglich gemacht werden. Eine Veröffentlichung und Vervielfältigung der Abschlussarbeit ist – auch in Auszügen – nicht gestattet.

Eine Einsichtnahme der Arbeit durch Unbefugte bedarf einer ausdrücklichen Genehmigung durch den Verfasser und das Unternehmen.

Diese Geheimhaltungspflicht ist auf zehn Jahre nach der Veröffentlichung der Arbeit - bis 12.03.2028 - begrenzt.

Aachen, March2018
Sahar Delavar

Contents

List of Figures

List of Tables

Abstract

Nowadays, digitalization becomes an inseparable part of each business but it can be seen that traditional ways are still there, especially in lower hierarchies.

However, it should be considered that if the jobs in lower hierarchies digitized first, higher ones could act more accurate and deliver better results.

Evaluators in the Ford Motor Company use pen and paper for collecting objective and subjective comfort assessment data of the climate control system.

In order to solve the problems that data acquisition department and evaluators may face, we introduce a mobile application called "CCSE" which tries to interactively provide a fast and accurate procedure for collection of data.

In this thesis, we explain the development of CCSE from the early stage through the working system.

Überblick

Heutzutage ist die Digitalisierung ein untrennbarer Teil jedes Unternehmens, aber traditionelle Arbeitsplätze und Unternehmenswege sind immer noch viele vorhanden, insbesondere in den unteren Hierarchiebenen.

Es sollte bedacht werden, dass wenn diese Jobs zuerst digitalisiert werden, die höheren Ebenen dadurch erheblich entlastet werden und bessere Ergebnisse liefern können.

Gutachter der Ford Motor Company, verwenden noch Stift und Papier, um objektive und subjektive Daten zum Komfort der Klimaanalagen in ihren Modellen zu erheben.

Um solche Aufgaben zu erleichtern und um Probleme, auf die Gutachter oder Forschungsabteilungen dabei stoßen könnten, schneller und besser zu lösen, möchten wir die mobile Applikation “CCSE” vorstellen. Die Applikation ermöglicht es, solche Daten interaktiv, schnell und präzise zu erfassen und zu verarbeiten.

In dieser Arbeit stellen wir den Entwicklungsprozess von CCSE, vom Entwurf bis zur funktionierenden Applikation, vor.

Acknowledgements

Foremost, I would like to thank Krishna Subramanian, M.Sc., my supervisor, for his guidance and insightful comments throughout this thesis.

I would like to thank Sandra Krolewski and Bernd Dienhart, who had given me the opportunity to work with their amazing team at Ford. This work would not have been possible without their support. It was an invaluable experience.

I would like to thank Prof. Dr. Jan Borchers, my thesis advisor, for giving me an opportunity to work on my thesis under the supervision of Media Computing Group.

I would like also to thank Prof. Dr. Thomas Rose, my second examiner, for his time and support.

Finally, I must express my very profound gratitude to my family for their unfailing kindness throughout my studies.

Thank you!

Sahar Delavar

Conventions

Throughout this thesis, we use the following conventions.

Definitions of technical terms or short excursus are set off in colored boxes.

EXCURSUS:
Excursus are detailed discussions of a particular point in a book, usually in an appendix, or digressions in a written text.

Definition:
Excursus

Android UI components and classes are emphasized e.g. the word *seekBar*.

The whole thesis is written in American English.

The text is written in first person plural because of aesthetic reasons.

Irrespective of the real users' gender we will use "he" as a reference to a single person.

Chapter 1

Introduction

"Setting goals is the first step in turning the invisible into the visible."

- Tony Robbins

1.1 A Brief History

Passenger's comfort during the journey is one of the main concerns of the car manufacturers. Regardless of how the weather is, the driver and the passengers need to feel fulfilled by the environment of the vehicle.

Starting from the Packard Motor Car Company, the first manufacturer that offered an air conditioning unit in its cars in 1939 (A/C Pro [2014]), till now, various factors have been considered to provide an ideal temperature in the vehicles.

First, the cooling system was not located in the dash but it was in the trunk, so the driver had to install or remove the drive belt from the A/C compressor manually in order to turn the system on or off (A/C Pro [2014]).

Later, it moved to the trunk in front of the cars which were manufactured by Pontiac and Nash in 1954 for the first time. It was a combination of the air conditioner and heater all into one in-dash system (A/C Pro [2014]).

The foundation of today's modern climate control systems

After that, in 1964 the Comfort Control was introduced by Cadillac. The functionality of this system was to set the desired temperature, and then it could adjust the A/C or heater output based on that temperature automatically. This was the foundation of today's modern climate control systems (A/C Pro [2014]).

1.2 Climate Control System

The functionality of climate control system

The functionality of this complex system has to be set via its front end interface whether it has different knobs and switches or is an electronic panel. Then it starts to interact with the back end of the system which consists of one or more blower motors and a couple of actuators that both control the fresh air circulation, air-flow and temperature, and also a refrigeration unit that acts as a cooler on the air and many ducts. In a group, they determine how air should be transferred into the cabin (PantherBB [2005]).

This ability of the system can dramatically increase the enjoyment and the contentment of journey for passengers. Now, the temperature of the vehicle either could be modified via the automatic system of HVAC or by using knobs and switches to control the airflow. In fact, after the introduction of HVAC system, car industrial companies started to capitalize on this system to improve customer satisfaction by preventing the struggle to regulate the temperature (PantherBB [2005]).

Definition: *HVAC*

HVAC:
HVAC (Heating, Ventilation and Air Conditioning) is the control unit that regulates the operation of a heating and/or air conditioning system using sensors to compare the actual state (e.g. temperature) with a target state. Then the control system draws a conclusion what action has to be taken (Wikipedia [2017]).

1.2.1 Climate Control System Evaluation

It is very crucial that we know such a system works properly and whether we have reached a pleasant environment inside the vehicle or not.

This is where climate control system evaluators' role highlighted. Evaluators test the functionality of this system in different scenarios, situations and weather conditions, and provide different types of assessment.

There are four major modes that a vehicle should be tested in various weather conditions:

- Heater and Mild Ambient Warm-up
- Steady State Drive
- A/C and Mild Ambient Pull-down
- Single Zone Control Curves

Each test-mode has its own specifications, therefore, the evaluators should fill out dissimilar forms of assessment. During each test, they have to write the information, organize them and keep the track of time to collect the assessment data about the performance of the climate control system in specific periods.

Assessment with pen and paper

Over the years of doing this work in a traditional way and dealing with all issues of physical documents, there have been significant needs to upgrade the procedures. To fulfill these needs, we introduce CCSE app as a means of digitalization.

Evaluation tests need to be upgraded

Digitalization:
Digitalization is the use of digital technologies to change a business model and provide new revenue and value-producing opportunities; it is the process of moving to a digital business (Gartner, Inc. [2017]).

Definition: *Digitalization*

1.3 Motivation

Potential problems of data collection in the traditional way

Among all the problems of the system evaluation in the traditional way, main troublesome ones are time tracking, organizing and searching in evaluation sheets.

Evaluation in specific periods of time may lead to the different problems such as crossing the time limit and lack of awareness.

Collecting assessment data manually may lead to unreadability and fail to enter values which both are because of hastiness and lack of time.

As there could be many tests during each evaluation process, storing all these sheets according to different tests and vehicles will lead to a big stack of papers which takes time to organize and needs a lot of work to search among them to find a specific one.

1.4 Contribution

This thesis develops an Android Application, called CCSE, to eliminate problems that evaluators may face.

From here, we will use the term “app” to refer to the “CCSE Application”.

Approaches of the app to solve the problems

This app tries to make the evaluator interact with it simply and figure out what is expected to do easily.

The well-structured activities of the app help to alleviate the pressure of time restriction and time crossing by keeping the evaluator aware through the processes and warning him when the time is over.

The output is a PDF file in a similar format to the physical documents.

The hierarchical order of stored files will help the evaluators to search and navigate among them significantly short and easy.

To summarize, the main purpose of this thesis is digitalization of the test procedures. So far, the tests have been done with pen and paper; thus, lots of resources needed to be used. This thesis is about the transformation of this physical documents into the digital ones.

In order to fulfill this purpose, these two stages have been applied:

- **Idea Creation and System Study**

  Investigating the possible issues and their solutions, the platforms and devices, the identification of the target users and test requirements.

- **Iterative Design**

  Starting from applying the ideas on flowcharts and prototypes then gathering feedback about them and going through the cycle of development, testing and debugging of the app to achieve the working system.

1.5 Outline

The thesis is organized as follows:

- Chapter 1, in this chapter, a short history of the climate control system and its definition, motivation, and contribution of this work were presented.
- Chapter 2, "System Study" introduces different evaluation tests, their definition and specifications.
- Chapter 3, "Design and Implementation" describes the iterative design of the app, evaluation of the design in each iteration, a detailed walk-through of the activities and layouts, and the process of output generation.
- Chapter 4, "Evaluation" takes a detailed look at functional and non-functional requirements of the system.
- Chapter 5, "Summary and future works" describes an overview of this work and recommendations for further developments.

Chapter 2

System Study

"It's a victory for the system that the jury is carefully studying the evidence and weighing the decision."

- Stephen Jones

2.1 Evaluation Tests

With respect to different functionalities of the climate control system, the evaluators test the performance of the system to recognize the highlights and the challenges. Each test has its own specifications and measurements.

2.1.1 Heater and Mild Ambient Warm-up

This test is performed on a vehicle when the engine is cold and the evaluator ignites the engine and inspects how quickly the engine becomes warm when started.

The duration of the test is for at least 30 minutes in which specific information should be recorded every 2 minutes.

From here, we will use the term "Warm-up" to refer to this test.

2.1.2 Steady State Drive

This test is to monitor the performance of the system and the comfort of the drive when the engine is warmed and the vehicle is on its journey toward the destination. The engine speed and load are held at a finite interval.

The duration of the test is for 30 or 45 minutes in which specific information should be recorded every 5 minutes.

From here, we will use the term "Steady-state" to refer to this test.

2.1.3 A/C and Mild Ambient Pull-down

The aim of this test is to check how does the system works when the engine is resting and the vehicle is not moving. This is the stop and go drive cycle and the goal is to have the minimum time to comfort, controlled undershoot and quick ramp down.

The duration of the test is exactly 30 minutes in which specific information should be recorded every 2 minutes.

From here, we will use the term "Pull-down" to refer to this test.

2.1.4 Single Zone Control Curves

This test is for checking the system behavior in different setpoints which in Celsius degree are 22, 18, 20, 22, 24, 26, 22 or the corresponding setpoints in Fahrenheit degree are 72, 64, 68, 72, 75, 79, 72.

Duration of the test is 85 minutes; the system should be stabilized for 15 minutes at first then it should be stabilized for 10 minutes at each setpoint.

From here, we will use the term "Control-curves" to refer to this test.

2.2 Information Classification

For each test, two categories of information should be recorded. General information which should be recorded only once during each test and specific information which should be recorded in a fixed time slice, repeatedly during the test (Fig. 2.1).

2.2.1 General Information

The required general information for all tests is the same as follows:

- Filename
- Program type
- Engine type
- Vehicle name
- Driver and passenger names
- Start and end locations
- The time and the date
- Weather conditions
  - Hot, Mild, Cold
  - Sunny, Partly cloudy, Cloudy, Night
  - Dry, Rain, Snow
- Vehicle Engineering Rating (VER)

2.2.2 Specific Information

The required specific information is almost the same for all tests and consists of subjective and objective data.

- **Objective data**

  In this type, user will record data from the sensors and the installed devices as follows:

  - Setpoint temperature, in Control-curves
  - Start time, all tests except Control-curves
  - Ambiance
  - In-car temperature
  - Breath values
  - Panel and floor discharge values
  - Sun-load
  - Blower volts
  - HVAC mode
    * Floor, Panel, Defrost, Auto
  - Recirc mode
    * Recirculated air, Fresh air, Auto
  - Air Condition (A/C)
    * On, Off
  - Engine Cooling Temperature (ECT), in Warm-up
  - Engine Round Per Minute (RPM), in Warm-up

- **Subjective data**

  In this type, the user will record his expectations and feelings as follows:

  - Thermal comfort scale
  - Abnormal cold or hot spots
  - Comments

2.3 Evaluators and Managers

Evaluators record general and specific information and check the system objectively and subjectively.

Managers consider their overall feeling and perception of the system for each test except Control-curves to see whether they are desirable. They will record general information and subjective data only once in a test and not repeatedly.

These two groups are the target users of the system which will be described more in detail in the next chapter.

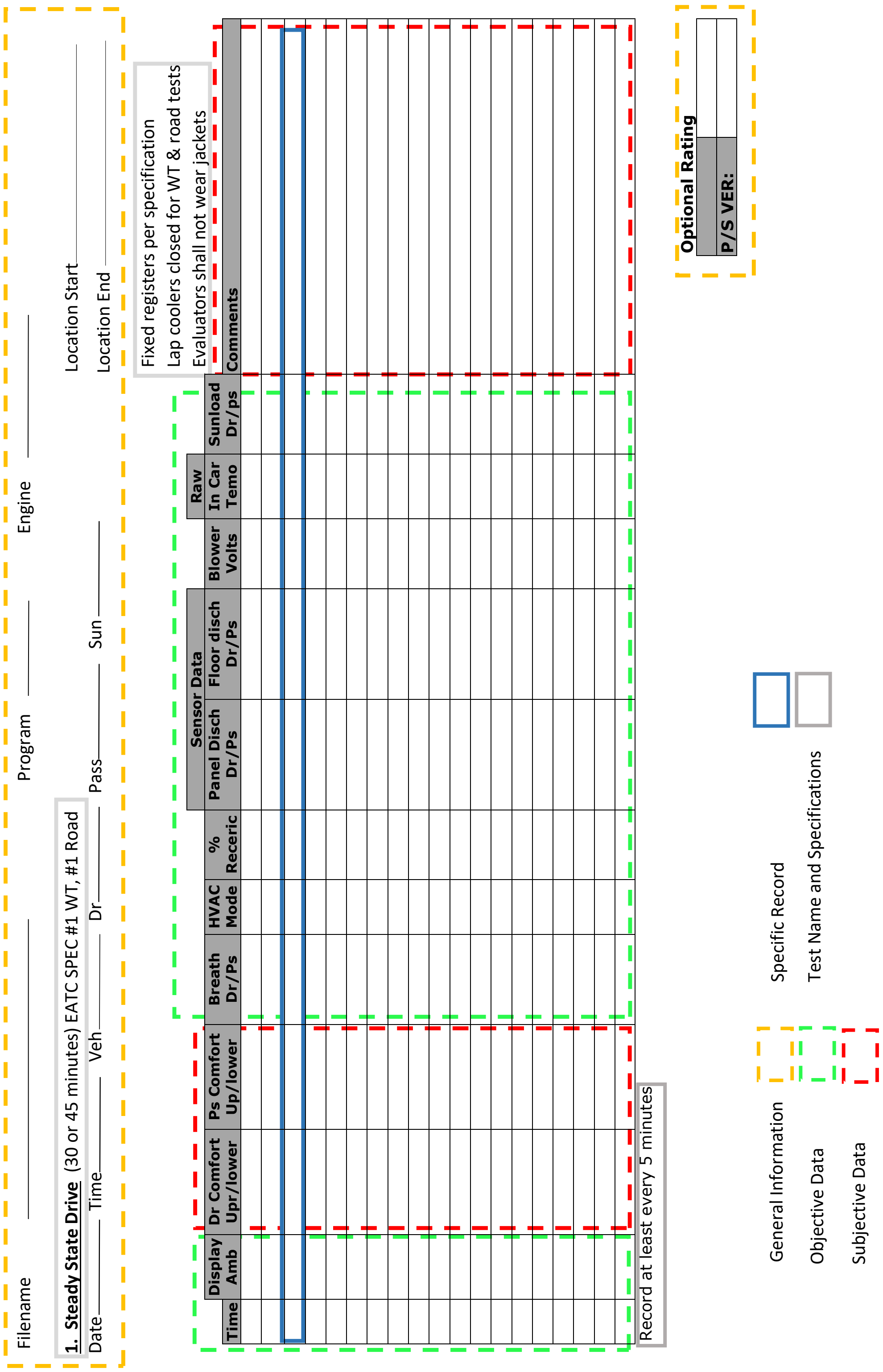


Figure 2.1: Sample test sheet with highlighted details

Chapter 3

Design and Implementation

"In theory, there is no difference between theory and practice. In practice, there is."

- Donald A. Norman

3.1 Brainstorming

To have a clear vision of the existing system, find appropriate scenarios for the situations, and know the users' expectations of the project and their requirements to finish a test successfully, various brainstorming sessions were conducted. Initial desires and demands were as follows:

- An Android Application to collect all the data during the test
- A PDF file as output
- Ease of access to the desired test file

Next, we attended a real test to have a better understanding of data collection procedure as well as gaining knowledge about the environment and potential problems of the tests.

After the first observation of a real test, the following questions still needed to be answered:

1. What is the meaning of each test?
2. What is the meaning of technical words such as ECT, HVAC, RPM?
3. Why it has to be filled also by the managers?
4. Is it acceptable to skip a test or a part of it?
5. Do all the entries have to be filled?
6. What could cause a test to fail regarding the input values?
7. What if the users could not finish the test on its time?
8. Are all the evaluators expert users? Are they familiar with all of the regulations and test procedures?
9. What is missed on the forms and can be added to them?

On further inspection, we found out two basic rules of the tests with respect to the validity of the records:

1. Neither tests nor their entries are allowed to be skipped.
2. The test fails if is not done in its specific period, as the whole point of the time slots is to see the behavior of the system in particular durations and periods.

To answer the other questions, we did some interviews with users and the explanations are discussed in the different sections of the thesis impliedly.

Gender: ○ Male ○ Female

Age: ______

Experience of working with smartphones:
○ Beginner ○ Intermediate ○ Advanced ○ Expert

Information about the different test processes:
○ Familiar with the tests and never forget their specifications.
○ Familiar with the tests but sometimes forget their specifications.
○ Familiar with the tests but always check their specifications.
○ Unfamiliar with the tests.

Duration of experience as a climate control system evaluator: ______

Figure 3.1: Experiment questionnaire to classify users

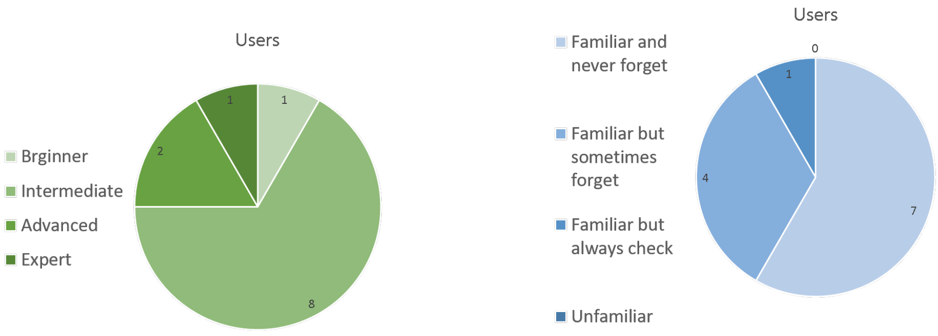
3.2 User Classification

The main concerns about users were their experience of doing the assessments, and their familiarity with smartphones. For this purpose, we provided a simple questionnaire (Fig. 3.1).

Considering the result of the questionnaire (Fig. 3.2), users are experienced and know the tasks well and almost all of them are intermediate users of smartphones. Thus users could be classified as average experienced users.

3.3 Iterative Design

To develop our app iteratively, we used DIA cycle. We designed a prototype and implemented it, then we evaluated our design by observation and gathering the feedback, and the next iteration started after applying the changes.



(a) Experience of working with smartphones **(b)** Information about the different test processes

Figure 3.2: Result of experiment questionnaire for user classification

Definition: *DIA cycle*

DIA CYCLE:
A design iteration is defined as a full DIA cycle (design, implement, analyze). Every time a functional prototype created, the problems analyzed, and changes made to the design (Ballagas [2007]).

3.3.1 Low Fidelity Prototype

We decided to use flipbook as the type of paper prototype, so we sketched different screens on separate pages (Fig. 3.3).

To guide the users interact with the prototype comfortably, the same order of collecting data as in a test sheet considered in the design.

Model Extraction Design Evaluation

To do the evaluation of the design with the users, we had a model extraction session.

We showed the prototype to the users and asked them to interpret what they see, think and desire. Users liked how

Figure 3.3: Low Fidelity Prototype - First Iteration

they can simply go into different screens to do the evaluation test and through their explanations, asked for some changes.

According to the first iteration, we got the following suggestions and we made the changes:

1. To add a sketch of a dummy:

   They suggested to add an illustration of a dummy, so they could specify the abnormal cold or hot spots of their body.

2. To add the specifications of the test:

   They recommended to remind them about the specifications of each test before the start of it, because of the rigidity of these rules for the test and to avoid mistakes.

3. To replace *Spinners* with *editTexts*:

   (a) Since there are numerous vehicles, and the number may increase with new products, they preferred to type the vehicle name and the program.

(b) As the locations are not foreknew and could differ every time or even during the test, they again preferred *editTexts* to enter the data. However, we discussed using GPS but they refused as they did not have access to the Internet during the test.

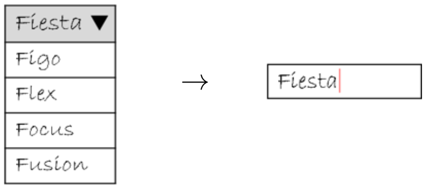


Figure 3.4: Replacement of *Spinners* with *editTexts*

Considering how to achieve more feedback and discover potential problems, we decided to continue with a new iteration of low fidelity prototype.

Second Iteration

Based on the feedback and observations of the first iteration, we redesigned the prototype and instead of papers, we simulated the transitions with clickable areas on the images to make the interactions and navigation through the paper prototype possible.

For this purpose, we used an application which is called Marvel[1] to create an interactive prototype. To design such a prototype, we scanned the last modified version of the sketches, set links between images and defined clickable areas on them.

Then, the design evaluated with the users by using various evaluation techniques.

[1]https://marvelapp.com/

- **Silent Observation**

  This technique was to understand if the users can interact with the prototype easily and if the design is apparent and understandable.

  Meanwhile the observation, we identified some misleading links between screens.

- **Think Aloud**

  We asked the users to work with the prototype and explain their actions.

  Through this observation, we recognized some gulfs of evaluation and execution.

- **Constructive Interaction**

  In this evaluation, two users started from the first screen and discussed their impressions of the design, what they wanted to add, modify or remove.

According to the mentioned evaluations and the observations, we got the following feedback and suggestions:

1. To add a component to set the HVAC system on auto mode, in manager section.

2. To add a component for T-set value.

3. To remove filename *editText* in manager section and generate it automatically in the output file.

4. To consider signed float for T-set value and signed integer for ambiance.

5. To remove the test-mode *buttons* from the last screen as a test could occur more than once, even in a sequence depends on the time and location.

6. To consider both Celsius and Fahrenheit degrees.

7. To add a timer that keeps the users aware of the remained duration of the record.

Definition: *Record*

RECORD:
We define a record as a set of specific information that should be collected in a fixed time slice.

To observe that the interactions in the low fidelity prototype meet the users mental model, we continued our design process with a new iteration at this level.

Third Iteration

In this approach, we redesigned the prototype based on the feedback and desires (Fig. 3.5). At the evaluation session of this iteration, users were satisfied with the overall performance of the prototype and there were no additional requests.

At this point we suggested some improvements that we had realized in our heuristic evaluation as follows:

1. To remove the date and time from the screen and have them only in the output file:

   Users refused this suggestion as they thought, it is better if they know such a feature is included.

2. To shift the general information to the beginning:

   Users agreed since different tests almost have the same general information e.g. vehicle, driver, passenger names.

So far, the structure of the app had been designed, almost all of the user expectations and requirements had been discovered and in three iterations were implemented.

As there were no more issues or suggestions in the evaluation session of the last iteration, we decided to continue with a new iteration as a medium fidelity prototype.

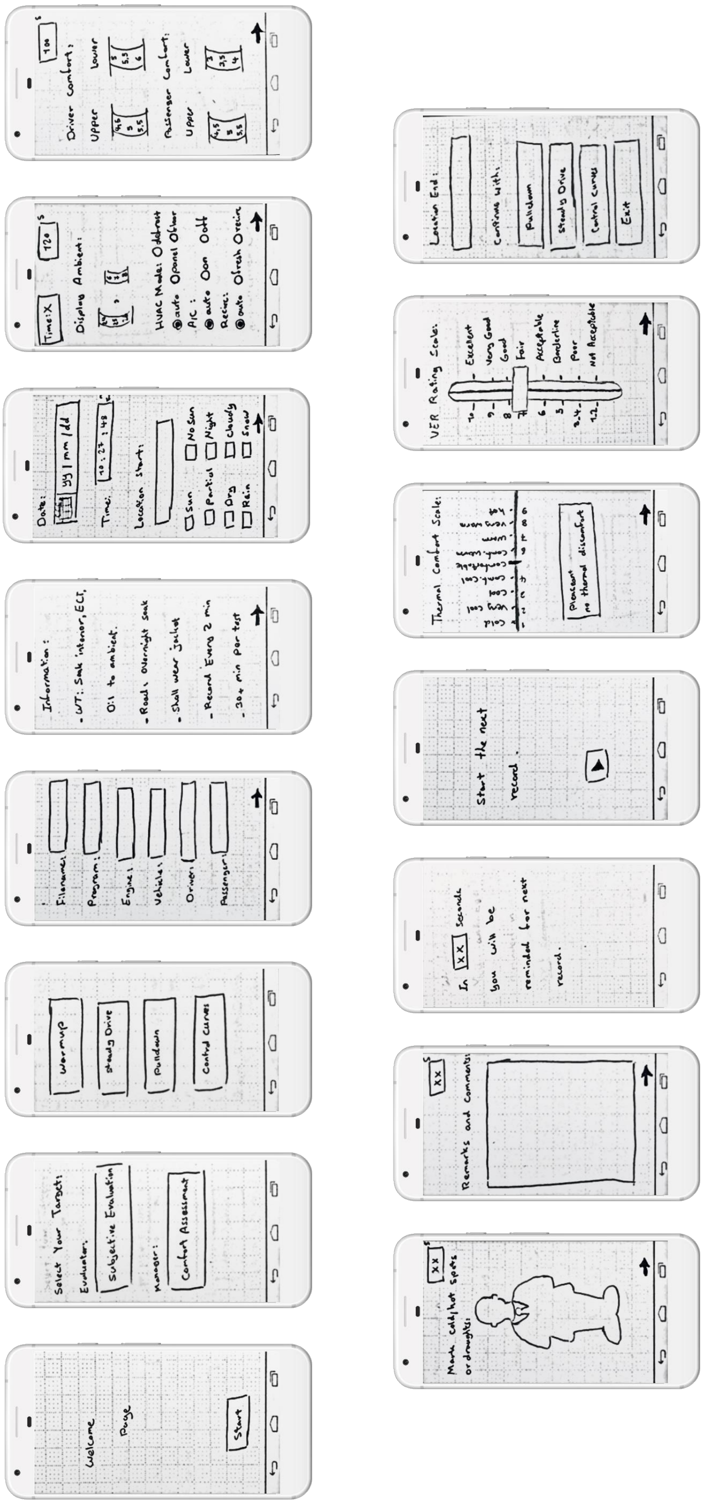


Figure 3.5: Low Fidelity Prototype - Second Iteration

3.3.2 Medium Fidelity Prototype

In this iteration, the main concern was to have a prototype with more details that user could become more involved to interact with it.

For this purpose, we decided to implement a basic application, meaning that the initial implementation of the User Interface (UI) included layouts and transitions between them.

We declared each screen of the last version of low fidelity prototype in a layout and their elements as placeholders in Android Studio[2] IDE in XML.

Definition: *Layout*

> **LAYOUT:**
> A layout defines the structure for a user interface in the app (Android [2009]).

The following reasons led us to choose the Android Studio IDE as the UI builder, and a basic application as the software prototype.

- From the beginning of the project, the considered final product was an Android application, and we were almost assured that the implementation of the app would be in Android Studio IDE.
- We assumed that with this IDE we could combine the vertical and horizontal prototypes easily.
- The implementation of a basic app as prototype could incrementally become the actual app.

Definition: *Horizontal Prototype*

> **HORIZONTAL PROTOTYPE:**
> Horizontal prototypes display a broad view of the UI but without concentrating on the functionality.

[2] https://developer.android.com/studio/index.html

VERTICAL PROTOTYPE:
Vertical prototypes focus on implementing a few subsystems and their functions comprehensively and do not attempt to show the entire UI.

Definition:
Vertical Prototype

The App Layouts

We designed the prototype in a fixed storyboard of screens, which for this iteration, the UI for the Warm-up test of the "Subjective Evaluation" section was created and all its functions were implemented (Fig. 3.6).

This section describes each layout of the prototype and for the ease of reference, each of them has a name.

1. The "Main" layout (Fig. 3.6.a) includes a *button* to start the app.

2. In the "Target" layout (Fig. 3.6.b) user will choose who is going to do the test. It could be an evaluator, who will do the "Subjective Evaluation" or a manager for the "Comfort Assessment" test.

   Based on the selection, the user navigates to a different layout.

3. Navigation through the "Subjective Evaluation" *button*:

   The "General Information" layout (Fig. 3.6.c) consists of six *editTexts* to enter the filename, program, engine, vehicle, driver and passenger names.

   In this layout, although all data are similar, we used the Gestalt law of closure to separate the relevant details e.g. vehicle details, evaluator details.

4. In the "Test-mode" layout (Fig. 3.6.d), user will choose the one of the test-modes as Warm-up, Steady-state, Pull-down or Control-curves. After the selection, next associated layout would be shown.

Figure 3.6: Medium Fidelity Prototype - First Iteration

5. Navigation through the Warm-up test-mode *button*:

   The "Weather Condition" layout (Fig. 3.6.e) consists of an *editText* to enter the start location of the test, a box that contains the time and the date, *radioButtons* to select the sun positions as sun, no sun and night and *checkBoxes* in two groups to select the general weather conditions which are weather type as hot, mild and cold and humidity as dry, rain and snow.

   As data differed in type, we used different design rules. Gestalt law of closure to separate the sections and the law of proximity to group the *checkBoxes*.

6. The "Specifications" layout (Fig. 3.6.f) is to remind the specifications and rules of the test.

   Since the graphical design of the UI had scheduled to be done in the high fidelity prototype, this layout had no content at this level.

   As the next layout is the first layout of the actual test, different shape of the next *button* in this layout can be regarded as a signifier for the start of the actual test.

   Moreover, by entering to the actual test, the timer on the top of each layout hereafter is another signifier for the users.

ACTUAL TEST:
We define the main part of the test as the actual test which consists of a set of records that should be collected in a defined duration.

Definition:
Actual Test

7. The first layout of the actual test is called "HVAC" (Fig. 3.6.g) and besides the timer, it consists of two *radioButtons* to indicate the temperature scales as Celsius or Fahrenheit, two *numberPickers* to specify the ambiance and in-car temperature values and *radioButtons* in three groups to indicate the HVAC mode which are airflow as floor, panel, defrost or auto, and air recirculation as recirculated, fresh or auto, and A/C mode as on, off or auto.

The timer is shown in two approaches:

- Numerical remaining time in minutes and seconds.
- A *progressBar* which as a Gestalt law of continuity helps the user to have a visual overview of the remaining time in all layouts of the actual test.

8. The "Comfort" layout (Fig. 3.6.h) consists of four *numberPickers* which have the values from 0 to 10 in which driver and passenger indicate their comfort for upper body and lower body.

   User can also choose the middle scale of this numbers to be more precise.

9. The "Dummy" layout (Fig. 3.6.i) consists of two illustrations of a dummy and there are *popupMenus* on twelve parts of each dummy which by tapping on them, the menu will be shown with the name of the spot as the title, and warm and cold as the items by which diver and passenger indicate their hot or cold body parts.

   The body parts are face, chest, lower and upper arms, lower and upper legs and feet.

10. The "Comment" layout (Fig. 3.6.j) includes an *editText* to enter comments for the current record.

11. The last layout of the actual test is called "Countdown" (Fig. 3.6.k) which consists of a *textView* to inform the user about the remained time of the record and when the time is over an alert message would be appeared (Fig. 3.6.l) which contains a *button* to start a new record.

    These layouts were designed to complete one record of the actual test, as the prototype was semi-functional the test will be over after the second record in this iteration, although this process have to be continued for the definite numbers of the records.

12. When the second record of the actual test is done, the "Thermal" (Fig. 3.6.m) layout will be shown which

consists of two *seekBars* to indicate the thermal comfort scale for the driver and the passenger.

The integer numbers point to different thermal comfort scales as follows:

(1) Bitterly cold, extreme discomfort

(2) Considerable discomfort

(3) Moderate discomfort

(4) Near comfort but slightly cool

(5) Pleasant, no thermal discomfort

(6) Near comfort but slightly warm

(7) Moderate discomfort

(8) Considerable discomfort

(9) Unbearably hot, extreme discomfort

13. The "VER" layout (Fig. 3.6.n) contains two *seekBars* to indicate the vehicle engineering rating scale for the driver and the passenger.

    Each *seekBar* has the values from 1 to 10. These scales show the performance of the climate control system in 8 states as follows:

    (1-2) Not Acceptable

    (3-4) Poor

    (5) Borderline

    (6) Acceptable

    (7) Fair

    (8) Good

    (9) Very Good

    (10) Excellent

14. The last layout of the test is called "Continue" (Fig. 3.6.o) which contains an *editText* to enter the end location and two *buttons* that user can choose to continue with another test or to exit the app.

Evaluation

The software prototype as an integration of defined layouts has bridged all initial sketches to the app.

- **Evaluation with Users**

  To test it with the users, we organized some evaluation sessions with the same techniques as before; silent observation, think aloud and constructive interaction.

  In the following, users feedback, our observation findings and recommendations are outlined.

  1. Users were confused in the "Dummy" layout (Fig. 3.6.i) as they did not get any feedback after indicating the spots.
  2. They asked to have a preview of the output file at the end of each test.
  3. They asked to replace the "Comfort" layout (Fig. 3.6.h) with the "Thermal" layout (Fig. 3.6.m) as these two layouts contain the same information, which should be indicated in each record not as a general information at the end of the test.
  4. We recommended involving voice record besides comments in the "Comment" layout (Fig. 3.6.j), to have more descriptive comments in a short time.
  5. We recommended to assign dark blue to dark red for a correct mapping of thermal comfort scales on *seekBars* which was refused as this color scheme is standard in the Ford e.g. red expresses discomfort whether it is hot or cold, and gradient to green is for progression to comfort.
  6. We recommended using *seekBars* instead of *checkBoxes*, for weather condition as a constraint for overlapping of choices.

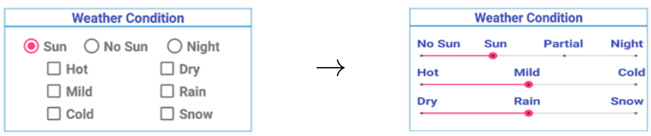


Figure 3.7: Replacement of *checkBoxes* with *seekBars*

- **Heuristic Evaluation**
  - To add notification for starting the new record on the anticipated time for each test, in case user was out of the app.
  - To replace *numberPickers* with *editTexts* to enter the value of the temperature.

    We considered that *numberPickers* may look better but for a wide range of data, it takes time to reach the value on the wheel.

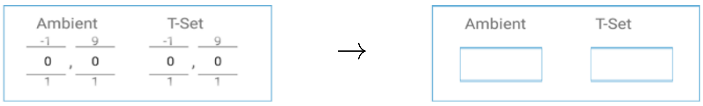


Figure 3.8: Replacement of *numberPickers* with *editTexts*

Second Iteration

At this level, we implemented the prototype horizontally, meaning that the "Comfort Assessment" section was added to the UI, we had an evaluation session with the users and the results were as follows:

1. To remove the "VER" layout from the manager section.
2. To remove Control-curves test from the manager section.

For the sake of simplicity, till this iteration, the app only had in-car temperature and ambiance as objective data in the design and other data such as blower volts, RPM, panel and floor discharge, which will be recorded by users from different sensors and instruments, were added in the high fidelity prototype.

This level of our project was one of the most influential approaches which let us have a clear overview of the users mental model, have a wide analysis of the system and lead us to a new iteration as a high fidelity prototype.

3.3.3 High Fidelity Prototype

For the high fidelity prototype which had the closest resemblance to the final app in terms of detailed appearance and functionality, the software prototype was fine-tuned to have a functional and graphical interactive prototype which included activities with their comprehensive functionalities, layouts, navigation schema, the interactive behavior of the elements and entry effects.

Definition: *Activity*

ACTIVITY:
Activities are one of the fundamental building blocks of apps on the Android platform. They serve as the entry point for a user's interaction with an app and are also central to how a user navigates within an app or between apps (Android [2016a]).

As the second iteration of the high fidelity prototype was the last iteration of the design process, and only some minor changes had been made to the first iteration, this section is about the final design although the minor changes of the first iteration will be discussed in the evaluation section.

In this section, we describe the activities' functionalities and upgraded style and aesthetic of their layouts in detail.

The App Activities

All the activities' classes are developed in Java, in which the interaction behavior of the app implemented.

All the layouts are declared in XML, in which the UI elements implemented.

Most of the drawable files are created in Adobe Photoshop[3] and some of them are created programmatically e.g. *progressDrawable* of *progressBar*.

[3]https://www.adobe.com/products/photoshop.html

The core design features of the activities and their layouts are as follows:

- The main color schemes are light gray and dark blue.
- The used font is Sans-serif.
- *AlphaAnimation* class is used for the *buttons'* click feedback by changing their opacity.
- The device *back button* is disabled for all screens which will be described in the evaluation section.

1. Main Activity

The app starts with the main activity and its layout contains a *button* to start the lifecycle of the app.

To provide a familiar entry point for the users we created the following design components in its layout (Fig. 3.9).

The background of the screen is the word cloud of the Ford vehicles' names. For this purpose, we tried different online word cloud generators such as Wordle[4] and Jason Davies[5] but as they were not customizable as desired, we created the background in Photoshop, moreover, the Ford font is used for the main *textView* on the screen and also the logo of the Ford is placed on the top.

The *button* view is similar to vehicle's ignition button with the "START APP" as its label to prevent the misleading affordance of the *button* based on the Gestalt law of experience.

At this point, the most important aspect to develop the activities was to know when and which method of the activity lifecycle should be called in order to navigate through, out or back to the activities, in other words, to know the state of the transitions and call a right method (Fig. 3.10).

[4]http://www.wordle.net/
[5]https://www.jasondavies.com/wordcloud/

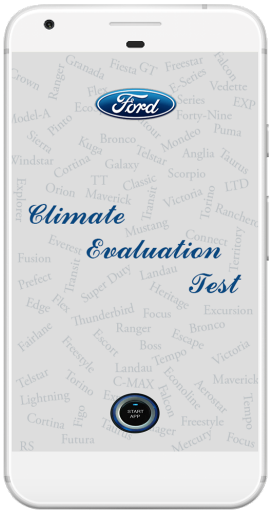


Figure 3.9: The final layout of the "Main" activity

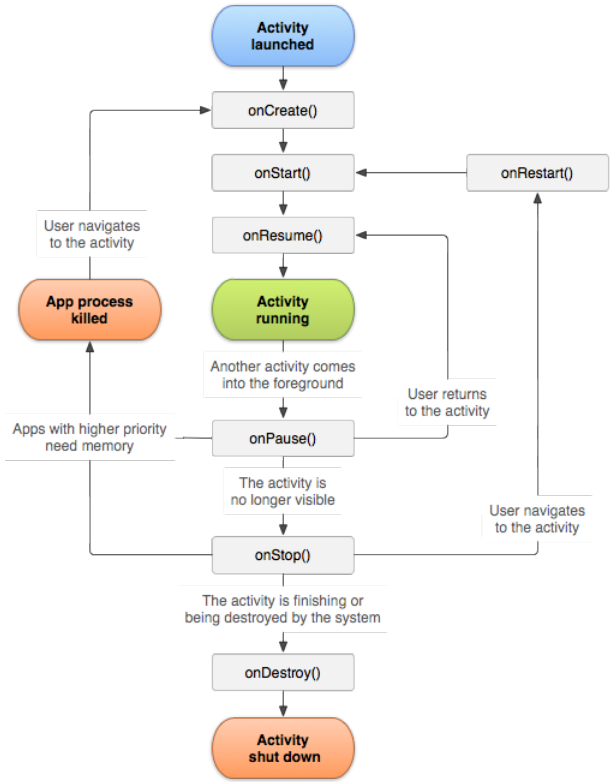


Figure 3.10: Visual representation of the activity lifecycle (Android [2016b]

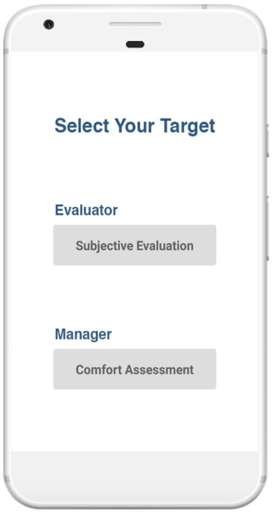


Figure 3.11: The final layout of the "Target" activity

2. Target Activity

This activity (3.11) contains two *buttons* which are for selecting the target user of the app, they also point to the type of the evaluation as their label.

3. Test-mode Activity

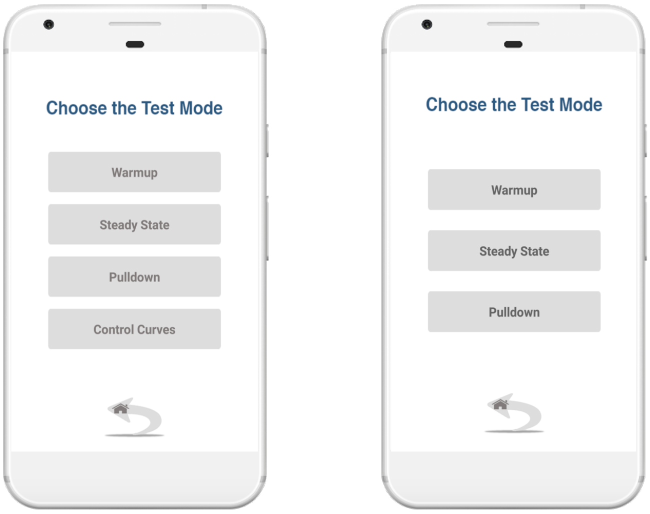
In this activity (Fig. 3.12), in addition to the four test-mode *buttons* of the old design, a new *button* is added.

As the *back button* of the device was disabled, in our heuristic evaluation, we decided to add another *button* to back to the "Target" activity to avoid the potential action-based slips e.g. manager chooses the "Subjective Evaluation" instead of "Comfort Assessment" in the "Target" activity.

The shape of the new *button* differs from the test-mode *buttons* to prevent the violation of the Gestalt law of similarity.

4. General Information Activity

In this activity (Fig. 3.13), in addition to six *editTexts* of the old design, we added a tiny clear *button* to each *editText* with regards to the CMN model e.g. deleting a 10 letter word by tapping only one key has around 240 ms percep-



(a) Evaluator **(b)** Manager

Figure 3.12: The final layout of the "Test-mode" activity

tion time while this could be up to 2400 ms for 10 times of tapping the delete key.

Moreover, we considered the Fitts's law for increasing the movement speed by decreasing the distance; by tapping the enter key on the keyboard in each *editText*, the cursor will be moved to the next one.

Furthermore, by tapping on each *editText* its hint will be shown as a small title above the entered text, which catches the attention to the blank fields.

Finally, keyboard and cursor will be hidden when the focus is out of the *editText*.

5. Weather Condition Activity

In this activity (Fig. 3.14), the *editText* has the same properties as described *editTexts*, an icon added besides the system time view, *radioButtons* upgraded to icons which are grouped in three rows along with three *seekBars*.

The *seekBars' thumb* and *progressDrawable's* colors will be changed by changing the progress, and the state will be

(a) Evaluator **(b)** Manager

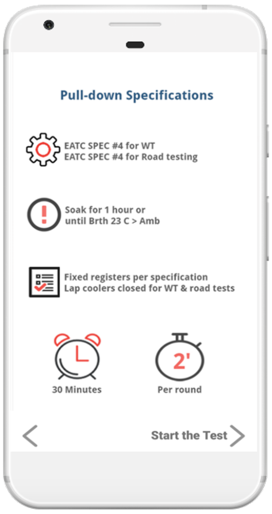
Figure 3.13: The final layout of the "General Information" activity

Figure 3.14: The final layout of the "Weather Condition" activity

toasted e.g. if the progress of the "Sun position" *seekBar* changes to "Night" the *thumb* and *progressDrawable* will be black and "Night" will be toasted.

(a) Warm-up

(b) Steady-state



(c) Pull-down

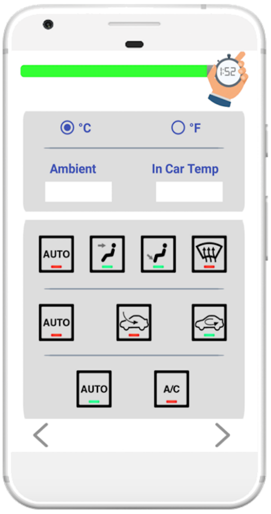
(d) Control-curves

Figure 3.15: The final layout of the "Specifications" activity

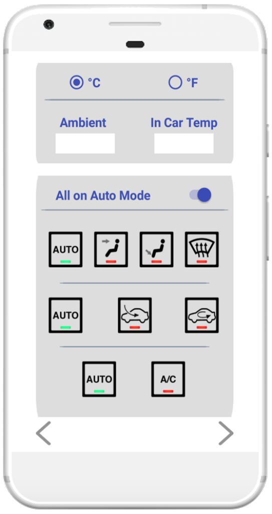
6. Specifications Activity

In the layout of this activity (Fig. 3.15), the test specifications, rules, and settings are described visually which are grouped based on Gestalt law of proximity, and the purpose of this layout is to avoid the rule-based and knowledge-based mistakes.

As the next activity is the start of the actual test, a *textView*, "Start the Test", is placed besides the *next button* as a signifier.



(a) Evaluator



(b) Manager

Figure 3.16: The final layout of the "HVAC" activity

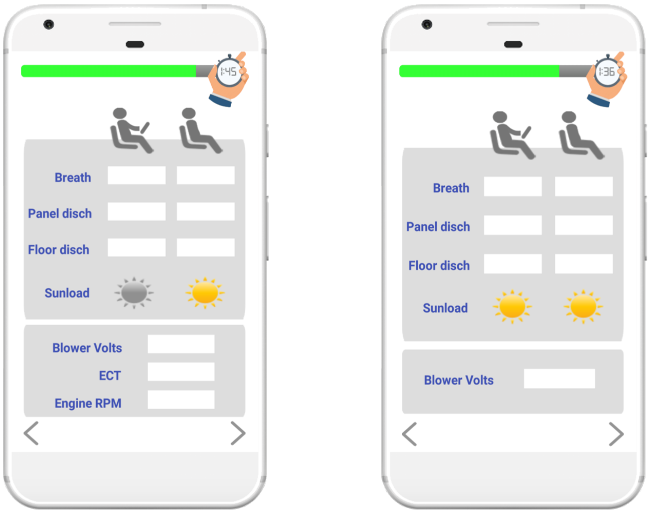
7. HVAC Activity

In this activity (Fig. 3.16), the *progressDrawable* color will be changed by the the fraction of the time, it starts with green and in the middle turns to yellow and in the last quarter turns to red.

Two *radioButtons* are placed for choosing the temperature scale, and two *editTexts* for entering the temperature.

To simulate the HVAC panel, *radioButtons* and their labels are upgraded to clickable icons and their shapes are adapted from the Ford user manual to be the same as the icons on the vehicle which are grouped in three rows. Each of these icons has an indicator as a light on them which shows if a setting is on or off. The first row icons behave as *checkBoxes*, and the other two rows behave as *radioButtons*.

The use of familiar icons will empower users to find the target faster as they use the knowledge in the head instead of the knowledge in the world.



(a) In Warm-up test **(b)** In other tests

Figure 3.17: The final layout of the "Sensor Data" activity

8. Sensor Data Activity

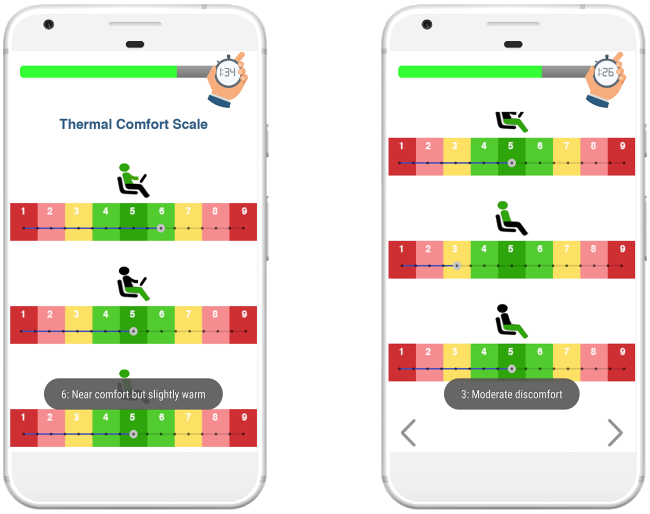
In this activity (Fig. 3.17), columns of *editTexts* are separated by passenger and driver icons. Other *editText(s)* are separated based on Gestalt law of closure.

Toggle *buttons* for the "sun-load" data upgraded to clickable icons as a gray and yellow sun to indicate yes or no respectively.

9. Thermal Comfort Activity

In this activity (Fig. 3.18), the four vertical discrete *seekBars* with colorful labels and numbers to indicate the states upgraded to horizontal *seekBars* with highlighted numbers. The gradient of colors of states, as a cultural analogy of mapping is known to the users and is in compliance with the known rule in Ford which green to red means the pleasant to unbearable state and vice versa. The states will be toasted when the progress changed. Each *seekBar* can be differentiated from others with an icon for its purpose.

The next and back *button* can be accessed by scrolling down to the bottom of the layout.



(a) Evaluator (driver) **(b)** Evaluator (passenger)

Figure 3.18: The final layout of the "Thermal Comfort" activity

10. Dummy Activity

In this activity (Fig. 3.19), the illustration of dummies as a perceptual analogy of mapping of the body is adapted from TAC[6], *imageButtons* are used to show each part of the body, the purpose of using *imageButton* instead of the clickable image is to have a clear feedback when tapped.

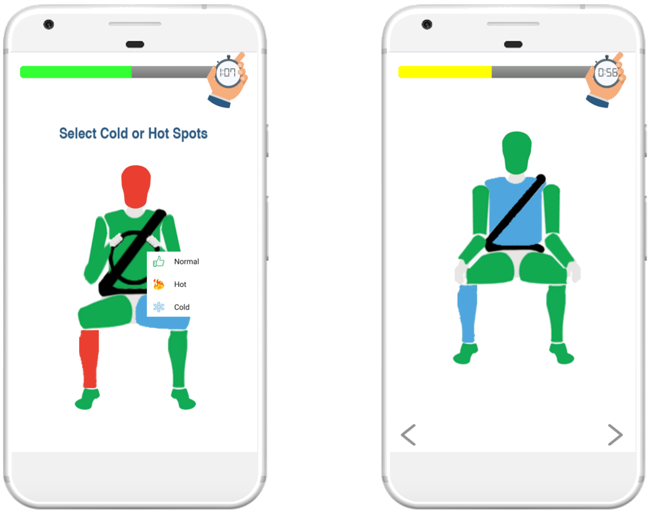
The menu that appears when *imageButton* tapped consists of three icons along with labels to choose the hot, cold and normal state, by choosing one of the items the color of *imageButton* will be updated as red, blue or green respectively. The green color is considered to provide undo for possible slips.

11. Comment Activity

In this activity (Fig. 3.20), in addition to the *editText* for the comments, four *buttons* to record, stop, pause and play are designed.

Play and record *buttons* are visible and the other two are in-

[6] http://www.howsafeisyourcar.com.au/2014/Ford/Fiesta/



(a) Evaluator (driver) **(b)** Evaluator (passenger)

Figure 3.19: The final layout of the "Dummy" activity

visible.When the record *button* tapped its *button* will be invisible and stop *button* will be visible instead and the play *button* will be disabled. Similarly, when the play *button* tapped, it will be invisible and pause *button* will be visible instead and the record *button* will be disabled.

The use of visibility and invisibility of *buttons* provides suitable feedback and signifiers to inform the users about the possible actions.

At the first time of using the app, the permissions for storage access and record audio will be requested.

The recorded file will be saved on the SD card as described in Section(3.6.5).

If the comment is too long the *editText* will be scrolled.

12. Countdown Activity

In this activity (Fig. 3.21), the background image is an illustration of an air conditioner components which adapted from CarParts.com[7].

[7] https://www.carparts.com/classroom/ac1.htm

(a) Initial state **(b)** Recording **(c)** Playing

Figure 3.20: The final layout of the "Comment" activity

As this activity is the last activity of the current record, user can choose to wait for the next record, to start the next record immediately or to end the actual test.

For these approaches three icons along with *textViews* are placed, the first *textView* shows the remaining time of the current record and when the time is over the *textView* will be invisible and a *button* to start the next record will be visible, for the other two options two *buttons* and *textViews* are placed as well, see Fig. 3.21.

Moreover, when the time is over, a notification will be shown in the notification bar with the logo of the app and "Times Up" as the title and "Time to start new record" as the message which by tapping on it, the new record will be started (Fig. 3.22).

13. VER Activity

In this activity (Fig. 3.23), *seekBars* have the same properties as the *seekBars* in "Thermal Comfort" activity.

There is no back *button* in this activity as the actual test is over.

Figure 3.21: The final layout of the "Countdown" activity

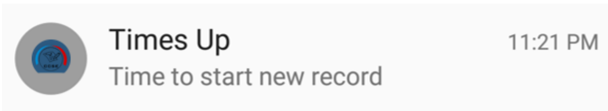


Figure 3.22: The notification which will be shown when the time of the test is over

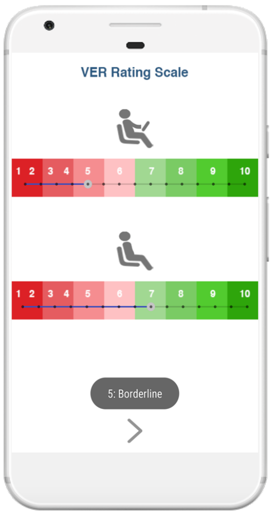


Figure 3.23: The final layout of the "VER" activity

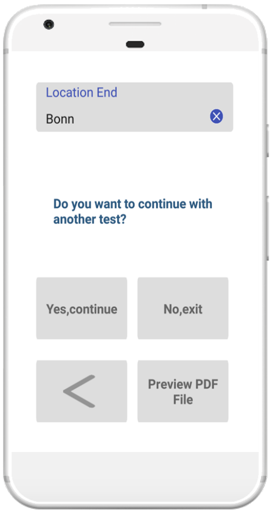


Figure 3.24: The final layout of the "Continue" activity

14. Continue Activity

In this activity (Fig. 3.24), the *editText* has the same properties as described *editTexts*.

As this activity is the last activity of the current test, user can choose to continue with another test which starts the "Target" activity, to exit from the test which shows the "Exit" activity, to navigate back to the "VER" activity or to preview the output file of the current test.

14. Exit Activity

This activity (Fig. 3.25), includes a *button* to stop the app and when tapped all the app processes and data will be destroyed and the home screen of the device will be shown.

The Logo

The logo idea is adapted from A/C knob in HVAC system and contains an evaluation symbol in the center (Fig. 3.26).

Evaluation

In the evaluation sessions of both iterations of the high fidelity prototype, the interaction of the users with the app

Figure 3.25: The final layout of the "Exit" activity

Figure 3.26: The Logo of the CCSE App

showed that they have a proper conceptual model of the app, and the design is explicit to them whereas there were some suggestions and modifications as follows:

First Iteration

- To consider half fractions for VER and Thermal comfort values.
- To change the navigation scheme in "Continue" activity for the "Yes, continue" *button* which is better to navigate to "Target" activity instead of "Test-mode" activity.
- To add two options for the immediate start of the new record and immediate end of the actual test in "Countdown" activity.
- To have more visible feedback for the *buttons* in

"Comment" activity rather than *button* effect.

- In "HVAC" activity, the row of the airflow modes as defrost, panel, floor, and auto *imageButtons* should behave as the *checkBoxes* rather than *radioButtons*.

All these changes have been discussed in the previous section, (see Section "The App Activities").

Second Iteration

The requested minor changes and observations of the first iteration led us to complete the prototype vertically which consists of the process of output generation along with enhancement of the design based on the user feedback. The logo of the app was also designed in this iteration.

- **Heuristic Evaluation**

  When the process of output file generation was completed, we did a heuristic evaluation and concluded that the device *back button* should be disabled.

  As the hierarchical relationships between screens were important because of entering their data into the output file, reverse chronological order of the back *button* could affect the output result by interfering in the activities lifecycle, moreover, in "VER" activity, the next activity to the actual test, user should not be able to navigate back to the actual test as the test records duration and functions are over and *back button* of the device should be disabled there as a lockout forcing function.

  So as to have a clear conceptual model, the *back button* disabled for all screens.

- **Constructive Interaction Evaluation**

  In order to complete our second iteration, we had a constructive interaction evaluation with users to test the usability of the app. As there were no issues, this iteration regarded as the last iteration of the iterative design.

Finally, we provided the Android Package (APK) file to the users as the beta version of the app and after they test it in the filed, we got their positive feedback.

The alpha version of the app released after our last heuristic evaluation, in which we checked whether the functional and non-functional requirements of the app were successfully fulfilled which are discussed in the next chapter.

3.4 Output File

To create the PDF file (Fig. 3.27), different libraries and packages are used and also different ways to save the data to access them later were surveyed.

3.4.1 SQLite

At the beginning, we used a local SQLite[8] database to save the data and create the output file from it. Algorithm 1 shows the implementation of saving and retrieving process of an actual test values as a pseudo code.

3.4.2 SharedPreferences API and 2D Array

Throughout the work, we decided to use the SharedPrefrences API to save and retrieve data in the form of key-value pairs for each record and a 2D array as a container to store data for all records of a test. Algorithm 2 shows the implementation of this process as a pseudo code.

As it can be concluded, algorithm 1 composes of six steps which are create, define, insert/update/remove, read and find, and write to the PDF file, but algorithm 2 is more straightforward and composed of four steps which are save, retrieve, write into the array and write into the PDF.

[8]https://www.sqlite.org/index.html

Steady State Drive

Filename: R6-32 | **Program:** ss321 | **Engine:** Fi111 | **Vehicle:** Tourneo
Date: Nov 14, 2017 | **Time:** 10:58:01 AM | **Driver:** Anna Patterson | **Location Start:** Köln
Weather Condition: Mild,Sun,Dry. | **Passenger:** Bryan Stevens | **Location End:** köln Ehrenfeld

Time	Display Amb	Dr Comfort Upr/Lower	Ps Comfort Upr/Lower	Breath Dr/Ps	HVAC Mode	AC Mode	Recirc Mode	Panel disch Dr/Ps	Floor disch Dr/Ps	Blower Volts	In Car Temp	Sunload Dr/Ps	Marked Spots Dr	Marked Spots Ps	Comments
10:59	12	6 / 6	4 / 4	23,4/23,3	panel	auto	fresh	12,8/12,8	-/-	5	10	No/Yes	(1:Hot)	(1:Hot)	
11:04	22,8	6 / 6	4 / 4	22,7/22,7	panel	on	recirc	13/12,5	-/-	5	27,3	No/No	(9:Hot)	(1:Hot)	Panel discharge at the lowest obseved
11:09	23,7	4 / 4	4 / 4	23,2/22,8	panel	auto	auto	12,8/13	-/-	5	27	No/No		(2:Hot)	
11:14	24,1	6 / 6	6 / 6	23,2/23	panel	auto	auto	12,8/13,7	-/-	5	27	Yes/Yes	(2:Cold)	(1:Cold)(8:Cold)	
11:19	26	5.5 / 6	5.5 / 4	22,1/23,2	panel	on	auto	11,8/11	-/-	5	26	No/No			
11:24	26,8	4 / 5.5	5.5 / 5.5	22,1/20,2	auto. panel	auto	recirc	12/12,3	-/-	5	27	No/No			Engine load is high
11:29	25,3	4 / 4	5.5 / 5.5	22,1/22,3	panel	auto	recirc	11,8/12	-/-	5	25,8	No/No	(4:Hot)	(3:Hot)	

D/S VER:	6
P/S VER:	6

Marked Spots: 1 = Face , 2 = Chest, 3 = Left Lower-arm, 4 = Right Lower-arm, 5 = Left Upper-arm, 6 = Right Upper-arm, 7 = Left Lower-leg, 8 = Right Lower-leg, 9 = Left Upper-leg, 10 = Right Upper-leg, 11 = Left Feet, 12 = Right Feet

(a) Evaluator

Steady State Drive

Temperature values of this test are based on °F

Program: R1-3 | **Engine:** mus-3275 | **Vehicle:** Mustang
Date: Nov 14, 2017 | **Time:** 10:07:10 AM | **Driver:** John Lehman
Weather Condition: Cold,No Sun,Snow. | **Location Start:** Aachen | **Location End:** Bonn

Display Amb	T-set	HVAC Mode	AC Mode	Recirc Mode	Thermal Comfort Upr / Lower	Marked Spots	Comments
24,9	37,9	floor. panel.	on.	fresh.	7.0/6.5	(6:Hot)(8:Hot)(10:Hot)	Abnormally hot on the right side.

Marked Spots: 1 = Face , 2 = Chest, 3 = Left Lower-arm, 4 = Right Lower-arm, 5 = Left Upper-arm, 6 = Right Upper-arm, 7 = Left Lower-leg, 8 = Right Lower-leg, 9 = Left Upper-leg, 10 = Right Upper-leg, 11 = Left Feet, 12 = Right Feet

(b) Manager

Figure 3.27: Generated test sheets of steady state drive by the CCSE application

Moreover, the amount of data are not large and could be considered as a small collection that can be handled by SharedPrefrennces API.

Finally, for the sake of clarity and speed we chose to use SharedPrefrences API along with 2D array.

Algorithm 1 Read and Write in SQLite

```
1. create DatabaseHelper.java
      for all activities do
         define table
         for all tables do
               define columns
         end for
      end for

2. in each activity
      if the next button tapped then
         get data from activity
         insert data into the associated cell of the table
         go to the next activity
      end if

3. in the last activity
      if preview button tapped then
         for all cell in PDF do
               read data from the associated table and cell
               write data to the cell in PDF
         end for
      end if
```

Algorithm 2 Read and Write, in a 2D Array and SharedPrefrences API

```
1. in each activity
      if the next button tapped then
         get data A from activity
         save data (A, A1) to the SharedPrefrences
         go to the next activity
      end if

2. in the last activity of record N
      get data with the key A1 from the SharedPrefrences
      write A as the Array[N][A1] element

3. in the last activity
      if preview button tapped then
         write the array into the PDF
      end if
```

3.4.3 iText

iText[9] Java library is used to create and manipulate the PDF file.

Imported iText Classes

- com.itextpdf.text.PageSize
- com.itextpdf.text.Rectangle
- com.itextpdf.text.Document
- com.itextpdf.text.DocumentException
- com.itextpdf.text.pdf.PdfPTable
- com.itextpdf.text.pdf.PdfPCell
- com.itextpdf.text.Phrase
- com.itextpdf.text.Chunk
- com.itextpdf.text.Element
- com.itextpdf.text.Font
- com.itextpdf.text.BaseColor
- com.itextpdf.text.pdf.PdfWriter

In order to have a similar document to the test sheets that evaluators used to fill, the following steps were considered:

- A table with the width of 95 percent of an A4 landscape document created.
- Different methods to create different cells of the table defined as follows:
  - A method to create the title cell which is the name of the test.
  - A method to create the header cells which are the general information of the test.

[9] https://itextpdf.com/

- A method to create the title cells which are the columns' titles.
- A method to create the cells which are the values recorded during the actual test.

and,

- A line at the bottom of the document as the keys for the marked spots, e.g. 6 = Right Upper-arm.
- A line at the top of the table which appears if the values are in Fahrenheit scale.

3.4.4 Java Libraries

Different Java libraries are used for the output stream.

- The java.io[10] package which provides classes for system input and output through data streams and file system.
- The java.util.List interface which represents an ordered list of objects, so the elements of a list can be accessed in a specific order, or by an index (Jenkov [2015]).

Imported Java.io and Java.util Classes

- java.io.File
- java.io.FileNotFoundException
- java.io.FileOutputStream
- java.io.OutputStream
- java.util.List

[10] https://docs.oracle.com/javase/7/docs/api/java/io/package-summary.html

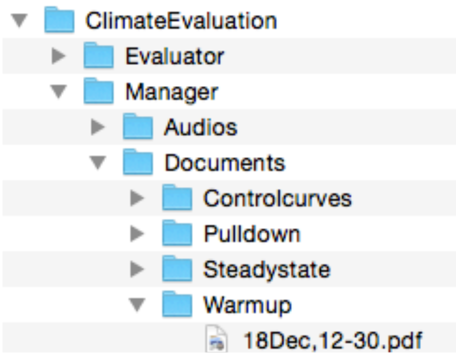


Figure 3.28: Presentation of a stored file in a hierarchical order

3.4.5 Hierarchical Ordered Output Files

For naming the PDF and audio files, the time of the test as a SimpleDateFormat `(ddMMM, HH_mm)`, e.g. 18Dec, 12-30.pdf is used.

The file will be stored on the SD card in a hierarchical order.

To store the PDF file, a root folder which is called "ClimateEvaluation" in which a folder regarding the target user as "Evaluator" or "Manager", a subfolder regarding the type of file as "Documents" or "Audios" and then a subfolder regarding the test-mode name e.g. "Warmup" will be created if they do not already exist. (Fig. 3.28).

Chapter 4

Evaluation

"We cannot solve our problems with the same thinking we used when we created them."

- Albert Einstein

4.1 Non-functional Requirements

By considering non-functional requirements, we defined how the system should work.

In the Table 4.1, some of the non-functional requirements with regard to the context of the app are listed and evaluated.

4.2 UI and Usability Testing

To ensure that the UI is interactive, easy to understand and predictable, we defined what the system should do and how the UI should behave.

The list of different input controls, navigational, informational and other elements and their tests description is shown in the Table 4.2.

Criteria	Description	Evaluation
Usability	The system should be interactive and easy to learn for the users to collect data.	As the DIA cycle is used to develop the app incrementally and with regards to the testing of the UI and evaluation of its functionality in each iteration, and based on the observation and positive feedbacks in final iteration the app could be regarded as a usable system.
Reliability	The system should provide an accurate output from the entered data.	As the generated output was checked for more than 50 tests and the values were the same as the recorded ones, the app could be considered as a reliable system.
Responsiveness	The system should be ready to respond possible interruptions.	With regards to the use of the *onPause* method and other required methods of activity lifecycle, the app handles different interruptions such as incoming calls, low battery notification.
Scalability	The system should deal well with increased volumes and workloads.	With regards to the use of SharedPreferences APIs to save and retrieve data, the app is able to serve as many as tests and records for each test as desired.
Compatibility	The system should support different screen sizes and pixel densities.	As the UI is developed in *constraintLayout* which offers seamless positioning of the elements across different screen densities and sizes, it could be regarded as a compatible system.
Backward Compatibility	The system should support older versions of Android.	As minSdkVersion of the app is 16, it is usable on all versions down to API level 16 which allows the app to be used on about 98% of all Android devices (Android [2011]).
Performance	The system should have fast response time.	Flattening the view hierarchy of layouts and avoid view nesting by using *constraintLayout* and fast data accessibility by using SharedPrefrences APIs minimize the response time of the app, so it has a fast performance.

Table 4.1: Non-functional Requirements Evaluation

Object	Description
EditText	To check if the keyboard and the cursor are shown when focused.
	To check if the keyboard and the cursor are hidden when tapped outside of it.
	To check if the cursor is moved to the next *EditText* when enter key tapped.
	To check if the entered text is cleared when its clear button tapped.
	To check if the hint text is changed to a title when edited.
Button	To check if it has the visual feedback when tapped.
	To check if it is discoverable.
	To check if its label is correct (if it has).
	To check if the visibility and invisibility function of buttons in the "Comment" activity works correctly.
	To check if next and back buttons have proper navigation.
	To check if device *back button* is disabled in all screens.
	To check if the *toggleButton* states work properly.
ImageButton	To check if it has the visual feedback when tapped.
	To check if there are no overlaps between edges of the body parts, in "Dummy" activity.
	To check if the body part menu is shown when tapped, in "Dummy" activity.
	To check if the color corresponds to the selection of the menu item, in "Dummy" activity
	To check if it has *checkBox* behavior in the first row, and *radioButton* behavior in the other rows in HVAC panel.
SeekBar	To check if the color of the *thumb* and the *progressDrawable* are changed based on the the progress change, in "Weather Condition" activity.

Continued on Next Page ...

Table 4.2 – Continued

Object	Description
	To check if the *thumb* is on the default state.
	To check if it is mapped correctly.
ProgressBar and Timer	To check if its color corresponds to the time.
	To check if trends are visible and mapped correctly
	To check if both digital and analog timers correspond to each other.
	To check if it is mapped correctly.
Notification	To check if the notification will appear, when the time is over, if the user is in or out of the app.
	To check if it sounds, and appears in the notification bar.
	To check if it has the right message and logo.
	To check if the next record starts when it is tapped.
Toasts and Messages	To check if the selected body part and its state will be toasted, in “Dummy” activity.
	To check if changed state of the *seekBars* is toasted correctly.
	To check if the user permissions are requested for the first time use.
	To check if the granted permission works properly.
All Layout Elements	To check if they are grouped based on the gestalt laws.
	To check if they are in the right positions.
	To check if their information content is proper.
	To check if the elements are relevant to the selected test-mode and the specifications and duration corresponds to it.
	To check if data is restored to the activity when navigating back to it.
	To check their consistency.

Continued on Next Page ...

Table 4.2 – Continued

Object	Description
PDF	To check if the actual test entries in the tables are positioned properly.
	To check if the general information entries are positioned properly.
	To check the properties of the stored file such as name and order.
	To check if all collected data entered to the PDF file correctly

Table 4.2: UI and Usability Testing

Chapter 5

Summary and Future Work

The App Overview	
Title:	CCSE
Description:	Program to collect assessment data of climate control system during the test drive.
Platform:	Android
Programming Language:	Java
IDE:	Android Studio
API Level:	16 or higher
Required Android:	4.1 and up
Language:	English
Permissions:	Storage Access, Microphone

Table 5.1: Overview of the CCSE App

5.1 Summary and contributions

The key aspect discussed in this work is the development process of an Android application to cover the difficulties of climate control system evaluation in the traditional way. In other words, to replace the pen and papers with mobile devices and digital documents.

With regard to the brainstorming sessions, system study and the findings of interviews, we identified the users and the procedures of test drives evaluation and defined our main approach, to solve the potential problems.

The main approach to develop CCSE app was the iterative design to build the final product progressively using three levels of detail in prototypes.

In the beginning, we presented our design to the users using the paper prototype to gather the initial feedback and changes. Subsequently, we designed the software prototype by implementing a basic application that contained layouts and their elements as placeholders to discover the human errors and design problems.

Finally, we integrated our software prototypes horizontally and vertically to finalize users interactions and explore the potential issues of the design.

As a result of 10 iterations of CCSE app development cycle with respect to different laws and principles of the design to meet all the expectations of the application and solve the issues of the assessment with pen and paper, CCSE provides simple procedures for users to collect the objective and subjective assessment data in compliance with the rules and specifications of four different test-modes and generates the output file from the entered data as the evaluation test sheet.

Moreover, it organizes the test sheets and the recorded comments in a hierarchical arrangement of files for the ease of accessibility.

5.2 Future work

As the scope of this work was to develop an app for collecting data by users, we focused on the design and implementation process of the application, however, there were other projects, related to the electrical area, running in the Ford to read the data from the instruments and sensors.

So as we provided a good hierarchy of the activities files with the implicit names and comments, extending this work in the future and making the modifiability is easy to get the objective data automatically from the car by data logging from the CAN Pack by Bluetooth.

The app is built to be used by a single evaluator who records the data for the driver and himself as the passenger. This could be extended in a way that the evaluator could also enter the data while driving e.g. by projecting the activities of the app into the windshield which user could interact with it using gestures and voice input.

Bibliography

A/C Pro. Evolution of Auto Air Conditioning, 2014. URL `http://acprocold.com/blog/evolution-auto-air-conditioning/`.

Android. Layouts, 2009. URL `https://developer.android.com/guide/topics/ui/declaring-layout.html`.

Android. Platform Versions, 2011. URL `https://developer.android.com/about/dashboards/index.html`.

Android. Activities, 2016a. URL `https://developer.android.com/guide/components/activities/index.html`.

Android. Activity Lifecycle, 2016b. URL `https://developer.android.com/guide/components/activities/activity-lifecycle.html`.

Rafael Ballagas. *Bringing Iterative Design To Ubiquitous Computing: Interaction Techniques, Toolkits, and Evaluation Methods*. PhD thesis, RWTH Aachen University, Aachen, August 2007.

Gartner, Inc. Digitalization - Gartner IT Glossary, 2017. URL `https://www.gartner.com/it-glossary/digitalization/`.

Jakob Jenkov. Java Collections - List, 2015. URL `http://tutorials.jenkov.com/java-collections/list.html`.

PantherBB. Ford EATC (Electronic Automatic Temperature Control), 2005. URL `http://www.pantherbb.com/p71interceptor/eatcswap/index.html`.

Wikipedia. HVAC Control System, 2017. URL `https://en.wikipedia.org/wiki/HVAC_control_system`. [Accessed: 20-Nov-2017].

Index

Typeset March 11, 2018