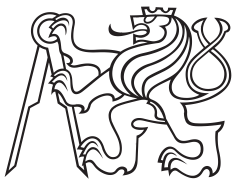
Master's Thesis



Czech Technical University in Prague

F8

Faculty of Information Technology
Katedra teoretické informatiky

Tablet infotainment system

Bc. Michael Bláha

January 2016

Supervisor: Ing Jan Šedivý, CSc.

/ Declaration

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

Abstrakt / Abstract

Tento dokument je pouze pro potřeby testování.

This document is for testing purpose only.

Contents /

Chapter 1
Introduction

Some stuff

1.1 Project

stuff

1.1.1 Motivation

.

1.2 Assignment analysis

1.2.1 Review existing Android applications for in-car use

One of the key approches in research project is reviewing the existing progress in the given field. Reviewing existing applications helps understanding the topic, seeing the bigger picture, learning from mistakes of others and last but not least, getting general idea about competition.

1.2.2 Review and analyse User Interface development methods for in-car infotainment applications

Cosindering the car environment, the user interface must deal with a lot of different problems than usual. This task should review existing User Interface development rules and apply them to the car environment, then analyse them and choose proper method for car-UI design process.

1.2.3 Analyze the in-car OBD API and exported data

On-Board Diagnostics API is a standard API provided by modern cars for gathering various information from speed to engine temperature. This task focuses on understanding and gathering data from the OBD API.

1.2.4 Design an application system architecture for accessing the OBD data and resources

Having the data from OBD and preparing an application for displaying them, designing proper architecture is required for everything to work well. The application has to gather data, while displaying them properly without unneccesary (FIX!) delay.

1.2.5 Design a tablet User Interface for in-car use

After reviewing existing applications and UI development methods, the next goal is to create new User Interface for in-car use, while considering the constraints this environment puts on it.

1.2.6 Design and implement in-car application offering the OBD data for Android tablet platform

With everything prepared and thought through, the application will be developed based on result from all the tasks accomplished so far. In this case, the Android platform will be used as explained later in the text.

1.2.7 Perform UI and application testing and evaluate results

For best results the application must and will be tested. Both code and UI must be tested properly, using various testing approaches, such as unit tests or UI testing with reaul users in a car simulator.

Chapter 2
Analysis

2.1 Existing applications

2.1.1 Applications

2.1.2 Google AutoUI

2.2 Platforms

2.3 Android platform

2.3.1 Architecture

2.3.2 Specifics

2.4 GUI

2.4.1 Basic principles

2.4.2 Car UI differences

2.5 Server

2.5.1 Functionality

2.5.2 Data storage

2.5.3 Communication

2.6 Development and support tools

2.6.1 Development environment

2.6.2 Quality Assurance tools

2.6.3 Version system

2.6.4 Test driven development

2.6.5 Continuous integration

Chapter 3
Design

3.1 Application architecture

3.1.1 Extensibility

3.1.2 Modularity

3.1.3 Adaptability

3.1.4 AutoUI preparation

3.1.5 Platform limitations

3.2 GUI

3.2.1 Basic elements

3.2.2 UI drafts

Chapter 4
Realization

4.1 Preparation

4.1.1 Environment

4.1.2 Versioning

4.1.3 Testing

4.1.4 Scripting

4.2 Core

4.2.1 Core

4.2.2 Data storage

4.2.3 Communication

4.2.4 Optimization

4.3 Modularity

4.3.1 Requirements

4.3.2 Integration

4.4 GUI

4.4.1 Common elements

4.4.2 Multiple designs

Chapter 5
Testing

5.1 Implementation

5.1.1 Unit testing

5.1.2 Integration testing

5.1.3 System testing

5.1.4 Qualification testing

5.2 GUI

5.2.1 Heuristic testing

5.2.2 Evaluation

subsubsub

Some text text

5.2.3 Testing with users

5.2.4 Simulator

subsubsub

5.2.5 Preparations

subsubsub

5.2.6 Course

subsubsub

5.2.7 Evaluation

subsubsub

5.3 Summary

Chapter 6
Conclusion

6.1 Assignment completion

6.2 Project life cycle

6.2.1 Present

6.2.2 Future

6.3 Summary