<PiPossible>

For <Android Application and Pi-Programm>

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <22/10/19> | <1.0> | <Version 1.0> | <PiPossible> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 3

1.1 Purpose 3

1.2 Scope 3

1.3 Definitions, Acronyms, and Abbreviations 3

1.4 References 3

1.5 Overview 3

2. Overall Description 3

3. Specific Requirements 3

3.1 Functionality 3

3.1.1 <Functional Requirement One> 3

3.2 Usability 3

3.2.1 <Usability Requirement One> 3

3.3 Reliability 3

3.3.1 <Reliability Requirement One> 3

3.4 Performance 3

3.4.1 <Performance Requirement One> 3

3.5 Supportability 3

3.5.1 <Supportability Requirement One> 3

3.6 Design Constraints 3

3.6.1 <Design Constraint One> 3

3.7 On-line User Documentation and Help System Requirements 3

3.8 Purchased Components 3

3.9 Interfaces 3

3.9.1 User Interfaces 3

3.9.2 Hardware Interfaces 3

3.9.3 Software Interfaces 3

3.9.4 Communications Interfaces 3

3.10 Licensing Requirements 3

3.11 Legal, Copyright, and Other Notices 3

3.12 Applicable Standards 3

4. Supporting Information 3

# Introduction

## Purpose

This document gives a description of the PiPossible project. It explains our idea and the features of our project and the prototype. Also, it offers an overview of the used hardware and software.

## Scope

The Android application allows users to send commands to the Pi. The app allows users to control the robot.

* Move the robot
* Music control
* Light control

## Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| Term | Description |
| IDE | Integrated Development Environment |

## References

|  |  |
| --- | --- |
| Title | Date |
| Blog |  |
| JetBrain |  |
| GitHub |  |

## Overview

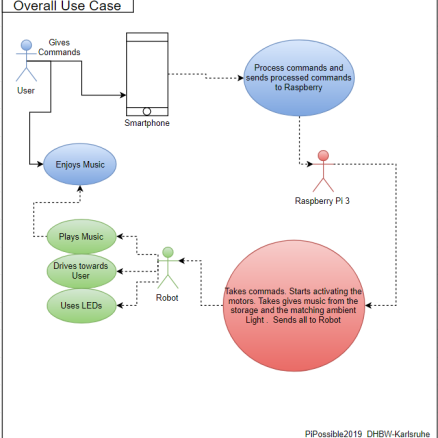
The next chapters provide information about our vision based on the use case diagram as well as more detailed software requirements.

# Overall Description

## Vision

The goal of the PiPossible project is to create a new kind of music enjoyment. The user has the ability to enjoy his music without headphones and without to be stationarity speaker. The user just enoy his music and the music follow him around. For the right feeling we use ambient lightning.

## Overall Use-Case-Diagram



## User characteristics

Our main target group are people who love to listen to music at home and want to her the music in every room depending on where they are.

# Specific Requirements

## Functionality

The key component of the project is a RaspberryPi 3 and the control application developed by us. The Raspberry itself controls the light and the speaker. All these components are assembled on a remote-controlled car.

## Usability

To do.

## Reliability

To do.

### Availability

The application will be developed for Smartphones and Tablets running Android. The robot assembled robot will be a prototype, but everyone will be able to build one by itself using oure code.

## Performance

The RaspberryPi 3 is a very powerful Microcontroller so it should handle the tasks easily. The application just needs the correct Android version (or higher), every smartphone can run it.

## Supportability

Our project is an open source project. We won’t support it after we finished it. But everyone who is interested in this project can use our base and use or extend it.

## Design Constraints

Limited by time the application will have a minimalistic design. The design of the robot prototype depends on how the components can be assembled on the base.

### Development tools

* Git: Version control system
* YouTrack: Project management application and sprint control
* Android IDE: IDE for appdevelopment
* Visual Studio: Programming of the Pi

## On-line User Documentation and Help System Requirements

## Purchased Components

* RaspberryPi 3

## Interfaces

### User Interfaces

The user controls the robot with an Android application using the touchscreen of a mobile device. To control we will implement buttons.

### Hardware Interfaces

The application will send the input via WiFi to the robot.

### Software Interfaces

The application will send the input via WiFi to the robot.

### Communications Interfaces

## Licensing Requirements

We just use open source software.

## Legal, Copyright, and Other Notices

We just use open source software

## Applicable Standards

To do.

# Supporting Information

To do.