



REQUIREMENTS DOCUMENT

Projectbalancing robot

Abstract

This document includes functional and non-functional requirements for the project about a self-balancing and app controlled robot.

Konghon Choo and Eveline Ververgaert

Table of Content

Functional requirements (MoSCoW).....	2
MUST	2
SHOULD	2
COULD	2
WOULD	3
Non-functional requirements.....	4

Functional requirements (MoSCoW)

MUST

ID*	Requirement	Finished
FA1	The app displays buttons to control the movement of the robot (forward, backward, left and right)	
FA2	The app sends commands over Wi-Fi	
FA3	The app can connect with the robot over Wi-Fi	
FR1	The robot can receive commands with Wi-Fi	
FR2	The robot can connect with the app	
FR3	The robot goes in idle mode when not connected with the app	
FR4	The robot goes in idle mode when the app has not send any commands for 10 seconds and the robot is standing still	
FR5	The robot stands still when either stop received, when disconnected or while no new command received	
FR6	The robot will always balance itself	
FR7	In idle the robot will only try to balance itself (NO riding)	
FR8	The robot can avoid static obstacles (e.g. a wall)	

*ID format: <Functional or Non-functional><App or Robot><unique number>

SHOULD

ID	Requirement	Finished
FA4	The app shows status of robot (e.g. battery level and busy or idle)	
FA5	The app can receive commands	
FR9	The robot can send commands	

COULD

ID	Requirement	Finished
FA6	App displays a button for options in which the user can select a control option (Buttons, tilt or possibly speech)	
FA7	Tilting control option	
FA8	The app is able to change speed: <ul style="list-style-type: none"> • Tilt mode: through tilting • Button mode: extra speed option (e.g. through slider) • (Possibly speech mode: through speech change speed) 	
FR10	Robot can change speed	

WOULD

ID	Requirement	Finished
FA9	Speech recognition for the 4 commands (forward, backward, left, right)	
FR11	The robot can avoid dynamic obstacles	
FR12	When in idle mode, the robot plays a song	
FR13	When in idle mode, the robot turns on LEDs	
FR14	Possible enclosure for the robot	

Non-functional requirements

ID	Requirement	Finished
N1	A form of Agile is used	
N2	Source Control tool GitHub is used	
N3	All existing test cases should be successful	
N4	All documentation should be in the same style	
NA1	Kotlin is used for application build	
NA2	Application's GUI is easy to read	
NR1	C or C++ is used for robot build	