

Version: 1.0.0

Release date: 25-11-2016

Riccardo Cattaneo 873647

Fabio Chiusano 874294

PowerEnJoy, Design Document

Table of Contents

1 Introduction 3

1.1 Purpose 3

1.2 Scope 3

1.3 Definitions, Acronyms, Abbreviations 4

1.4 Reference Documents 5

1.5 Document Structure 5

2 Architectural Design 6

2.1 Overview: High level components and their interaction 6

2.2 Component view 6

2.3 Deployment view 6

2.4 Runtime view 6

2.5 Component interfaces 6

2.6 Selected architectural styles and patterns 6

2.7 Other design decisions 6

3 Algorithm Design 7

4 User Interface Design 8

5 Requirements Traceability 9

6 Effort spent 10

7 References 11

# Introduction

## Purpose

This document describes the hardware and software architecture of the PowerEnjoy System. Therefore, it outlines hardware tiers and all parts of the software and how they will work and cooperate together.

In particular, this document contains information about:

* Architecture Design with related pattern used;
* Main components and software interface design;
* Runtime behaviour of the system
* User Interface Design

## Scope

PowerEnjoy is a car-sharing service based on a mobile web application.

It allows the user to see, thanks to the help of an external search-on-a-map handler, where the electric cars are, only if they are close to an address provided by either the user or his/her GPS Location.

Hence, it allows users to reserve an electric car and to get on board when he/she is close to it.

The car software takes into account the minutes of usage of the car, the number of passengers, the battery level and the location of release.

The system then calculates the charges the user for the ride.

The main purpose of the system is to create a new and smart car-sharing service, that incentivize virtuous and green behaviours.

## Definitions, Acronyms, Abbreviations

FROM TheraWii EXAMPLE:

1.3.1 Physical Therapy

Posture

The orientation of any body segment relative to the gravitational vector. It is an angular measure

from the vertical [1].

Balance

The dynamics of body posture that prevents falling. It is related to the inertial forces acting on

the body and the inertial characteristics of body segments [1].

Center of Mass (COM)

A specific point at which the system’s mass behaves as if it were concentrated

[1].

Center of Pressure (COP)

The point location of the vertical ground reaction force vector. It represents

a weighted average of all the pressures over the surface of the area that is in contact with the ground.

It is also called the Center of Balance (COB) [1].

1.3.2 Nintendo Wii

Wii Remote

Device that communicates through Bluetooth wireless protocol to the Nintendo Wii Gam-

ing System. Data communicated includes button press and releases, accelerometer readings, and an

Infrared (IR) LED pointing system.

Wii Balance Board

Device that communicates the COP through Bluetooth wireless protocol to the Nin-

tendo Wii Gaming System.

1.3.3 Software

Therapy

A series of tasks that is completed in one session.

Session

A given time in which a user completes a therapy.

Task

A subunit of a therapy that has an objective with success and fail criteria.

## Reference Documents

## Document Structure

Explain what chapters from 2 to 5 contain.

# Architectural Design

## Overview: High level components and their interaction

FROM TheraWii EXAMPLE:

- Description of the problem

- Technologies used

- System architecture (list components, informal)

- System operations (sequence diagram example, possibly involving all the components)

## Component view

High level class diagram of the main classes (that are the components, probably).

Diagram that shows the interactions between components.

## Deployment view

Deployment units:

- App

- Server

- Car program

Diagram that shows their interactions.

## Runtime view

Looooots of sequence diagrams.

## Component interfaces

Components class diagrams, very detailed. Text explanations of all classes: what the class does, what its methods do, design constraints, permormance issues…

## Selected architectural styles and patterns

Two tiers, event-based, MVC…

## Other design decisions

Maybe data model (class diagrams, Object Relationship diagrams) and data storage (ER diagrams, SQL stamements for tables creation).

# Algorithm Design

- GPS usage (not sure)

- Search for near cars:

1. App asks server for near cars and sends it its location;
2. Server retrieves cars locations quickly thanks to some spatial data structures (https://en.wikipedia.org/wiki/Spatial\_database);
3. Server sends the locations to the app;
4. App is happy.

- Search for near parkings: similar to search for near cars.

# User Interface Design

All the screens!

How to go from a screen to another.

Description of each screen.

Some scenario examples.

# Requirements Traceability

Table with the following columns:

- Requirement

- Description

- Design reference (all the references in this document that together satisfy the requirement).

# Effort spent

# References