WordZoo Project:

Software Requirements Specifications

[1 Introduction 2](#_Toc500501003)

[2 Overall Description 3](#_Toc500501004)

[2.1 Product Perspective 3](#_Toc500501005)

[2.2 Product Functions 3](#_Toc500501006)

[2.3 User Characteristics 3](#_Toc500501007)

[2.4 Interfaces 3](#_Toc500501008)

[2.5 Constraints 3](#_Toc500501009)

[3 Specific Requirements 4](#_Toc500501010)

[3.1 External Interfaces 4](#_Toc500501011)

[3.2 Functional requirements 4](#_Toc500501012)

[3.3 Performance Requirements 4](#_Toc500501013)

[3.4 Logical Database Requirements 4](#_Toc500501014)

[3.5 Design Constraints 5](#_Toc500501015)

[3.6 Software System Attributes 5](#_Toc500501016)

# Introduction

This document formally specifies the functions of the software created as part of the WordZoo project, and sets clear development objectives to achieve them. The format of this document has been adapted from the IEEE Guide to Software Requirements Specifications (Std 830-1993).

# Overall Description

## Product Perspective

The WordZoo system will be independent and self-contained, but will bring together several sub components. It brings together aspects of existing education video games, learning management software, and will also contain components that allow for large-scale data analysis.

## Product Functions

### The system must authenticate individual users who are school pupils, school teachers or system administrators.

### The system must present a series of educational literacy games that test and teach spelling, reading or phonics.

### The system must record the performance of school pupils within the game.

### The system must present pupil performance data to teachers, and allow for it to be extracted by system administrators.

## User Characteristics

### School Pupils

The product will be designed to be used by school pupils aged 4-7 (Reception/Y0, Y1, Y2 in UK primary school classes). It will be assumed they have little technical fluency, but can independently work with a tablet touch-screen interface. We will also aim to use graphical descriptions and avoid text instructions where possible.

### School Teachers

The product will be designed to be used by school teachers, who could be of a wide age range and with a variety of technical expertise. The system will be designed as intuitively as possible with this in mind, with clear explanations or tutorials to introduce them to the system.

### System Administrators

The product will be designed to also be used by system administrators (for account management and data extraction) who will be of high technical expertise.

## Interfaces

#### All interfaces will be intuitive and uniform

#### The interfaces for school pupils will be adapted for a very young audience, and will employ a cartoon style to make the system engaging.

## Constraints

The system will have to comply with UK and EU data protection requirements/legislation.

# Specific Requirements

## External Interfaces

### User interfaces

## Functional requirements

### School Pupil

#### The system shall present a simple login interface to pupils that requires a unique code which can be given to them by their School Teacher

#### The system shall present a menu screen which displays a link to each game

#### The system shall present the selected game

#### The system shall provide intuitive feedback within each game that indicates if they are correct on each question

### School Teacher

#### The system shall present an email/password login interface to teachers

#### The system shall provide a form to manage classes which records a name for the class and the names of pupils

#### The system shall list the class and the names of pupils and generate a login code which the pupils can use to login following ‎3.2.1.1

#### The system shall present an individual pupil detail page which shows the pupil name, and their activity in the system.

#### The system may generate intelligent feedback to teachers on each pupil with specific advice on items to teach the pupil

### System Administrator

#### The system shall provide a separate management interface which presents an email/password login to system administrators

#### The system shall list schools registered in the system with links to view the details of the school and modify or delete information

#### The system shall provide a form to add a new school

#### The system shall provide a school detail page which lists teachers at the school, and a sub-form to add new teachers and classes

#### The system will allow teachers to be added to multiple classes

#### The system shall allow the system administrators to extract anonymised performance data which can be used for analysis

## Performance Requirements

#### The system shall allow for real time access from at least 100 users simultaneously

#### The system shall respond to web requests within 5 seconds

## Logical Database Requirements

#### The system shall store information about schools

Unique ID number, name, contact email, contact telephone number, contact address

#### The system shall store information about teachers

Unique ID number, name, email, encrypted password

#### The system shall store information about classes

Unique ID number, class name, teachers linked to the class via a relation table

#### The system shall store information about pupils

Unique ID number, name, access code, and class ID number

#### The system shall store information about games

Unique ID number, game name

#### The system shall store information about words (to be used in the games)

Unique ID number, word, difficulty value

#### The system shall store information about pupil game interaction (‘sessions’ and response to words)

Unique ID number, pupil ID, game ID, timestamp

#### The system will store information about system administrators

Unique ID number, email, encrypted password

### ER Diagram

## Design Constraints

## Software System Attributes

### Availability

#### The system shall be available on a 24/7 basis

### Security

#### The system shall employ encrypted password to securely authenticate users

#### The system shall use the HTTPS protocol to securely transmit data between the server and users

### Maintainability

#### The system shall be written in a modular fashion

#### The system shall be written following common good coding standards, including commenting

### Portability

#### The system shall be able to run on any server capable of running NodeJS 8 and MySQL