
Table of Contents

1. Introduction

- 1.1 Overview
- 1.2 About this document
- 1.3 Glossary

2. General Description

- 2.1 Systems Functions
- 2.2 User Characteristics and Objectives
- 2.3 Operational Scenarios
- 2.4 Implementation Constraints

3. Functional Requirements

- 3.1 Sign up / Register
- 3.2 Login
- 3.3 Create group (Teacher)
- 3.4 Join group (Learner)
- 3.5 Practice (Learner)
- 3.6 Manage Class (Teacher)
- 3.7 Delete Account / Class (Teacher)
- 3.8 Tour Website
- 3.9 Log-out

4. System Architecture

Fig 4.1 Simple illustration of system

5. High-Level Design

5.1 Context diagram

5.2 Data Flow Diagram

6. Preliminary Schedule

1. Introduction

1.1 Overview

The system is an interactive web application that allows school teachers, children or parents to teach or learn the basics of Irish grammar in a fun and interactive way.

We decided to devote our project to the process of constructing grammatically correct Irish phrases. A student with basic vocabulary and some knowledge of grammar rules may want to practice them by experimenting with the construction of simple sentences. At this stage, it is important to make sure that the sentences are built properly, and if not, the student gets appropriate feedback.

By creating sentences, the student can learn through a "try & fail" methodology. In more advanced scenarios, feedback might include hints on the proper use of words and grammar.

- A user can check whether a word is appropriate in a certain context. Suppose the student knows that "*D'ith mé uachtar-reoite*" i.e. (*I ate ice cream*) makes sense, but can one "*D'ith mé rothar*" i.e. (*I ate bike*) ?

The basic idea behind the project is the construction of a valid sentence (wall) from a given selection of words (blocks).

The main inspiration behind the project is the Scratch programming language learning environment, in this the blocks are shaped like jigsaw-puzzle pieces so it impossible to create a syntactically incorrect program, unfortunately natural language is more complex and this will therefore be more difficult to implement.

Irish sentences are formed in the following order 'verb - subject - object', these will be color coded for beginners and become uniform as the user reaches more complex levels to add to the challenge.

The words will be separated up in tables inside a relational database based on their families and characteristics (i.e. Noun, Verb, Plural etc), allowing for the representation of appropriate categories for the user to select from.

1.2 About this document

What this specification does?

This document provides outlined functional specifications and requirements for the Word Bricks web application project. It is designed to guide system development and design, including:

- User Interface design
- Accessibility
- Database architecture
- Site functionality
- Use cases

A 'Living Document'.

This specification will change continuously as the project proceeds. We will add details and edit existing information as the back-end, middle-tier and front-end components evolve during the course of the project.

1.3 Glossary

UI : User Interface

Cascading Style Sheets (CSS): Is a style sheet language used for describing the presentation of a document written in a markup language.

Hypertext Markup Language (HTML): Is a standardized system for tagging text files to achieve font, colour, graphic, and hyperlink effects on World Wide Web pages.

Uniform Resource Locator (URL): Is a reference (an address) to a resource on the Internet.

Hypertext Preprocessor(PHP): Is a server-side scripting language.

DB: Database.

2. General Description

2.1 Systems Functions

The following is a preliminary list of the system functions. We intend for this list to be pliable, as in we can add/remove features where appropriate as we progress through the development of the system. These functions will be explained/illustrated in detail in the functional requirements section.

- Sign up

- Login (Learner/Teacher)
- Create group (Teacher)
- Join group (Learner)
- Practice (Learner)
- Manage Class (Teacher)
- Delete Class (Teacher)
- Delete Account
- Tour Website
- Log out

*Requires Login

2.2 User Characteristics and Objectives

This tool will be aimed at primary school teachers with access to computer labs or interactive whiteboards, younger students who may have access to a computer or laptop at home and parents who want to gain a basic understanding so they can help their children and get involved.

As the application will target a younger demographic whom we can safely assume will have limited technical know how, the user interface will be simplistic and intuitive. The chosen colour scheme is vital, many people suffer from colour blindness so we will have to keep this in mind when selecting colours for classification of different families of words.

2.3 Operational Scenarios

As the range of potential users varies for this application, we will need to account for the different scenarios under which the system will have

to perform. Namely the goals of the users i.e. the student will want to practice their skills where as the teacher will want to monitor their student's progress.

We intend to have two main authentication types, a "Learner" for individuals outside the classroom who want to practice Irish, a "Teacher" who wants to create a class group and monitor their progress and finally a "Student" login.

The latter is slightly different from the first two in that no personal details are required for login, this is for students wishing to join a class group. We wanted to make this as simple as possible so the only details required will be their username and the group ID provided by the teacher.

Unregistered User (*Teacher*)

We anticipate this user to be tech-savvy teacher eager to try new methods of teaching.

- Click "Sign up"
- Choose "Teacher" level account
- Fill in necessary details & activates account
- Tour site

Unregistered User (*Learner*)

This will most likely be an individual doing some practice outside school or a parent trying to pick up some basic grammar skills.

- Click "Sign up"
- Choose "Learner" level account

- Fill in necessary details & activates account
- Tour site

Registered User (*Teacher - Not logged in*)

- Click "Login"
- Select "Teacher"
- Enter (username / email address) & password

Unregistered / Registered (*Student - Not logged in*)

These students can log in to the site using the class code meaning they do not have to create an account themselves, they enter a name and are added automatically to the class group.

- Click "Join Group"
- Enter name and group ID

Registered User (*Learner (Individual) - Not logged in*)

- Click "Login"
- Select "Learner"
- Enter (username / email address) & password

Registered User (*Teacher - Logged in*)

- Create new class group / Choose existing group
- Add / Accept Students to said group
- Set challenges / View group stats
- Give feedback
- Log out

Registered User (*Student - Logged in*)

- "Join class group" using unique key provided by teacher
- Attempt set challenges
- View progress
- Log out

Registered User (*Learner (Individual)- Logged in*)

- Choose difficulty level
- Practice chosen challenges
- View progress / stats
- Log out

2.4 Implementation Constraints

Time Constraints

The project must be completed within the given time frame. We will have six weeks to develop the system while still partaking in other college modules.

User requirements

We will need to stick closely to design and accessibility principles in order to ensure the user interface is intuitive to use for younger users. Color blindness will be an issue we will have to consider at each step of the design process.

Technical Difficulties / Database Memory

Can we fit the entire Irish language? No, but in terms of scalability for the future the database memory may be an issue, for now we will be

working with freely available tools and the basics of Irish grammar will suffice.

Limitations of error prevention system

One of the main challenges for us will be the implementation of an effective error prevention system. The reason this will be such a problem is because natural language sentences are more complex than programming languages in that they can be syntactically correct but contextually incorrect. For example, "I ate a table" is a valid sentence in terms of the pattern of the sentence construction but generally does not make sense.

3. Functional Requirements

3.1 Sign up / Register

Description:

A teacher registering will have to set up their account with name, email address and password.

Students joining a class group do not need to sign up, they can login immediately instead.

A regular user will be able to sign-up by entering their full name, password and email address.

After each user has successfully entered their details they will be able to start using the application.

Criticality:

Mandatory accounts for all users ensures users can keep a record of their progress and also allows the system to track progress and in turn we can use this data to generate statistics and present them as feedback to users and teachers.

Technical issues:

Have the capacity to add new users to the database easily and have a secure and reliable method of storing password hashes.

Dependencies with other requirements:

Manage class - Unique student users represented in teachers view i.e. Graphs, Statistics.

Delete class group - Teacher delete class group then student accounts are also deleted.

Delete account - Need to create an account before it can be deleted.

3.2 Login

Description:

Teachers or regular users must login with their email and password, these will be entered into a PHP form, the password will then be hashed against the appropriate account and either accepted or rejected.

A student in a class who wishes to join a group will have to login with their full name and enter the class key to join their class group. Their teacher will provide this key. After logging in initially their progress will be stored for the next time.

Criticality:

The login is a crucial feature of the system to differentiate between users and track their progress. Without this function the user will not be able to interact with the web application.

Technical issues:

The system must be able to distinguish between account types.

Difficulties here will be making sure the database stores the correct details for each user and that all the users in the database have unique username and password. Another issue is the scenario where a student logs in with the wrong class code, we will need to ensure that the student cannot join the wrong group by having the teacher approve requests to join.

Dependencies with other requirements:

Practice - Only users that are logged in can use the system.

Manage class - The teacher must login to view their class details.

Join group - Student use the group ID to login to their account.

Create group - Teachers need to login to create a new group.

3.3 Create group (Teacher)

Description:

A teacher will have the ability to have multiple classes/groups within their account. By having this the teacher will be able to assign appropriate work and watch their progress.

Criticality:

This feature will allow a teacher organise and plan all their classes efficiently.

Technical issues:

We need to make sure all group ID's generated are unique, we may need to find a way to accommodate students with the same name in the same class.

Dependencies with other requirements:

Delete class - Need to create a class before it can be deleted.

3.4 Join group (Learner)

Description:

When a student logs in with their username and a group ID this will automatically place them into the correct group and will store this students details in association with that group for future logins.

Criticality:

This feature is very important on behalf of a learner within the system, it allows young children easily be added into their class without having to remember a password and add themselves into a group.

Technical issues:

We need to ensure students cannot accidentally join the wrong group by having the teacher accept join requests.

Dependencies with other requirements:

Practice - Login is required to practice.

3.5 Practice (Learner)

Description:

The user will be able to practice their Irish by constructing correct Irish sentences.

The user will be able to continue where they left off or select a level of difficulty and start new challenges. Each level will have increasingly complex phrases and the easier levels will have colour coding.

Users will be able to drag and drop the words from the dictionary section into the editor linking the words like a jigsaw. If an incorrect word is drag the sentence will reject it throwing it back into the dictionary, we also hope to throw an error when a contextually incorrect words are used and give this feedback to the user and to teachers.

Criticality:

This is the fundamental function of the application, everything else is based around the idea of being able to practice Irish grammar.

Technical issues:

Making the different words fit together like a jigsaw will be challenge, the main problem we will face is contextual situations where the word fits the sentence but the sentence as a whole is nonsensical. We also need to construct our database tables carefully so we can provide appropriate word options to suit different difficulty levels.

Dependencies with other requirements:

All features are dependant on this feature as the system is based around it.

3.6 Manage Class (Teacher)

Description:

The teacher will be presented with statistics and graphs to illustrate their students progress, we would also like to implement a feature that highlights common mistakes made in the class which will alert the teacher to areas they should revise or cover in more detail.

Criticality:

Without this feature the "Teacher" account type isn't very useful at all. This feature is very important to help get the most out of the application as a whole and help teachers really utilise it as part of their classes.

Technical issues:

Recording, storing and transforming the data gathered into accurate statistics may be difficult.

Dependencies with other requirements:

Login - Dependent upon students being logged into the correct group.
Practice - This feature needs the data generated by students practicing.

3.7 Delete Account / Class (Teacher)

Description:

These functions allow a users to delete their account and a teacher to delete a class group.

Criticality:

They are not very important but would help to preserve space in the database, if we have the time we can implement an automated feature to clear out any accounts that have been inactive for a certain amount of time (e.g. one year of inactivity).

Technical issues:

The main consequence here is how often there are dependent records in other tables whose referential integrity is lost when the parent record is deleted. We are not certain if this will be a major problem for us as of yet.

Dependencies with other requirements:

Sign Up / Create group - Else there is nothing to delete.

3.8 Tour Website

Description:

This feature may prove very helpful when catering to a younger demographic in particular, it will guide first time users through the operation of the application and explain any elements that may be ambiguous. We will need unique tours for different account types.

Criticality:

This feature will be very important for teaching the children to use the application, the UI will be very simple so it may not be as necessary for adults, regardless it will be present for all to use.

Technical issues:

We have not identified any issues with implementing this feature

Dependencies with other requirements:

Sign Up - The option to take the tour will be the first thing a new user of the application will see.

3.9 Log-out

Description:

Simply allows a user to end their session.

Criticality:

Without the logout feature, the login feature would be redundant.

Technical issues:

We need to make sure users are logged out properly so the next person to access the site from that browser doesn't somehow access the first person's account.

Dependencies with other requirements:

Login - A session needs to be active to be able to log out of it.

4. System Architecture

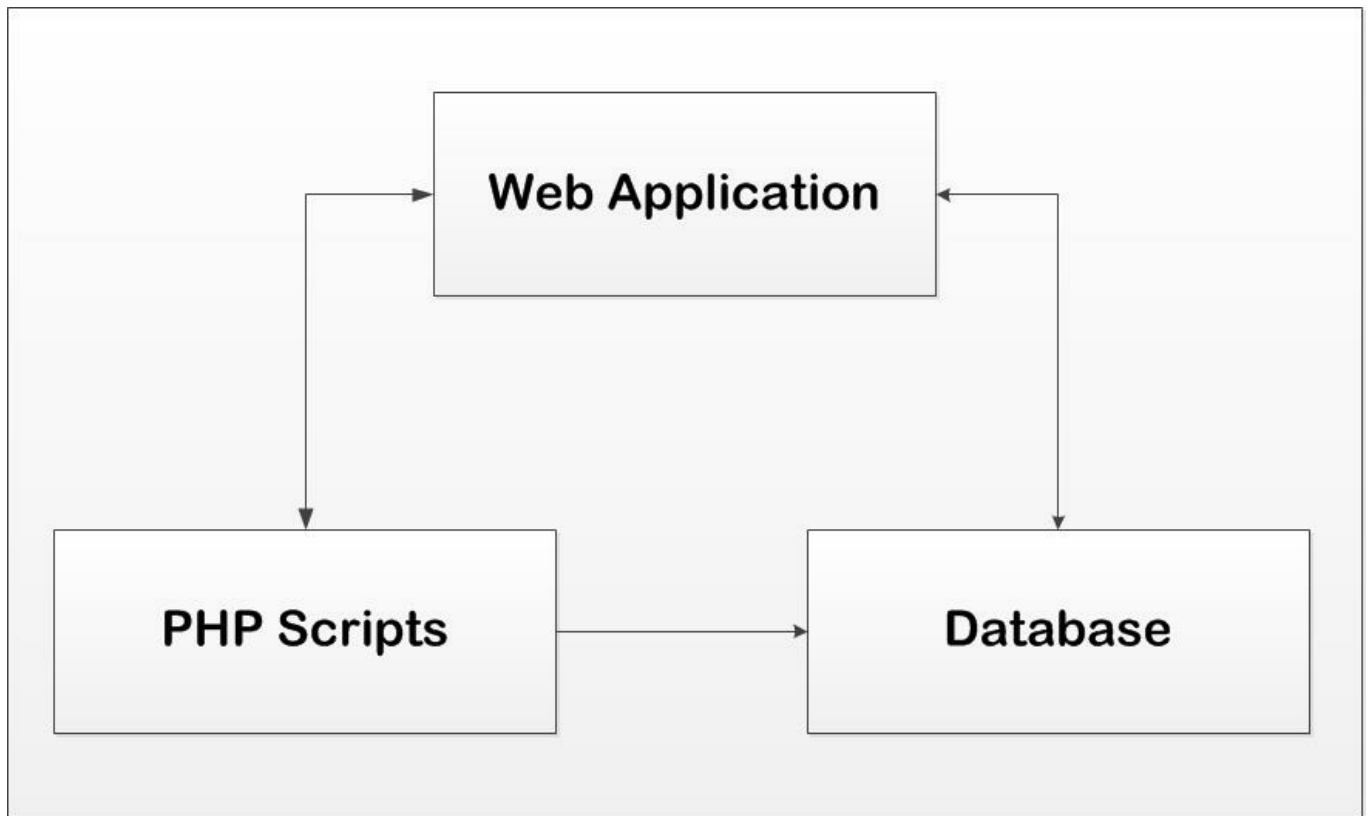
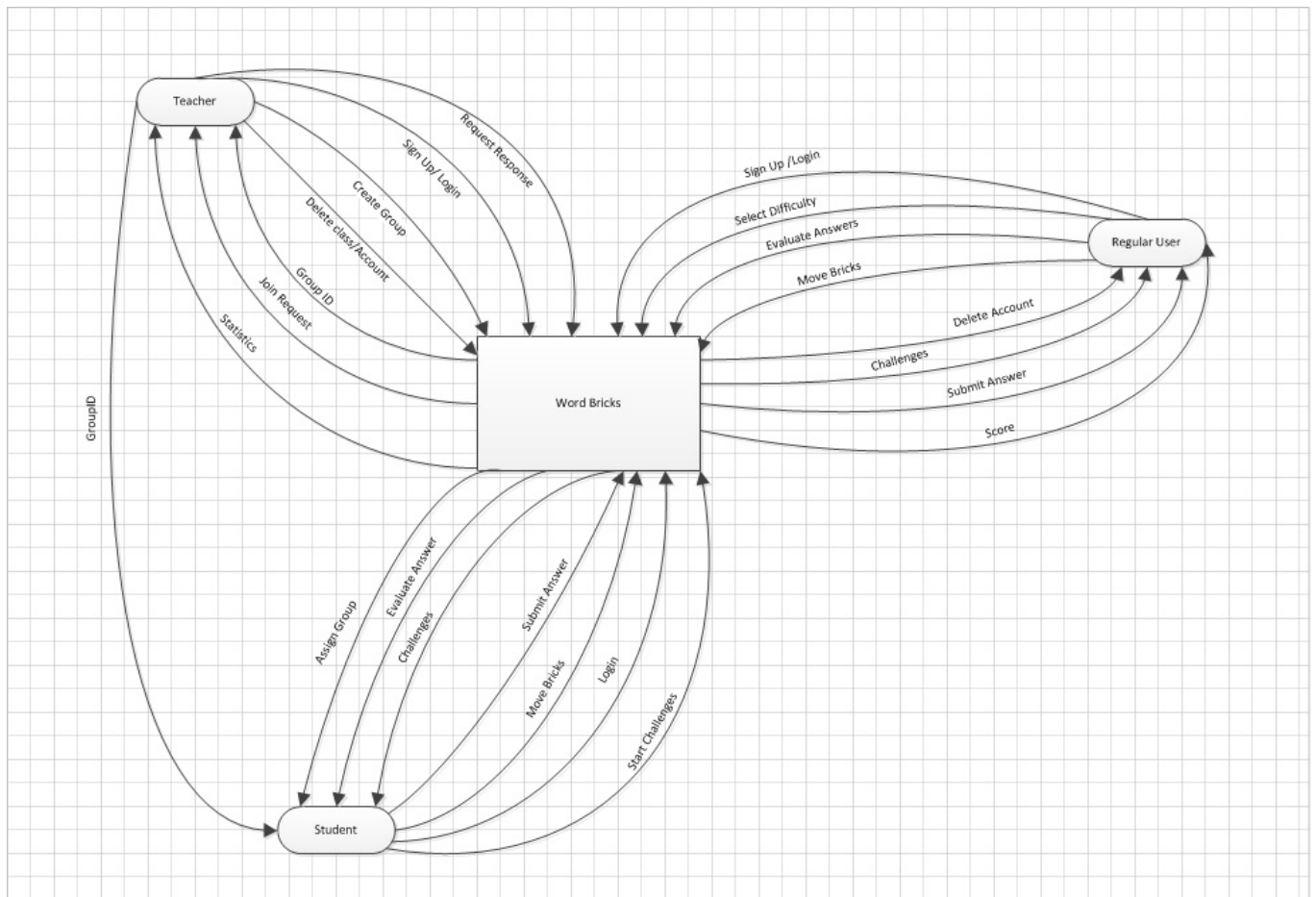


Fig 4.1 - Simple illustration of system

The first component of the system is the Web Application (front-end. What the users see), then the PHP scripts (middle-tier. query the database) and finally the database (back-end. stores the site content).

5. High-Level Design

5.1 Context diagram



5.2 Data Flow Diagram

