

Language Learning Website

6001PROJ Final Year Project 2017 – 2018

Project Supervisor – Denis Reilly

Project Author – Nathan Stanley

Student Number – 710768

Course – Computer Studies

Submission Date – 20 / 4 / 18

Table of Contents

Table of Contents.....	2
Abstract.....	5
Acknowledgements.....	6
List of Illustrations and Tables.....	7
Acronym Guide	9
Chapter 1. Introduction.....	10
1.1 Reasons Why	10
1.2 Project Goals.....	11
1.3 Project Scope	11
1.4 Roadmap.....	11
Chapter 2. Project Background	12
Chapter 3. Literature Review.....	14
3.1 Benefits and Importance of MFL in the Professional World.....	14
3.2 Benefits and Importance of MFL in the Social World	15
3.3 Arguments Against Learning a Language	17
3.4 Examination of Existing Systems.....	18
3.4.1 Duolingo.....	18
3.4.2 Memrise	21
3.4.3 Busuu.....	24
3.5 Regarding Usability	25
3.6 Literature Review Conclusion	26
Chapter 4. Problem Analysis.....	27
4.1 What is Being Tested?.....	27
4.2 The Problem.....	27
4.3 How StanLearn Intends to do This.....	28
4.4 StanLearns Goals.....	28
Chapter 5. Methodology.....	28
5.1 Prototyping Method	28
5.2 Waterfall Method	30
5.3 Selected Method and Justification	32
5.4 Method of Requirement Obtainment	32
Chapter 6. Requirements and the User.....	32
6.1 What Do Users Need?	32
6.2 User Requirements Questionnaire.....	32
6.2.1 Questions Included	32
6.2.2 Questionnaire Results.....	35

6.3 Functional Requirements	44
6.4 Non-Functional Requirements	44
6.5 Personas	44
6.5.1 Young Teen / Child Persona	44
6.5.2 Teen Persona	45
6.5.3 Adult Persona	45
6.5.4 Senior Persona	45
6.6 Use Case UML Diagram.....	46
Chapter 7. Project Design	46
7.1 Site Mapping	46
7.2 Site Storyboarding	47
7.2.1 Homepage	48
7.2.2 About	49
7.2.3 Contact	50
7.2.4 Swedish	53
7.2.5 French.....	54
7.2.6 Spanish.....	55
7.2.7 German.....	57
7.2.8 Quiz Pages	58
7.2.9 Word Lists.....	59
7.3 Activity UML Diagram	60
Chapter 8. Development of the Language Learning Quiz.....	61
8.1 How Will the Quiz Work?	61
8.2 HTML Base	63
8.3 CSS Styling	65
8.4 Java.....	67
8.4.1 Java Class Diagram	70
Chapter 9. Implementation	70
9.1 Consideration of Software	70
9.1.1 Notepad++	71
9.1.2 Sublime Text 2	72
9.2 Chosen Software	74
9.3 Website Creation	75
9.3.1 Homepage	75
9.3.2 About Page	77
9.3.3 Contact Page	80
9.3.4 Language Introduction / Entry Pages	83

9.3.5 Language Quiz Pages.....	92
9.3.6 Word List Pages.....	99
Chapter 10. Testing	104
10.1 Test Cases	104
Chapter 11. Evaluation.....	106
11.1 Site Aesthetics and Usability.....	106
11.2 Site Functions and Purpose Fulfilment	108
11.2.1 Strengths.....	108
11.2.2 Weaknesses	108
11.3 Significance and What Areas Could be Further Explored	108
11.4 Evaluation Methods	109
Chapter 12. Conclusions and Further Work	109
Chapter 13. Project Management	110
13.1 November Meeting	110
13.2 December Meeting	111
13.3 January Meeting	112
13.4 February Meeting	113
13.5 March Meeting.....	114
References	115
Bibliography.....	116
Appendices	116

Abstract

The purpose of this report is to document and explain the process behind the creation of StanLearn, a language learning site that offers users the ability to study the basics of Swedish, French, Spanish and German (European modern foreign languages) by the process of splitting each language into four basic topics – animals, numbers, colours and food – and testing users on how well they remember these newly learned words by creating wordlists with which to first view, then test their memorisation skills on.

Acknowledgements

Thank you to my family for creating me and by extension this paper, thank you to my friends for the endless support, who without I would surely not have made it this far, and thank you to Denis for supervising my project.

List of Illustrations and Tables

Figure 1 – A table of languages ordered by number of native speakers.....	16
Figure 2 – An extension to the table of languages ordered by number of native speakers..	16
Figure 3 – Most natively spoken languages worldwide, including geographical location.....	17
Figure 4 – Exploration into Duolingo's language learning process.....	19
Figure 5 – Further exploration into Duolingo's language learning process.....	19
Figure 6 - Further exploration into Duolingo's language learning process.	20
Figure 7 - Further exploration into Duolingo's language learning process.	20
Figure 8 - Further exploration into Duolingo's language learning process.	20
Figure 9 - Exploration into Memrise's language learning process.....	21
Figure 10 - Further exploration into Memrises' language learning process.....	21
Figure 11 - Further exploration into Memrises' language learning process.....	21
Figure 12 - Further exploration into Memrises' language learning process.....	22
Figure 13 - Further exploration into Memrises' language learning process.....	22
Figure 14 - Further exploration into Memrises' language learning process, regarding sentence choice.....	23
Figure 15 - Further exploration into Memrises' language learning process, regarding sentence choice.....	23
Figure 16 - Exploration into Busuu's language learning process.	24
Figure 17- Exploration into Busuu's language learning process.	24
Figure 18 - Diagram explaining the prototype methodology.	29
Figure 19 - Diagram explaining the waterfall method(ology).....	30
Figure 20 - Questionnaire made to gather user requirements.	33
Figure 21 - Questionnaire made to gather user requirements.	33
Figure 22 - Questionnaire made to gather user requirements.	34
Figure 23 - Questionnaire made to gather user requirements.	34
Figure 24 - Questionnaire made to gather user requirements.	34
Figure 25 - Questionnaire made to gather user requirements.	35
Figure 26 - Questionnaire made to gather user requirements.	35
Figure 27 - Results obtained from responses to the user requirements questionnaire.....	36
Figure 28 - Results obtained from responses to the user requirements questionnaire.....	36
Figure 29 - Bar charts of results obtained from responses to the user requirements questionnaire.	37
Figure 30 - Fixed bar charts of results obtained from responses to the user requirements questionnaire.	37
Figure 31 - Fixed bar charts of results obtained from responses to the user requirements questionnaire.	38
Figure 32 - Bar charts of results obtained from responses to the user requirements questionnaire.	39
Figure 33 - Results obtained from responses to the user requirements questionnaire.....	40
Figure 34 - Pie chart of results obtained from responses to the user requirements questionnaire.	40
Figure 35 - Pie charts of results obtained from responses to the user requirements questionnaire.	41
Figure 36 - Pie charts of results obtained from responses to the user requirements questionnaire.	42
Figure 37 - Results obtained from responses to the user requirements questionnaire.....	43
Figure 38 - Fixed bar charts of results obtained from responses to the user requirements questionnaire.	43

Figure 39 – Pie chart of results obtained from responses to the user requirements questionnaire.	44
Figure 40 - Diagram depicting use cases.	46
Figure 41 - Site mapping diagram.	47
Figure 42 - Visio created object depicting site early designs.	48
Figure 43 - Image of site header.	49
Figure 44 - Visio created object depicting site early designs.	49
Figure 45 - Visio created object depicting site early designs.	50
Figure 46 - Image used in the site as the StanLearn "office location".	52
Figure 47 - Visio created object depicting site early designs.	53
Figure 48 - Visio created object depicting site early designs.	54
Figure 49 - Visio created object depicting site early designs.	55
Figure 50 - Visio created object depicting site early designs.	57
Figure 51 - Visio created object depicting site early designs.	58
Figure 52 - Visio created object depicting site early designs.	59
Figure 53 - UML activity diagram.....	61
Figure 54 - Diagram depicting how the site will display user results.	62
Figure 55 - Visio object depicting final design of quiz pages.....	63
Figure 56 - Code snippet depicting the header and intro to the language learning page for Swedish.	63
Figure 57 - Code snippet depicting the question in the Swedish animals quiz.	64
Figure 58 - Image depicting the submit button.....	65
Figure 59 - Code snippet depicting correct answers that display upon submitting answers.	65
Figure 60 - CSS code snippet.	66
Figure 61 - CSS code snippet.	66
Figure 62 - CSS code snippet.	66
Figure 63 - CSS code snippet.	67
Figure 64 - Java code snippet.	67
Figure 65 - Table depicting how the Java reads the questions in the HTML.....	68
Figure 66 - Java code snippet.	68
Figure 67 - Java code snippet.	69
Figure 68 - Java code snippet.	69
Figure 69 – UML Class Diagram.	70
Figure 70 - Screenshot of Notepad++	71
Figure 71 - Screenshot showing functionality of Sublime Text 2.....	73
Figure 72 - Screenshot showing functionality of Sublime Text 2.....	73
Figure 73 - HTML snippet of final code used in site.....	75
Figure 74 - CSS snippet of final code used in site.	76
Figure 75 - Homepage in Google Chrome.....	77
Figure 76 - Homepage in Microsoft Edge	77
Figure 77 - HTML snippet of final code used in site.....	78
Figure 78 - CSS snippet of final code used in site.	79
Figure 79 - About page in Google Chrome.....	80
Figure 80 - About page in Microsoft Edge.	80
Figure 81 - HTML snippet of final code used in site.....	81
Figure 82 - CSS snippet of final code used in site.	82
Figure 83 - Contact page in Google Chrome.	83
Figure 84 - Contact page in Microsoft Edge	83
Figure 85 - Swedish language area banner graphic.	84
Figure 86 - HTML snippet of final code used in site.....	84
Figure 87 - HTML snippet of final code used in site.....	85

Figure 88 - HTML snippet of final code used in site.....	85
Figure 89 - HTML snippet of final code used in site.....	86
Figure 90 - CSS snippet of final code used in site.	87
Figure 91 - CSS snippet of final code used in site.	88
Figure 92 - CSS snippet of final code used in site.	89
Figure 93 - CSS snippet of final code used in site.	90
Figure 94 - Swedish language area in Google Chrome.	91
Figure 95 - French language area in Google Chrome.	91
Figure 96 - Spanish language area in Google Chrome.	92
Figure 97 - German language area in Google Chrome.	92
Figure 98 - Code snippet of HTML used in final site.	93
Figure 99 - Code snippet of HTML used in final site.	93
Figure 100 - Code snippet of HTML used in final site for quiz questions.	94
Figure 101 - Code snippet of HTML used in final site for submit button.	95
Figure 102 - Code snippet of HTML used in final site for question answers.	95
Figure 103 - Code snippet of HTML used in final site to return user to different pages.....	95
Figure 104 - Code snippet of CSS used in final site.	96
Figure 105 - Code snippet of CSS used in final site.	96
Figure 106 - Code snippet of HTML used in final site.	97
Figure 107 - Submit button not highlighted in Google Chrome.	98
Figure 108 - Submit button highlighted in Google Chrome.	98
Figure 109 - Submit button not highlighted in Microsoft Edge.	99
Figure 110 - Submit button highlighted in Microsoft Edge.	99
Figure 111 - HTML depicting wordlists with Swedish words and English translations.....	100
Figure 112 - Code snippet relating to table styling.....	101
Figure 113 - CSS code used in final site.	101
Figure 114 - CSS code used to style table headers, rows and columns.	102
Figure 115 – Swedish wordlist in Google Chrome.	102
Figure 116 - French wordlist in Google Chrome.	103
Figure 117 - Spanish wordlist in Google Chrome.	103
Figure 118 - German wordlist in Google Chrome.	104
Figure 119 - Table of use cases.	106
Figure 120 - Swedish language area graphic.	107
Figure 121 - French language area graphic.	107
Figure 122 - Spanish language area graphic.	107
Figure 123 - German language area graphic.	107

Acronym Guide

NDA – No Date Available. (used in referencing when referring to a missing or undiscoverable date).

MFL – Modern Foreign Language(s).

LLW – Language Learning Website (my website project, also known as StanLearn).

ST2 – Sublime Text 2

N++ – Notepad++

Chapter 1. Introduction

The function of this report is to inform the reader of the reasoning behind the creation of the language learning website. This report will inform the reader of the websites reason for existence and its purpose, and justify the creation of it by informing the reader which problems it tackles in society, as well as what kind of people it has been created for. The created website is an interactive website for use by people in the age range of 15-50+ (broadly, for young teens and older members of the population). The reason for this broad age range is to accommodate users of all levels of computing ability, and to help push the notion that anyone can decide to learn a language, and at any age.

The current language of the site is English, and the languages selected to be learned are all modern foreign languages in Europe. The languages chosen for the website are French, Swedish, X and Y. The reasoning behind this choice is that these are useful modern foreign languages, which are spoken widely in Europe, with some of these spoken worldwide. The most useful on the site may be French, as it is considered by most European countries to be the most important language to learn second only to English [\[Statista,2015\]](#). Among the other included languages is Swedish. I felt it important to include Swedish as it is what is known as a “North-Germanic” language, and shares similarities between Scandinavian countries such as Norway, Denmark, Iceland and more. Therefore, users that choose to learn Swedish can also advance onto learning other North-Germanic languages with relative ease. [\[Claus Skovbjerg,2014\]](#)

In addition to Swedish and French, I also found it appropriate to keep with the theme of European MFLs and include German and Spanish. The decision to include Spanish and German was done based on a study [\[European Commission, 2012\]](#). In the study, Europeans were asked which two languages were the most important for personal development. It was found that most Europeans find that English is the most useful language with 67% (but English will not be included in StanLearn), seconded by German at 17%, French at 16% and Spanish at 14%, with Chinese trailing behind with 6%. Because of this, and the fact that most Europeans will now be looking to actively study those languages, I have decided German, Spanish, French and Swedish will be the language courses my site offers.

Knowledge of other modern foreign languages – especially European languages – can greatly boost the chance of employment for many people. If language no longer presents a barrier to a person, they are open to employment opportunities not only from their own country, but from other countries as well. Knowing another foreign language makes you more attractive to employers, and proves to them that you are committed to learning and bettering yourself in your free time, outside of your work environment [\[Rachel Wagers,NDA\]](#). It has also been found that knowing a foreign language increases skills in decision making [\[University of Chicago, 2017\]](#), and can help to lower the risk of Alzheimer’s and dementia [\[M. Perquin, 2013\]](#).

1.1 Reasons Why

The reasoning behind the project comes from personal experience of language learning, and experience using those languages in foreign countries to communicate with people. I believe that knowing any MFL greatly increases not only a person’s chances to become meaningfully employed, but also to broaden their outlook on the world, and help them to communicate meaningfully with other people across the world, and helps them to gain mutual understanding. Some of this personal experience stems from a trip to Paris, France I undertook five years ago in which I tested my own abilities in the French language, and found deep satisfaction in communicating with locals and shop owners. After this experience, I also undertook learning Swedish, which has not only helped me to design my

final year project website, but has made clear to me that knowing a MFL can broaden a person's mind and provide them with significant advantages over others in both employment and social environments.

By creating this website, I can assist people of all ages to learn a language and develop this same mutual understanding I experienced years ago and still experience to this day. By creating the website for people of all ages, I can specifically provide children the opportunity to learn the skills I learned, except from a much younger age, in order to increase their cognitive abilities and possibly make them more socialised.

1.2 Project Goals

This subsection will aim to detail the goals of the project, and may also include whether or not they have been achieved. It is important to record what the goals are at the start of the project, so it can be mentioned if they were achieved when viewing the later sections, particularly those referencing user requirements, and what kind of users the site is to be built for.

The main goal of the language learning website project is to provide an effective platform for young teens, students, adults and senior adults to become proficient and eventually fluent in one or more MFLs.

The project also has secondary goals. These being to provide the user with quantifiable feedback on their performance in the quiz. As this project is only a prototype, this will be in the form of "you got X questions correct! The correct answer to Y question is: Z", etc. In an expanded site design, the user could be graded, or potentially given a progress bar.

1.3 Project Scope

While creating the project, it is important to remember its advantages and its drawbacks, so that these problems can be noted and addressed in the project's later life. First, this project will only be a prototype, and may be a very barebones prototype in and of itself. This means that while it may have an impact, that impact could be drastically reduced by the fact that the site is not fully finished and has a severe lack of complexity, and may have stylistic elements missing as well as functional requirements (these will be detailed in the functional requirements section).

For the project to be as effective as desired, and to produce its full impact on the MFL learning market, it must be developed to contain many non-functional requirements (could possibly involve personal tracker, level system etc.) that helps to maintain its non-essential quality of life. In doing this, the project can potentially be used in housing for senior citizens, or perhaps used in schools to teach young children.

In terms of keeping the product updated – languages do not change much. While some obscure languages or languages that have not been fully translated may require updates to the website and quiz so that the wrong things are not being taught, this is not a concern when discussing European MFL, as they are very well developed, documented and current language translations used in the website are used in schools and other MFL teaching apps/sites.

1.4 Roadmap

This section is designed to detail the structure of the report, and will also explain why it has been done this way. The chapters in this report will be listed almost chronologically – meaning the chapters and the format they are in are roughly representative of the order they were completed.

First is chapter 2 which details the projects Background. This short chapter builds in small part from chapter 1.1, and describes who the project has been created for, and why I feel its creation was necessary, with regards to personal experiences as well as evidence and sourced claims backing up its reason for existing.

Chapter 3 is the Literature Review, which is an assessment of work already done in my area, as well as the implications that my project can have on a person's social and professional life, as well as the benefits it can bring to them as a person. It will detail and explain the importance of learning a language and why it is almost a necessity in a globalised world.

Chapter 4 is the Problem Analysis, where the problem that StanLearn aims to tackle is detailed specifically.

Chapter 5 details the Methodologies that could've been chosen to structure this project, and which one was selected to be the final methodology the project would follow. Towards the end, it explains why it is important to gather requirements first, and therefore why chapter 6 is a necessity.

Next is the Requirements in chapter 6, which details what potential users may require, and how the site can be developed to accommodate as many of those users and their personas as possible. This section also details exactly what the site will require functionally speaking, as well as requirements that are not necessary for the site to function, but are necessary for users. It also contains the questionnaire that was used to gather these results.

Chapter 7 is the Project Design, which contains the structure the site follows in the form of the site map, as well as storyboards that explain the design process behind each page.

The next chapter, 8, will concern the development of the quiz that is used in the site, including how it functions and how it is integrated with the other elements of the site.

After this chapter 9 will be focused on Implementation. This means the implementation of the functionality detailed in chapter 8, into each of the webpages. It will also inform about how each page of the website was created, along with which software was considered for the task of creating the site, as well as the software that was used in the end.

Next is chapter 10, which regards testing. Testing will focus on user testing and self-testing, and features a list of test cases as well.

Succeeding this is the Evaluation section. The evaluation will seek to provide the reader with a concise rundown of what has been learned in the process of creating this website, as well as measuring how successful the site has been in achieving its goal. It is also an assessment of where the project has excelled, as well as where its weak spots are. This is done using testing methods that are detailed in the section.

After this is the Conclusions that can be drawn, and the Further Work that can be touched upon. This is where the effectiveness of the site overall is quantified, and explains how it has affected the problem area, as well as how the site itself has been affected by it.

The final chapter is Project Management, which details how effectively I managed my time during the project from its beginning to its end.

Chapter 2. Project Background

This short chapter will detail who this project is for, why it is for them and why it is necessary.

This project is not aimed at any one particular person. In fact, it is not aimed at any one particular group (unless “people who want to learn a language” is a group), it is aimed to reach out and draw in as many different people as possible. This includes people of different ages, different cultural backgrounds, different computing ability. Currently, there are multiple different sites and apps that offer the ability to learn a language. Not only are these successful, they are also widely used with Duolingo alone boasting over 120 million *active* users [[Duolingo, 2016](#)], with other free language learning resources existing to serve the same purpose.

However, since each of these language learning resources are all free (and as discovered in chapter 3 subsection 3.4 below), they are missing features that would otherwise be locked behind a paywall, namely unless the user has a “premium” account they will not be able to fully unlock the extent of the features offered by these different services, and the paid alternatives are pricey and catered to a specific audience. They are complex, expensive, and can be exclusionary based on computing ability. This is where I feel my Language Learning Project comes into play. While in its current state it will only be a prototype, it will aim to offer the best service possible for all users (including those of entry-level ability) to begin learning a new MFL.

Since the desire to learn a new language can occur at almost any age, it is important that sites offer the chance to learn to anyone. However, this is not always easy, as different ages may require different site designs, as well as people who are of poor computing ability or who may have audio / visual impairments. I feel it is important to show people the benefits that come with learning a new language, not just the ability to communicate with new people globally, but the other social and even professional benefits it brings. For example, learning and knowing two languages has been proven to increase cognitive ability and enhance decision making [[University of Chicago, 2017](#)], as well as decrease the risk of dementia [[M.Perquin, 2013](#)].

Currently, the way languages are taught to young people (more specifically, at secondary school level) does not seem adequate, and in fact seems to be failing to properly teach people in certain areas about why languages are learned, and how they are useful. For example, in the United Kingdom, 62% of people “could not speak any other foreign language”, which is an overwhelming majority when concerning its surrounding area of Europe, where bilingualism is the norm and linguistic skills can potentially expand into knowing 3 or more languages. Schools do not use language learning applications or sites very often to teach their pupils, instead it is mostly done with paper tests and audio recordings. A way they can currently start using sites more efficiently would be to use an existing free service such as Duolingo, which has millions of active users return to learn and test the skills they have learned. While other, better alternatives do exist, they are often paid such as Rosetta Stone, which charges monthly but can be paid in bulks of 6 months or 1 year, however these can be expensive to purchase and maintain, especially for a large number of students.

I feel my LLW (named StanLearn) can help to teach people new modern foreign languages at any ability level, and more importantly with the intention of starting people off learning these languages at a young age, so they can fully grasp the benefits of being able to read, write, listen to and communicate with a learned MFL. As the world is becoming increasingly more used to applications and sites, particularly as learning resources, it may become easier to introduce sites and applications as the new learning norm, or at least may serve in convincing people that this can be the case. Especially in the case of StanLearn, where new

questions can easily be added, answers and questions can be changed, and the site layout will still stay free of restriction based on age, language ability or computing ability.

Chapter 3. Literature Review

This literature review will review the current work that has been carried out in this field, and will aim to explain the problems in the field itself. It will also examine and discuss existing controversies and explain why they are present. This literature review will explain and develop upon the reasons that being able to communicate with other people in one or more MFLs is an important step to be taken, and why it is a reasonable and beneficial choice for people everywhere to make. It will explain why certain other learning mediums that already exist in the field are successful, and where my project will aim to emulate them.

3.1 Benefits and Importance of MFL in the Professional World

First, it is important to understand why people already learn MFLs in the first place. This subsection along with subsection 3.2 will aim to point out the usefulness of knowing one or more MFLs in different situations. This subsection, 3.1, will explain why knowledge of one or more MFLs is beneficial in a professional environment.

Being bilingual, trilingual or multilingual can have a huge impact on a person's career prospects. While it may be easy to dismiss this skill as unimportant in some lines of work, as it may not be a needed or required skill in any shape or form, it is more about the professional implications that learning a MFL carries. For example, if there are two candidates that both have the same credentials, except one of the candidates knows two or three languages, the candidate who knows the language is more likely to acquire the position. Why? Because it proves that they make use of their personal time to improve themselves, rather than sit around or use it on other activities that do not necessarily lead to them gaining new skills or experiences, which some companies often look for. Knowing one or more MFLs makes you much more marketable, and therefore makes you a much more attractive candidate in the long run. For companies that operate in one or more countries, or operate inside diverse areas such as, for example, some southern states of North America where the number of Spanish speakers exceeds 37 million [[Pew Research, 2013](#)] knowing Spanish can be almost invaluable.

High-level companies may also be interested in candidates that know MFLs. For some companies who desire to operate, expand or connect globally to other companies abroad for the purposes of merging will look for candidates that are knowledgeable in one or more MFLs to improve the company's image, and to attain the best chance possible of securing a deal with a foreign company. As the world becomes more globalised and connected, dealing with companies from foreign countries becomes less and less of a peculiarity, and more of a normality in the modern world. For example, the employment numbers for people in the translation and interpretation industry are projected to grow 18 percent from 2016 to 2026 [[U.S.B.L.S, 2018](#)], which is much faster than the average for *all* occupations, citing that the demand will remain for modern foreign languages and will not slow down, and also states that there will be an increase in demand for languages such as Arabic and "other middle eastern languages" (this will likely include languages such as Hebrew with 7 million native speakers [[Students Abroad, 2013](#)] and Turkish with over 70 million native speakers [[EU Commission, 2012](#)]). The article also says the demand "should be strong" for languages such as Chinese, Japanese and Korean, and also mentions emerging demand for native Mexican languages such as Zapotec, Mixtec and Mayan dialects. Despite all of these languages mentioned however, the article says that the "demand will likely remain strong" for European languages such as French, German, Portuguese, Russian and Spanish.

Moreover, in knowing one or more modern foreign languages the opportunity arises for a candidate to find employment outside of their own country, instead of simply being more attractive for positions in their native country. For example, the translation and interpretation industry in Germany is set to pose strong job growth along with its economy [\[Slator, Statista, 2016\]](#), inferring that while translation jobs can provide employment to non-natives and allow for MFL-speaking candidates to find employment abroad, they may also be able to earn more as the foreign countries economy they are employed in grows.

Finally, in candidates can take advantage of the free movement through Schengen areas in the European Union and assess offers from multiple different companies in different countries. For example, in industries such as sales and marketing in Europe, knowing MFLs can add between 10% and 15% to a wage [\[Guardian, 2010\]](#), and could make the difference between them doing business in only western Europe, or branching out to countries such as Ukraine and Kazakhstan asking for Russian to English interpreters, where the English-speaking population may be low or in decline, or just not of adequate translation quality.

In conclusion, knowing one or more modern foreign languages can undoubtedly help a candidate to not only be the first choice for businesses based on their use of time to actively improve themselves, but also to help them acquire work they may find more interesting such as being suggested for handling translations for the company they work for. Even if the learner cannot communicate vocally with these foreign contacts, there may be work that requires them to only translate textual documents for said company.

3.2 Benefits and Importance of MFL in the Social World

This subsection will aim to point out the usefulness of knowing one or more MFLs in a social environment. An important distinction to make is what qualifies as a modern foreign language. Professionally speaking, a modern foreign language is one that is used for day-to-day communication, not one that has little known speakers or is a “dead” language, meaning it has no speakers or no use in the modern world. While some would consider the most useful languages to learn to be ones that would assist in the business world (European languages such as French, English, German etc.) this section will detail *worldwide* modern foreign languages, and the social implications that possessing knowledge on them brings.

As the world grows more and more globalised, people across the world become interconnected and have the opportunity to communicate where they would not before. This allows us to become more exposed to languages and the way they are spoken *casually* in the world today, and the importance of knowing these languages grows. For example, in the EU there are 23 officially recognised languages, with approximately two thirds of working age adults knowing at least one foreign language [\[Europa, 2011\]](#). This has broad implications, as it means that there is now a much larger percentage of people who can speak a foreign language conversationally, and opens up the opportunity to converse with many more people who perhaps do not know your native language, but you know theirs. This means that people all across Europe at least, now have the opportunity to speak to potentially millions of people who they could not before, increasing the likelihood of making new friends and partaking in new social experiences. Furthermore, it increases the likelihood of the person who knows the modern foreign language of learning things they might not have known before, or not been able to understand. By knowing a language, you are by proxy exposed and perhaps introduced to that languages’ culture, and by knowing and learning a new culture you are more likely to build meaningful relationships with new people who belong to that culture. By becoming exposed to these new people and their cultural tenets, there is the possibility of the learner becoming more tolerant, accepting or at least much more understanding of the culture and the strengths it possesses as well as its

shortcomings. Continuing from the cultural aspect, there is perhaps a much broader point to be made about understanding and accepting foreign cultures, and this is the possibility and inference that knowing one or more modern foreign languages can reduce the possibility of a cultural clash and perhaps avert a person from forming cultural or perhaps racial biases based on preconceptions they may have had before knowing that language. There is the possibility as well that between countries who may be geo-political rivals (for example, India and Pakistan) that having people from each country know the language of the other and have them interact online or in person, there is the possibility for tensions to be defused, at least between the populace.

However, while Europe's dense population and concentration of languages is an important point, globalisation has highlighted the importance of knowing languages globally. This is because while the business language of the world may be English, and while some people may consider European languages to be the most important, there are emerging social opportunities with people around the world (for example cultural exchange programs, university exchange programs, immigration, global travel), and studies have shown that European languages are not the most widely spoken by far. In fact, English is only the third (3rd) most natively spoken language in the world, with Chinese (and its dialects) eclipsing English's 378 million speakers with its 1.2 billion native speakers [[Ethnologue, 2018](#)], with Spanish behind it at 442 million native speakers. Other languages in the list are;

Position	Language	Native Speakers (in millions)
4 th	Arabic (and its dialects)	315
5 th	Hindi	260
6 th	Bengali	243
7 th	Portuguese	223
8 th	Russian	154
9 th	Japanese	128
10 th	Lahnda (and its dialects)	119

Figure 1 – A table of languages ordered by number of native speakers.

[[Ethnologue, 2018](#)]

While these languages numbers are impressive, it is important to note that of the ten shown on this list, only 4 are European languages. Of the next 10 languages in the same table, just 2 of those are European (discounting Turkish in favour of it being considered Middle-Eastern).

Position	Language	Native Speakers (in millions)
11 th	Javanese	84.4
12 th	Turkish	78.5
13 th	Korean	77.2
14 th	French	76.8
15 th	German	76
16 th	Telugu	74.8
17 th	Marathi	71.8
18 th	Urdu	69.2
19 th	Vietnamese	68
20 th	Tamil	66.7

Figure 2 – An extension to the table of languages ordered by number of native speakers.

[[Ethnologue, 2018](#)]

Language	Geographical Location
Chinese	Asia
Spanish	Europe
English	Europe
Arabic	Middle East
Hindi	Asia
Bengali	Asia
Portuguese	Europe
Russian	Europe
Japanese	Asia
Lahnda	Asia
Javanese	Asia
Turkish	Europe / Middle East (can be subjective)
Korean	Asia
French	Europe
German	Europe
Telugu	Asia
Marathi	Asia
Urdu	Asia / Middle East (can be subjective)
Vietnamese	Asia
Tamil	Asia

Figure 3 – Most natively spoken languages worldwide, including geographical location.

These tables highlight the importance of extending the label of modern foreign languages to Asia and the Middle East, as while some of these languages names may be unknown to some, they are some of the most spoken, and some among the most rapidly growing. By learning some or more the more obscure languages in this list (particularly the lesser-known Asian languages), there is the potential for these languages to become more well known worldwide and increase the likelihood of a more globalised, connected population.

In addition to this, it was found that the number of users currently learning the Welsh language in Duolingo (a free website/app, detailed in subsubsection 3.4.1) had surpassed the actual number of people who are able to speak Welsh [[inews, 2018](#)]. This has very broad implications, as it shows that through studying a second language, and by extension language learning websites, can help to bring languages that are declining in number of speakers back into use and perhaps back into popularity. The article states that in a 2011 census, the number of people who said they could speak Welsh was only 562,000 [[Office for National Statistic, 2011](#)] – just over half a million – whereas Duolingo stated they had over a million people learning the language through their software. This helps to back up the implication that people learning a language can help bring it back into social use, and perhaps implies that “dead” languages that have been documented and recorded can be brought back into use if they are included in language learning repositories.

3.3 Arguments Against Learning a Language

This section will present arguments against learning a language. So, while there are arguably multiple benefits to learning a language, there are some who believe it is of little importance to them to learn any MFL at all. For example, in the United Kingdom, while geographically very close to the European continent, has little number of native English-speaking people who can speak another foreign language [[British Council, 2015](#)]. In fact, only 38% of Britons speak a foreign language, with only 18% speaking two and 6% speaking three or more. This is in stark contrast to most of mainland Europe, where it is expected to be able to speak the countries’ native language in addition to English, and perhaps more languages beyond that based on culture and geography. For example, in Luxembourg, the

native language is Luxembourgish, but it is expected to know German and French, as well as English being taught in schools. The argument for not learning or otherwise not wanting to learn another language for native English people specifically, is that because the English language is the lingua franca and spoken *almost* universally, there is little reason for them “needing” to learn a foreign language to communicate, especially if they have no interest in leaving the U.K. or working outside of the U.K among foreigners. Moreover, even if someone from England was to apply to live outside the country or even emigrate elsewhere, the chances of English being widely spoken are still quite high, with the chances of English being the business language also high. This is because it is estimated that in the world, there are around 1.12 billion people who speak English [[Ethnologue, 2018](#)] (includes natives and second-language speakers). This is a considerably large portion of the world population, so the argument is that it is safe to assume that most places in the world will simply speak English.

3.4 Examination of Existing Systems

Before beginning the creation of my website, I began to look at and study other language learning repositories, such as websites and apps. These three that follow were what I decided to be the most informative and most eye-catching MFL teaching services. Some content shown below will be screenshots and writing excerpts taken from my monthly report in the month November (2017). All example screenshots will be in Swedish or French. Some websites also feature user networking, but as this feature is very complex and an area I am not familiar with, it will not be considered for implementation in my prototype, and so will be discounted from discussion in the section(s) below.

3.4.1 Duolingo

Overview

Duolingo is a language learning website and mobile application that launched in 2012 for public use. Its aim is to use audio, visual and textual cues to help people learn languages. The typical process consists of first teaching the user several new words, and then either reading it back to them and asking for an answer, or asking them to select which word is which among a multiple-choice environment. For example, if the user decides they want to learn about clothes, they must first advance through the basics of sentence structure and then they will be allowed to access that section. The lessons will be split into numbers, each containing several words the user must learn before they can advance to the next lesson. A small yellow tick helps the user to distinguish between lessons they have completed, and lessons they have not.

The screenshot shows the Duolingo mobile application interface. At the top, there is a red header bar with a white icon of a sock on the left, followed by the text "Clothing" and "2/3 Lessons completed". On the right side of the header is a white button with the text "TEST OUT". Below the header, there are three cards representing lessons:

- Lesson 1 of 3**: Contains the text "sko, på, kläning, jacka, kjolarna, tröjor, sig" and a yellow ribbon icon with a checkmark. Below it is a red "REDO" button.
- Lesson 2 of 3**: Contains the text "strumpha, byxorna, kläderna, kostymerna, hattarna, skjortor" and a yellow ribbon icon with a checkmark. Below it is a red "REDO" button.
- Lesson 3 of 3**: Contains the text "skärp, handske, slipsen, ficka, knapp, halsduk, rocken" and a yellow ribbon icon with a checkmark. Below it is a red "START" button.

Below these cards, there is a section titled "Tips and notes" with the heading "Wearing clothes". The text explains the reflexive particle verb "har på sig" and provides examples for each person pronoun:

jag har på mig
du har på dig
han/hon har på sig
vi har på oss
ni har på er
de har på sig

Figure 4 – Exploration into Duolingo’s language learning process.

In the first lesson, the user will be shown the English word, 3 pictures with the foreign word beneath them, and are asked to select the correct answer.

Select the word for “a skirt”

The screenshot shows a vocabulary matching exercise. The title is "Select the word for ‘a skirt’". There are three cards, each with an image and a question:

- enhatt 1
- kläder 2
- enkjol 3

Figure 5 – Further exploration into Duolingo’s language learning process.

When the user selects any of these options, a soundbite will play in the foreign language. In this case – Swedish. The site/app will do this a few more times before it advances to the next section. Now that the user knows a few words and how to construct sentences, the app can begin to insert these new words into sentences. If the user is correct, a green bar pops

Figure 8 - Further exploration into Duolingo's language learning process.

Figure 7 - Further exploration into Duolingo's language learning process.

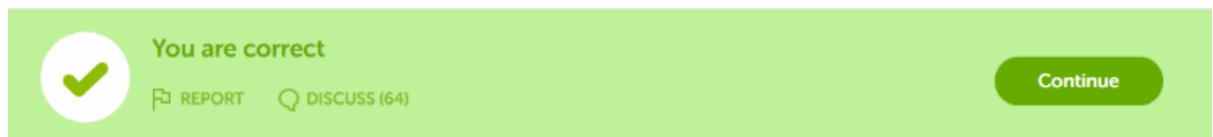


Figure 6 - Further exploration into Duolingo's language learning process.

up at the bottom of the page to tell them this. This entire process is done in one window, and in one page, making the users experience tight and quick, in addition to being incredibly simple and easy-to-use.

The example above shows how the site displays a sentence to the user with a relevant word (in this case it is clothes, with *klänning* being the Swedish word for dress), then prompts the user for a response in English – meaning the app asks them to type out what the sentence is saying in Swedish into English. The way Duolingo teaches users languages is decidedly complex, but effective. While I do not feel that I will include features like audio / visual cues in my final prototype, they would be a valuable luxury to include if the project were to be continued. Its design is simplistic, yet incredibly effective, and it is considered to be one of, if not *the best free language learning software available [PCMAG, 2018]*.

Advantages

- Offers 25 foreign language courses.
- Offers 3 constructed language courses.
- Completely free.
- Tracks user progress, their weakest words, and schedules regular refresher courses.

Disadvantages

- Only useful for short study sessions.
- Will lack complete comprehensive breakdowns that paid language software will not.
- Cannot and does not correct user pronunciation.
- Some language courses are not fully complete, and some are so incomplete they are not available for use.

3.4.2 Memrise

Overview

Memrise is a more in-depth, albeit much more complex piece of software than Duolingo. It provides a few more ways of learning the language, and does so in more engaging, one-on-one way, providing the user with more of an emphasis on “casually” using the language and interacting with actual foreign people.

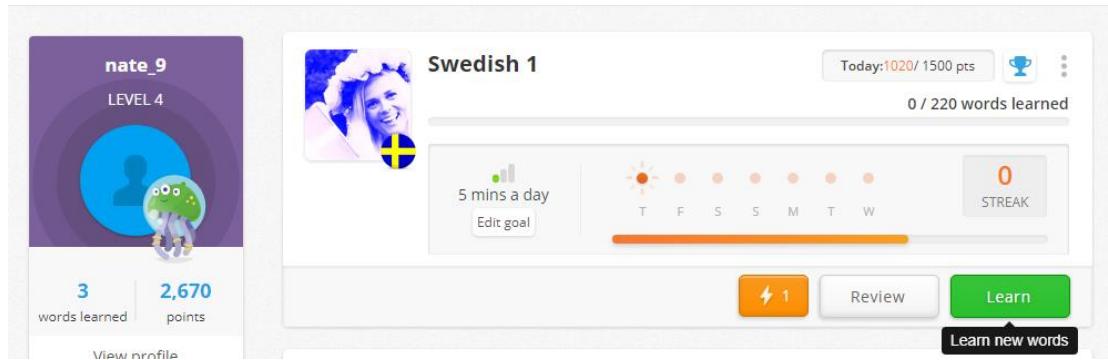


Figure 9 - Exploration into Memrise's language learning process.

First, it can be noted above that it does not split the learning into specific categories quite like Duolingo does. While it does split lessons categorically in places, it is instead done in a way where you must learn a variety of new words relating to speaking to people. For example, the first section is called “I come in peace”, and is a variety of different languages and sentences you can use to say greet someone, say goodbye to them, or interact with them on a basic, pre-conversational level.

This screenshot shows a grid of language learning cards. At the top, it says '10 / 220 words learned (0 in long term memory)' and '0 ignored'. Below are two rows of five cards each. The first row includes 'Launchpad!', 'I Come in Peace', 'Being Human', 'Fuel Your Vocab: Food', and 'What Do You Like?'. The second row includes 'Where in the Universe?', 'Fuel Your Vocab: 10, 9, 8 ...', 'Them and Us', 'Getting Fed', and 'Tell It How It Is'. Each card has a small icon of a flowerpot and a number indicating the word count. At the bottom, there's a note 'Phrases: Make Them Smile'.

Figure 11 - Further exploration into Memrises' language learning process.

This screenshot shows a list of learned words under the heading 'Level 1 Launchpad!'. At the top, it says '13 words' and '0 ignored'. Below is a table with columns for the word, its meaning, and a status indicator. The words listed are: 'hej' (hello, hi), 'dur är läget?' (what's up?), 'nu kör vi!' (let's go, let's do this!), 'skål!' (cheers!), 'tack' (thanks; thank you; please), 'varsågod' (you're welcome; here you go), 'ja' (yes), 'nej' (no), 'förlåt' (I'm sorry), 'godmorgon' (good morning), 'godnatt' (good night), 'vi ses' (see you), and 'hejdå' (goodbye). Each word has a small flowerpot icon and a status bar indicating its current status.

Figure 10 - Further exploration into Memrises' language learning process.

Memrise's approach creates a much more complex, but well-rounded language learning environment for the user, as it uses many different processes to try and teach users the language. One of the ways it tries to prepare the user for casual conversation is through the process of showing the user an actual person speaking a word or phrase.

The image shows a screenshot of a language learning interface. At the top left is a video frame of a woman with blonde hair speaking. A green circular icon with a white play button is overlaid on the video. At the top right, there is a purple button labeled "1.hejdå". Below the video, a question asks: "Choose the correct English for the Swedish above:". There are four options numbered 1 to 4:

- 1. to return
(something)
- 2. what's up?
- 3. I loved him
- 4. fortunately;
luckily

Below the options is a "See answer" button with a circular arrow icon.

Figure 13 - Further exploration into Memrises' language learning process.

Figure 12 - Further exploration into Memrises' language learning process.

What happens here is that the user of the site, upon entering into a category (for the example above, it is the introduction) is shown a clip with audio and visual of a real person speaking a word or phrase in a foreign language. The user must then select what word they think was being spoken, and then the site will inform them if their answer was correct or not. The process of having a real person speak a word or phrase to a user is one that I feel will help people in using the language in a social environment, and also help their pronunciation skills when learning this new language.

In addition to offering this face-to-face skills memorisation, Memrise also offers an approach like Duolingo – one example being where the site shows a word or phrase in English, and the user must spell the word or phrase in the foreign language. In the example below, the site is asking for “let’s do this” in Swedish which would be “nu kör vi”, once the user clicks the buttons below in the correct order, the site will inform them the answer is correct. This format is also used in a similar way, except the user must use their keyboard to type the word or phrase out (the site offers clickable buttons below the input box for accents, in case the user’s keyboard does not have them).



Figure 14 - Further exploration into Memrises' language learning process, regarding sentence choice.

One final process Memrise uses is giving the user a word or sentence, and clicking the correct sentence from a multiple choice of sentences. This assists users in learning basic sentence structure, further feeding into its overall goal of helping users to use the language they are learning in a face to face, social environment, rather than just being able to read, write, and listen to the language similar to Duolingo.



Figure 15 - Further exploration into Memrises' language learning process, regarding sentence choice.

Advantages

- Assists users a lot more with pronunciation, and the personal / social aspect of speaking a language...
- ...possibly leading to a greater chance of vocal fluency.

Disadvantages

- As the site is free, the translations may not be of the best quality.
- All sentences / words are user submitted and user checked, which could lead to incorrect translations (intentionally or not).

3.4.3 Busuu

Overview

Busuu is another language learning resource that is free for users who create an account with the site. While features are present for users to learn, some are restricted or purposefully attenuated. It functions similarly to Memrise, showing a flashcard with a still image, accompanied by the foreign word with the word below it. It also plays an accompanying soundbite.

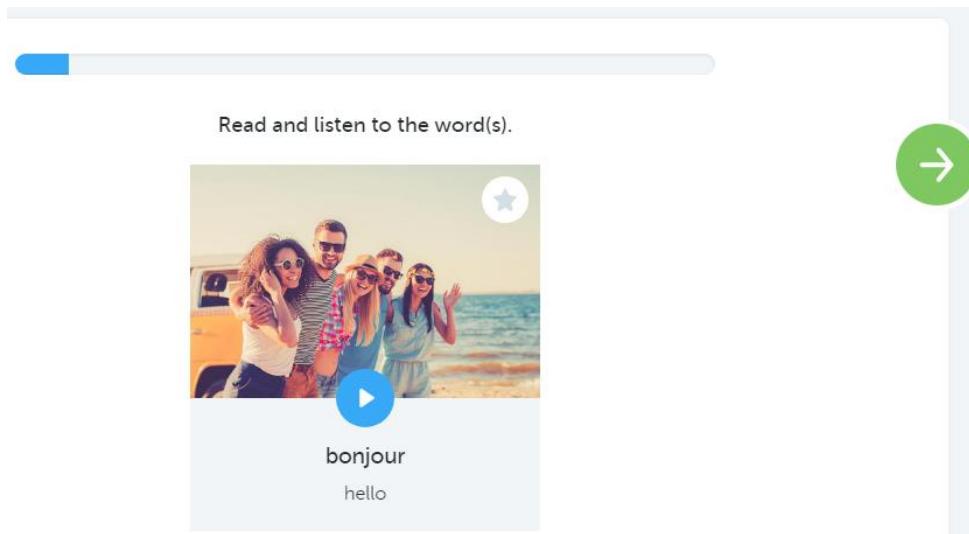


Figure 16 - Exploration into Busuu's language learning process.

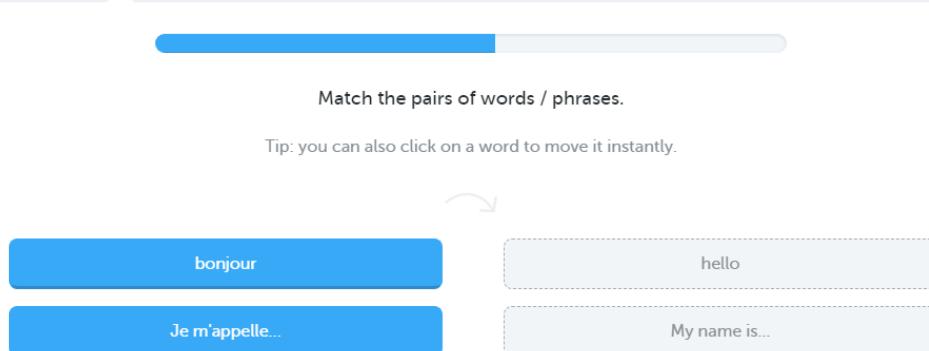


Figure 17- Exploration into Busuu's language learning process.

In addition to the example shown, Busuu also shares a similar learning method to Duolingo—matching words / phrases to their counterparts. This seems to be a popularly used, and therefore possibly much more efficient method of teaching a language. This method seems to be one to emulate for my own site when it begins development.

Advantages

- Jack-of-all trades approach. Seems to be a “light” version of both Duolingo and Memrise in one.
- Is adequate in all methods it delves in to (flashcards, soundbites, word-matching etc).

Disadvantages

- Overly simplistic.

- Requires user to pay to become a premium member before accessing advanced features that are very beneficial to the user such as PDF files, and lessons that are designed to assist learners with reading / writing.

3.5 Regarding Usability

As stated previously, my website design is supposed to be as inclusive as possible. While it is still a prototype and some features may still be missing or in development, it will have to be appealing to as many people as possible. Methods to do this include:

- Gender neutral colours, so as to not discriminate between male and female.
- No childish or overly-professional designs, to maintain an atmosphere that is designed not to restrict user access by age.
- Larger text (or at least an option for it) so that those who may be visually impaired or have reduced visibility.
- Clear colour use – there is a reason black text on a white background is prevalent in almost all websites; it is easy to read and distinguish from other page elements.
- Not an overuse of soundbites or a reliability on audio features so as to not exclude the hard of hearing or the deaf.
- Maintain the platform as free-for-use, so as to not exclude those in less fortunate economic situations.
- No overly-complex designs that may throw off younger users, or those who are novices to computing.
- No colour-sensitive or colour critical site elements, for those who may have deuteranopia, tritanopia or other colour-blind related impairments.

When looking at these features, I must then apply them to my site through the lens of usability, and its most important tenets (at least in my opinion); effectiveness, efficiency, and user satisfaction. Site usability is a critical factor, and is the decider on whether or not a user will come back to continue using the service, whether they may return later once the site is in a better place aesthetically, or whether they may not come back at all, due to having such a poor experience that leaves them with the impression the site is inefficient, ineffective and dissatisfactory. When creating the site, it is imperative I take each of these three usability problems into account, such as how to make the site more

- Efficient

Site efficiency hinges on how well the user can make use of site features once they are familiar with them.

- Effective

Site effectiveness means what the quality of the results from the site is. In terms of language learning, this could mean how fast a user becomes fluent in the language, or how quickly they learn and are able to recite by memory learned words.

- Satisfactory

User satisfaction is a bit harder to measure, as it is subjective. It could be measured externally by counting and logging user visits to the site; how many users are visiting? How many are new users? How many are regulars? Is the site maintaining regular (and hopefully consistent) traffic? The opposite approach is internalised measurements. This means that users on the site could be asked to fill out a survey on how satisfied they are with the site, with accompanying questions being either open ended, or multiple choice.

However, as it is a measurement and a request of objective user opinions, an open-ended survey is preferable.

Concerning how the site will look in general, when considering all users, a suitable colour scheme should be chosen. While this does not completely discount the use of colours other than white for a background, it is important to remember how these colour schemes look when considering all users, and how the use of that colour may affect different users, and play into difficulty using the site (yellow text / backgrounds are often less visible and therefore very problematic for usability.) Use of graphics and images should be lessened, as they are not important for site function and can result in cluttered layouts, along with conveying an air of unprofessionalism. Graphics should be clear, crisp and not look "stretched" or pixelated. Any photographs used (unlikely in this context) should be high-quality and look appropriately professional for the atmosphere the site is attempting to convey. In addition to making sure images and content is aligned properly, the site should also be tested on different screen resolutions, screen colour settings, browser types, and possibly OS types. By creating and maintaining a readable, usable presence on multiple browser platforms (Chrome, Firefox, IE, Opera, Safari) and resolutions (1920 x 1080, 1024 x 768, 1366 x 768) the site will boast usability for as large an audience as possible.

The website's layout and thereby the elements inside it must be easily navigable, on clear display and require few user operations to access said related content. Users should not have to scroll far down a page, or click through several windows to access what they came to the site for. Site mapping must be logical, and labelled buttons / tabs should take users exactly where they say they should go.

Finally, content relating to the sites purpose should not be hidden away. The user is coming to the site to learn a language – not learn how or why it was created. While contact pages are important, the existence of language-learning content and lessons associated should be clear to the user. Content should also be refined and of high-quality.

3.6 Literature Review Conclusion

In conclusion, it is clear to see that there are many benefits to be found when considering the use of having knowledge of one or more MFLs both in the professional world and the social world. Having knowledge of a modern foreign language can make a person more focused, able to make decisions quicker, appear as a better-quality candidate and a wiser hire and potentially a more tolerant, accepting person who is able to communicate with an even larger number of people than would be possible knowing just one language.

While it is more than possible for a person to become meaningfully employed and have meaningful social interactions with others knowing just the one language, the possibilities and opportunities offered by understanding more people, possibly worldwide, are definitely quantifiable. Knowing and practising a MFL in the business world and the social world are a positive investment for people that produces positive results in their own lives and perhaps the lives of others.

I feel that the creation of my project will serve as another medium for learning that people can create studies from – for example, the amount of people who successfully learn the basics of a MFL by using StanLearn can be quantified, and therefore the conclusion drawn can be that the method I employed to teach people a new foreign language was successful – or not. Perhaps if my prototype was used as a form of experimentation, like exposing children to learning the basics of one or more modern foreign languages, and testing if they

have measurably better decision-making skills than children who were not exposed to these MFLs. Or even using StanLearn to teach older people MFL basics and see if it reinforces the hypothesis that learning a language can lower the likeliness of Alzheimer's and Dementia.

Speaking from a design point of view, I now have a much clearer idea of what exactly my site will look like, and what the process will be that it uses to teach users the language they wish to learn. Ideally, my design will have a clean, sleek look that leaves no part of the site hidden or difficult to access, and provides the user with a “missing word” scenario that they must fill in based on prior knowledge. There will be limited use of images, and more of a focus on created graphics (not too many, to maintain an air of professionalism), a focus on a gender-neutral design, a focus on clear and perhaps large text size and finally a focus on the textual side of the teaching method that has been found in the examples above. This means that video and audio samples will not be included in the site, unless an opportunity to find speakers of German and Spanish arises within the time I have left.

Chapter 4. Problem Analysis

This chapter will concern the wider problems of the area my project operates in. This includes a discussion of the wider problem relating to people learning MFLs, what my ultimate goals are for this project, and what exactly I am testing with the process of creating my website.

4.1 What is Being Tested?

My hypothesis is that by using the method of having users view and memorise foreign words from a word list, and then apply those memorisations to a quiz, they can successfully learn and recall words in order to provide a base with which they can build MFL skills onto later. In addition to seeing if this method measurably works, it also tests to see if a fine-tuned user experience can bring in people of all ages to start learning languages, and raise the number of people who can speak one or more MFLs in addition to their native language.

4.2 The Problem

The problem at large is that not enough people are learning MFLs, and so are missing out on huge career and social opportunities, as well as a number of proven health benefits. StanLearn as a project aims to tackle this problem by creating a website with an atmosphere that is welcoming to as many diverse groups as possible, including diversity of age, gender, and economic background. By providing a free alternative to beginner level premium language services, StanLearn hopes to raise the overall MFL literacy rate, mainly of native English speakers, who as shown in earlier statistics in chapter 3, have poor grasp of any other language apart from their native English. A lot of what the wider problem is has been explained above in chapter 3, but I will reiterate it here explicitly and with more focus.

Seeing as the problem is that people are not learning MFLs and maintaining a level of foreign literacy, there are a number of areas where this problem could've developed. First, a reasonable assumption can be made that some of these people were not taught the subject at school to a level they found satisfactory, or not taught at all in some cases. To tackle this problem itself, StanLearn aims to create an environment that can be accessed by schools for use in lessons, or by pupils in their own time for free use and exploitation. By creating this online environment, a method for language learning has been deployed onto a digital platform that can be accessed by anyone and of any age. Given that people are becoming more and more technologically acclimated at younger ages [\[Reuters, 2013\]](#), making language learning resources available online increases their chances of being used by younger people and therefore assists in teaching them MFL skills that they will be able to

recall in social and professional environments, as well as provide them with cognitive and preventative health benefits as mentioned in studies in chapter 3.

Another problem in this field can be attributed to ignorance. As stated earlier, English is the most spoken language in the world, with an estimate of 1.12 billion total speakers of English [\[Wikipedia, Ethnologue, 2018\]](#) as a native or second language based on an accumulation of figures. Because of this, the problem exists whereby people who only speak English must be convinced of the benefits of learning another language; perhaps even if that person intends never to use it to communicate with others.

4.3 How StanLearn Intends to do This

It is therefore the purpose of StanLearn to show these people that learning at least the basics of a modern foreign language can be done simply, and that it is a skill that will assist them in more ways than just being able to understand more people in certain areas of the world. By creating a free, accessible and user-friendly website with which to learn a new MFL, StanLearn brings an easy method of learning new skills that people will be encouraged to use. Its simplistic design, easy to understand feedback and diversity in languages and topics will aim to prompt and hold the interest of anyone that visits the site. Even in its prototype stage, StanLearn offers the basics of Swedish, French, German and Spanish in four different categories. Coupled with the easy process by which more topics and questions can be added, StanLearn if developed in future could provide a suitable teaching platform that provides social skills and professional skills as well as enhanced cognitive ability that persists throughout life.

4.4 StanLearn's Goals

By creating and maintaining an online presence, StanLearn will first aim to improve the overall foreign language literacy rate of native English speakers, and aim to decrease the percentage of people in the UK that cannot speak any other language than English, which is 62% [\[British Council, 2015\]](#). In doing this, StanLearn also creates a secondary goal of helping anyone who speaks English to learn another foreign language by providing basic words in English along with their translations to help generate interest in MFLs for people to expand upon with more advanced courses.

Chapter 5. Methodology

This chapter and sections within will aim to explore the different methodologies that could've been and have been during this project's development cycle. It will describe the final selected methodology and how it was applied to this project.

5.1 Prototyping Method

Prototyping is the method by which multiple versions of the site are created and shown to the client to see which elements of the site they find positive and like, and also to identify which parts they don't like or wish to see changed. Using the prototype method would help the developer (me) to better understand the user requirements, as the client can go into as much detail as they like regarding which features and elements they would like to see implemented, which designs they would like changed along with which parts are satisfactory and unsatisfactory.

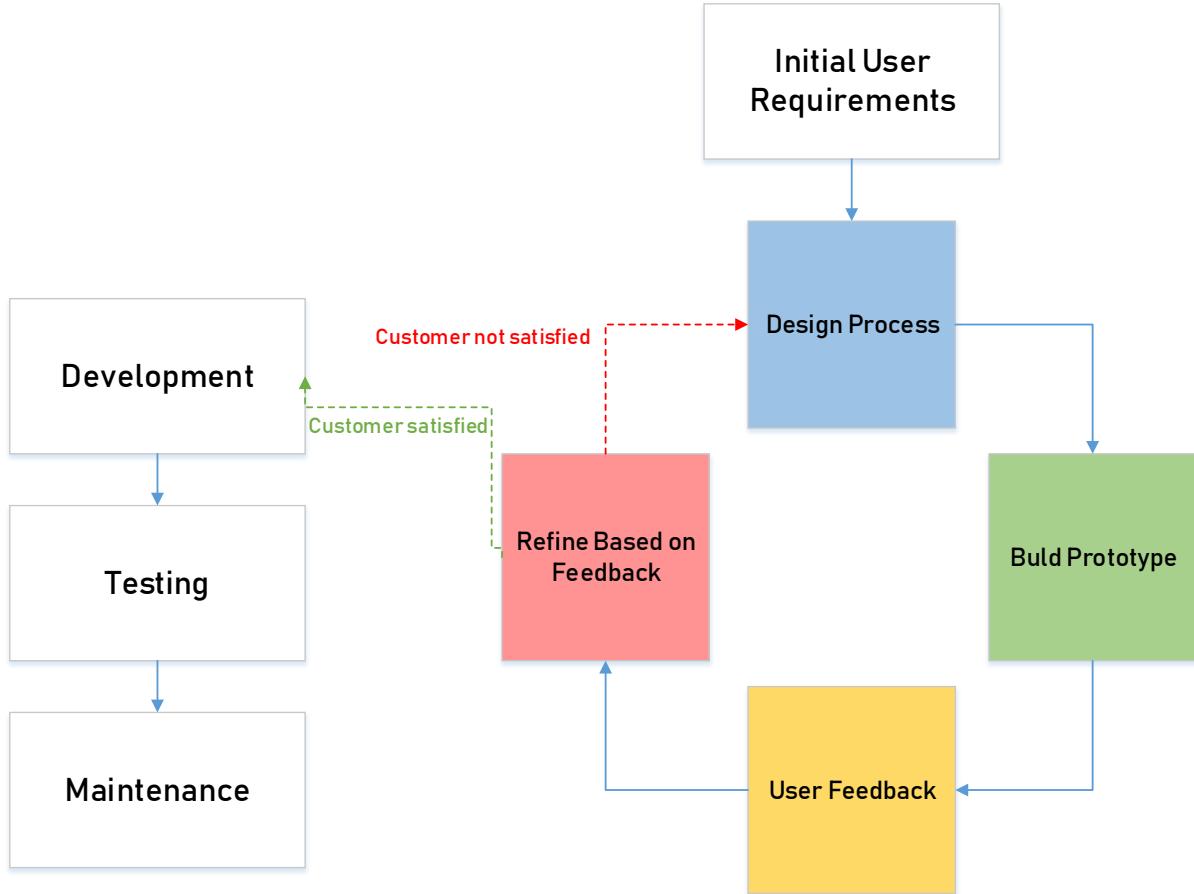


Figure 18 - Diagram explaining the prototype methodology.

The method of prototyping is shown here, and is simple to understand and follow. A big advantage of the prototyping method is that it allows direct user-developer contact, so it is more likely than not that the final product is exactly what the user wants, as they have had a much larger hand in developing it than would be usual. This results in a highly specialised product that has been purpose built for the user, and should have all their requests accommodated for. If not, as is visible in the chart, it is part of the natural methodology to have another pass at the design process, building the prototype again, obtain feedback, refine based on it and then the next iteration pends the user's approval.

Some advantage also rests in the ability to have multiple passes at a project – what this means is that if the developer produces the product but it is not up to the users' standard, they are more than free to design the prototype again based on that feedback. While this extends project time, it presents the developer with more time to fix small aesthetic issues in addition to logical issues (bugs, incomplete functionality, etc).

While this is heavy involvement is largely beneficial for the user, it can be problematic if the user does not know what they want in terms of specific design. This can happen if the user is vague or nondescript about features they want, or if the prototypes are not constructed as the user requests. This can lead to frustrations between the user and the developer and can extend the project development time by an unnatural and dangerous amount, increasing both financial and time costs. Another negative factor of having the user involved heavily is the user's potentiality to request increasingly complex features to make their design feel fleshed out, without recognising what amount of work the developer will have to undertake to include these complexities.

While during this project I will have extended access to some form of a client (project supervisor), the possibility of additional complexity being added considering how complex the project already is may not be a good choice of methodology, and can result in development time being extended past a necessary point. For this reason, prototyping will not be the selected methodology for this project.

5.2 Waterfall Method

The waterfall methodology is a much simpler and much more linear development process. It only requires one pass at development before it is ready, and reduces the amount of time spent revisiting the project, due to the way it's structure works.

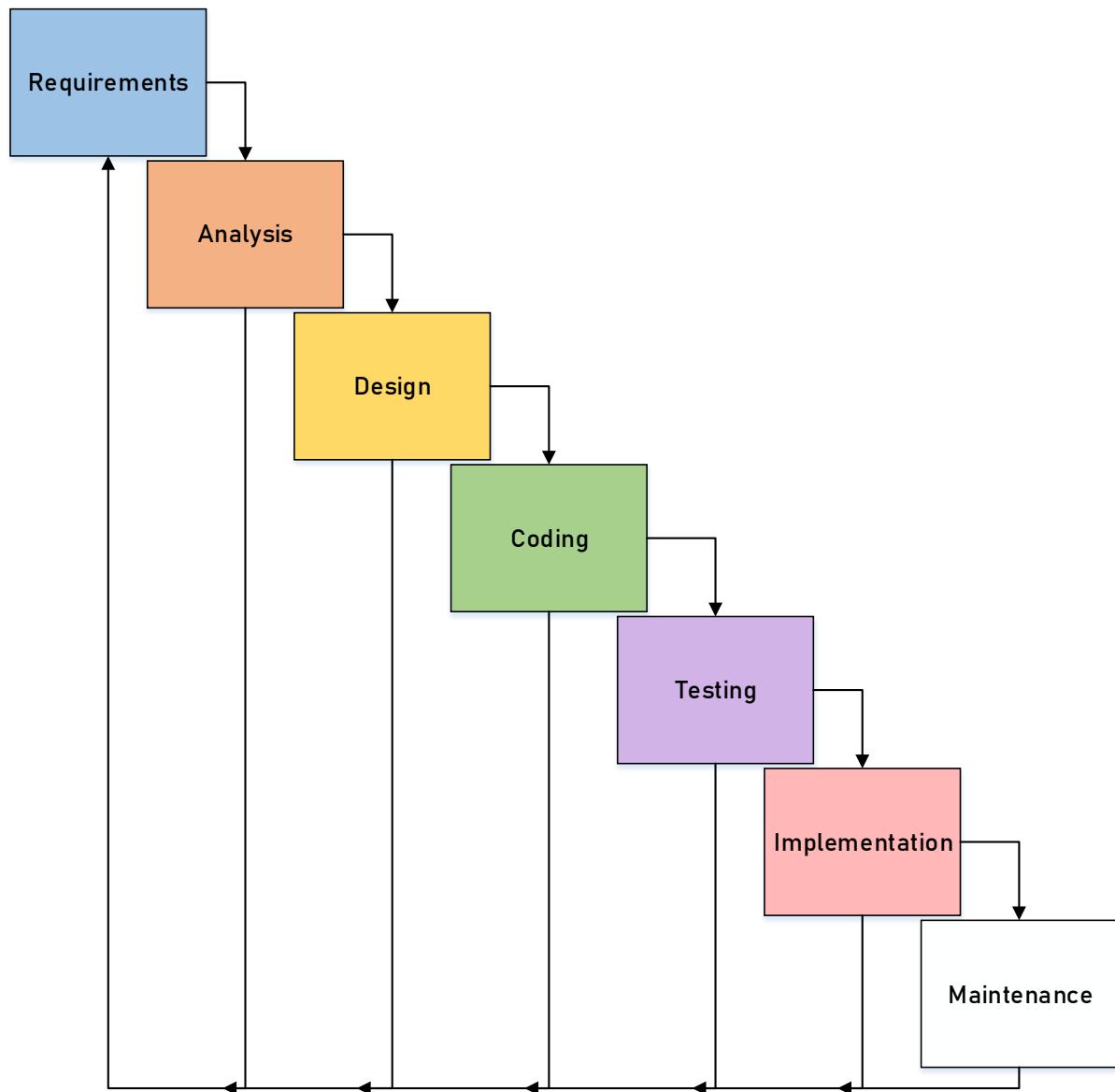


Figure 19 - Diagram explaining the waterfall method(ology).

As shown here, the waterfall model is continuous, and does not require the developer to keep going back and repeating steps, whereas in the prototyping methodology the user would have an extended hand in suggesting changes and amendments to be made in the site. The waterfall model is linear in that it is all done in succession from the last stage, and finally ends when the project is successfully coded, tested and implemented and only has need for interference in the maintenance stage (for updates and fixing of bugs). The repeat

arrow shown at the bottom of the diagram is there to signify how the project would begin again, meaning that individual stages would not be revisited, and that a new project would start in the requirements phase.

In the waterfall model, each stage is finished fully before the next stage is entered. While this is of course impossible for maintenance as that is the projects logical end, the rest of the stages follow this format. The tasks in these stages are as follows

- Requirements

The process where the user is asked what they need (require) from the software. In the waterfall model, these requirements *must* be known and described thoroughly by the user, as they will not be contacted again for involvement on the software until the testing phase.

- Analysis

Analysis is the assessment of the given requirements to see if the development project is worth pursuing. This results in the creation of models and feasibility tests before design begins.

- Design

Design is the process where the base of the software is created (namely its architecture). This is not where the software is completed, but where the base is created for functionality to be created and inserted later.

- Coding

Coding is where the functionality of the software is developed, and ultimately integrated with the design.

- Testing

In testing, the code and the design is compiled and implemented and testers look for and report bugs that interfere with functionality, as well as functionality that is missing or not working as intended. This is not the final product, but is essentially an almost-complete beta test before the software is rolled out.

- Implementation

Implementation is the rolling out of the final product into the environment it was created for. This step considers installation onto machines, whereby the user who asked for / commissioned the project will migrate their data onto the new software.

- Maintenance

Maintenance is the final step of the waterfall method, and signifies project completion. In this stage, the software has finished development, and is now only worked on when a bug is discovered, or when the user pays for more functionality to be added.

The waterfall model is a good example of only moving on to the next stage when the current stage has been completed to a satisfactory standard. The advantages of this model are that it allows the developer to be able to predict where problems may arise in the analysis stage. It also allows for feasibility studies to be conducted so that the project is much less likely to fall through into a dead end.

5.3 Selected Method and Justification

For my project, I have selected the waterfall method to begin developing my project. It seems like a much more valuable use of time, especially on a constraint, to evaluate how my project will look in each stage, and also provide me with valuable milestones on which I can base my completion rate off. I would much rather be able to first assess and plan out how each stage is going to work and look before I embark upon creating it, whereas with prototyping I will not have such luxury.

With prototyping, if I find later in my projects life cycle that it is too complex or that the features I desire cannot be implemented with the time left, I will not have a functioning website, if one at all by the deadline date. As well as this, it would become problematic to constantly arrange regular meetings with my supervisor for discussions on small details in the prototype, especially considering my project supervisor is not a client and only an example with which to discuss my prototype with. This does not seem an efficient use of time that could be better spent in the preliminary stages planning out the project before design begins. Expanding on the point about the client for prototyping, my project has no specific one client, or even group of clients. It is a project that requires use by many different age and ability groups and building the project around one client would defeat the purpose of the project.

5.4 Method of Requirement Obtainment

As mentioned above, I will be using the waterfall methodology and therefore I will need to gather the requirements needed before beginning my project. As my project will need to accommodate the requirements of a group of people, I will obtain these requirements using a questionnaire, which will be developed and distributed among a diverse group of people, with the numbers of this group being specified later in chapter 6.

Chapter 6. Requirements and the User

This chapter will delve into the different types of users and their requirements, and will go into further detail concerning personas and analysis of different users.

6.1 What Do Users Need?

Before asking this question, it is also important to note who the “users” are. In the case of StanLearn, it is not particular one user, but many, that could potentially be from much different situations from each other. For example, one visitor could be a child who is a computer novice and has little experience with websites, and the next could be an adult who is a computing expert who has been using websites for years.

6.2 User Requirements Questionnaire

This subsection details the questionnaire that was used to gather opinions and possible requirements that users may have. The pages and the questions within them are shown below. The questionnaire was created using google forms and was distributed among a diverse discord server which has members in many different countries in Europe and globally.

6.2.1 Questions Included

For the first question I thought it was important to know the questionnaire takers age. The reason for this is that it is a big factor in my sites purpose – the site needs to reach as diverse an audience as possible especially in terms of age, therefore I thought it appropriate to know how old the people taking the questionnaire are, so I know if my attempt to create the site to certain standards is justified. For this answer, I used a multiple-choice input, with 4

different options to reduce the chance of having data too highly specialised (under 18, 18-20, 21-24, 25-28, etc).

User Requirements Questionnaire, 6001PROJ 2017 - 2018

Please fill out the questions below to the best of your ability. Long, detailed sentences are welcome, but are by no means required.

How old are you?

*

- Under 18
- 18-24
- 25-30
- 30-40
- 41+

Figure 20 - Questionnaire made to gather user requirements.

The next question I asked was how many languages do you speak? This question is expanded upon later with a follow up question, but I thought this question would help me better understand the audiences' ability level, and so can adjust the site and its content accordingly. Again, this question didn't need specific answers, so pre-set multiple-choice answers were created.

How many languages do you speak?*

- 1
- 2
- 3+

Figure 21 - Questionnaire made to gather user requirements.

The next two questions I included were "what language is your native language" and "what languages can you currently speak now". Including these two questions next to each other provides me with some very informative information that will influence what languages I include on my website. For this option, I opted to use the short text input. The user needs to specify which language(s) they speak personally, so a written response is required.

What language(s) is/are your native language? *

Short answer text

What language(s) are you able to speak now? *

Short answer text

Figure 22 - Questionnaire made to gather user requirements.

After these two questions, I asked which language respondents would desire to learn. Instead of using multiple choice answers, I allowed the respondents to input a short textual answer. In hindsight, I would not have done this, or would've placed more emphasis on the fact only one language is required, as some respondents included multiple, as will be shown in the answers subsection (6.2.2).

If you wanted to learn another language, which language would it be? *

Short answer text

Figure 23 - Questionnaire made to gather user requirements.

Now that some basic information on respondent's language knowledge was attained, I next asked them what kind of features they would like to see on a MFL learning site. I also asked if the approach I was considering (the "fill in the missing word" approach) was an effective choice. I thought this would give me some good ideas of what to include in my site, and how to make the quiz work if my proposed method was found to be undesirable.

What features would you like to see included in a language learning website? *

Long answer text

Do you feel a "fill in the word" approach to learning a language is an effective * one?

Yes

No

Figure 24 - Questionnaire made to gather user requirements.

For the next part, I asked respondents if they felt the languages specified were good choices to implement in a language learning site. These were four very simple yes / no answers. I also asked respondents which language they would suggest adding to the site. I included this question as if one of the languages received a significant amount of negative responses, I could potentially remove that language and obtain a translator for the most suggested one,

and begin the process of including it. Also, if one language was suggested a lot of times, and if time constraints allowed it, I could possibly include a fifth language into the site. This seems unlikely, but would be a good addition to the site if I have time left to obtain a translator and create questions with answers.

Do you feel Swedish is a worthwhile language to include in the site? *

Yes
 No

Do you feel French is a worthwhile language to include in the site? *

Yes
 No

Do you feel Spanish is a worthwhile language to include in the site? *

Yes
 No

Do you feel German is a worthwhile language to include in the site? *

Yes
 No

Which language(s) would you suggest adding to the site? *

Short answer text

Figure 25 - Questionnaire made to gather user requirements.

The final question I asked was if the users thought my topic choices for the quizzes were good introductions to a MFL for beginners. This last question again used simple multiple-choice answers.

Do you think Animals, Numbers, Colours and Food are good starter topics for * people beginning to learn a language?

Yes
 No

Figure 26 - Questionnaire made to gather user requirements.

6.2.2 Questionnaire Results

From the questions included in the earlier questionnaire, I was able to gain invaluable information that assisted me with the site development process. The answers to each question will be explored in depth below. The questionnaire received 11 responses total.

The first question had some fairly predictable results. The most common age of respondents was 18-24, with 54.5%, this was followed by the 30-40 age range at 27.3%, then under 18 and 41+ were tied at 9.1% each. This has affirmed my belief that the site should be created more with young people in mind, but this result will not lead to other age ranges being less accommodated for.

How old are you?

11 responses

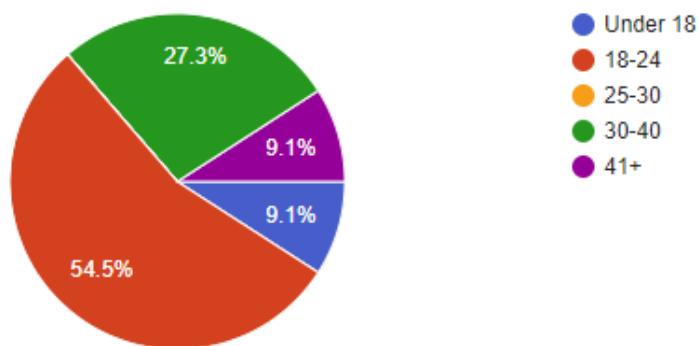


Figure 27 - Results obtained from responses to the user requirements questionnaire.

The second question also produced some predictable results, but did produce some surprising results regarding the amount of people who could speak 3 or more. Most people could only speak 1 language, with a percentage of 54.5%, seconded by this was people who could speak 3 or more, at 27.3%, and finally remained people who are bilingual, at 18.2%.

How many languages do you speak?

11 responses

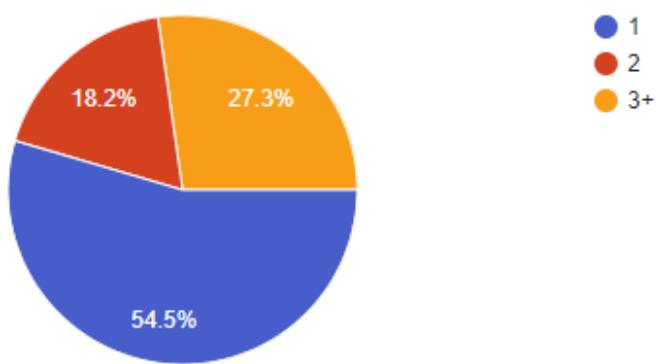


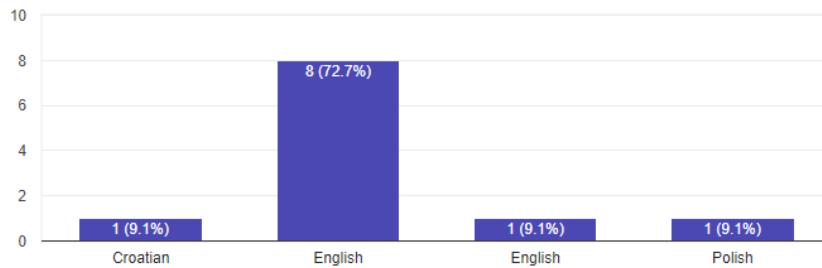
Figure 28 - Results obtained from responses to the user requirements questionnaire.

Next, was the questions regarding native language, and what languages respondents are currently able to speak. Unfortunately, due to a lack of foresight and an error in quiz compilation on my behalf, some answers to these questions contain duplicates. I will include the original graphs, and then I will resort them to display the answers more efficiently.

Regarding the data however, I was amazed to find one single respondent who could speak English, Croatian, Serbian and Bosnian.

What language(s) is/are your native language?

11 responses



What language(s) are you able to speak now?

11 responses

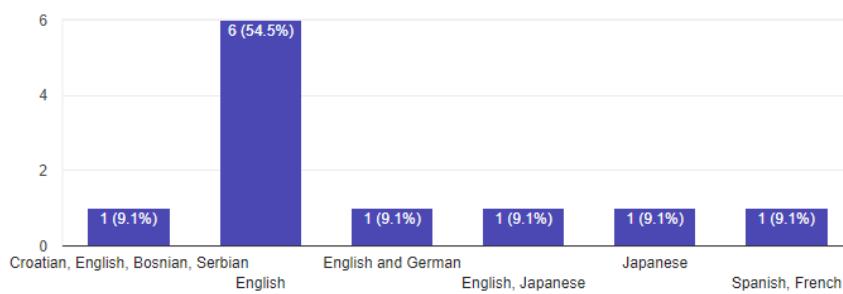


Figure 29 - Bar charts of results obtained from responses to the user requirements questionnaire.

This first fixed chart represents native languages. As is seen here, of the 11 respondents, 9 were native English, 1 was a native Croat and 1 was a native Pole. While Croatian nor Polish are to be included in the site, it is intriguing to see different native languages nonetheless.

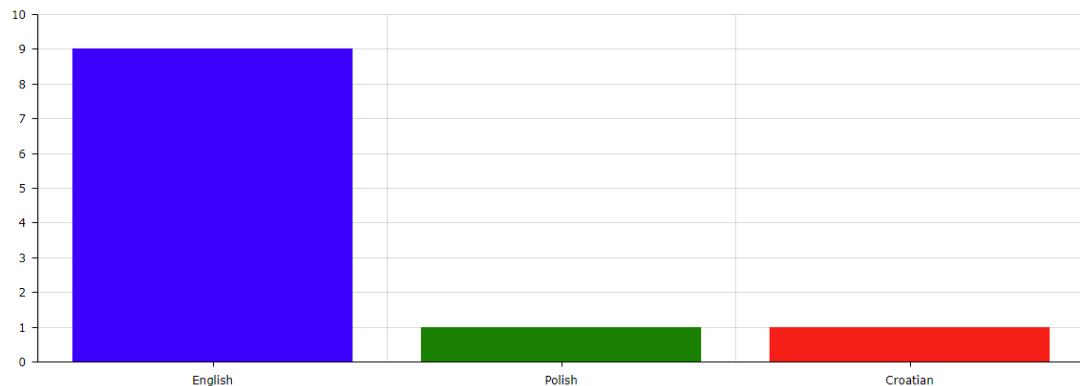


Figure 30 - Fixed bar charts of results obtained from responses to the user requirements questionnaire.

This second fixed chart is the what languages are you able to speak now chart. The number of English speakers has been amended to 11 (the number of respondents), as they all speak

English but did not include it in their answer. Polish has also been added to the chart, as this respondent did not include it in their answer.

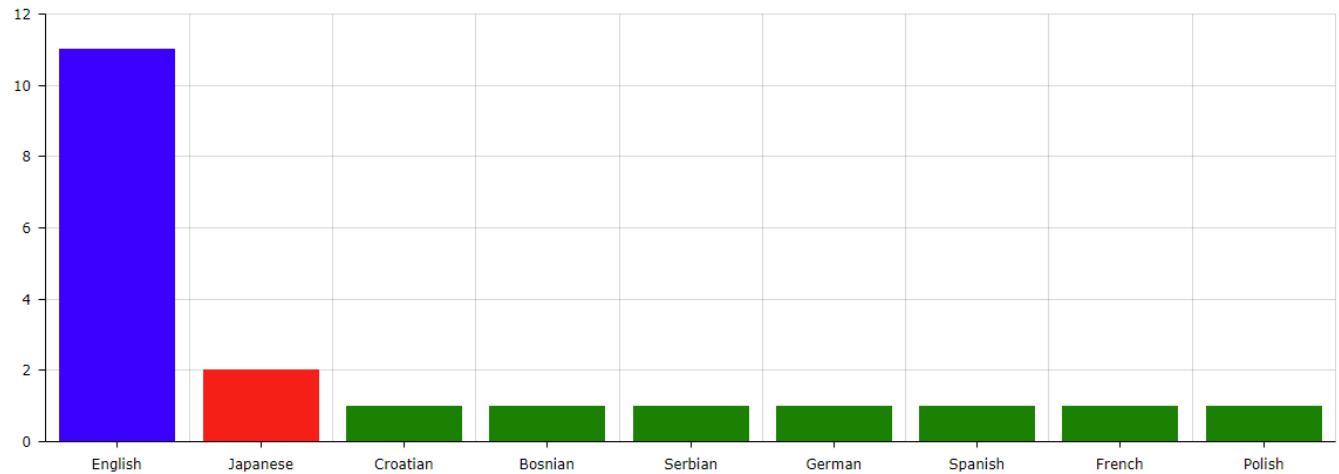


Figure 31 - Fixed bar charts of results obtained from responses to the user requirements questionnaire.

Overall there is an interesting mix of different native and spoken languages. Japanese was another standout, as these responses came from the two native English speakers.

After this came the question that asked respondents what language they would like to learn if they had the opportunity. This was an important piece of data, as it could decide if languages are added to or removed from the site. Fortunately, French and German are first and second on the list respectively. This is good, as work had already started on these two areas, it also plays into an earlier personal prediction that people would choose popular European languages. Some responses are quite unique, with Gaelic and Korean being unexpected but welcome suggestions, as well as three other European languages being suggested. Of these three other European languages suggested, I will continue with Spanish, to avoid typing with the Cyrillic alphabet (which I am not familiar with, and do not have contact with a translator for). The same rule applies to Norwegian and Korean as well – I am not in contact with a translator for either language, nor am I familiar with Korean or Norwegian letters. While Gaelic is an interesting choice, resources to learn the words for display on a translation guide may be hard to find.

For these reasons, French, Spanish and German will continue to be used, backed up by this list. Swedish will remain, as it is a language I can easily implement personally.

If you wanted to learn another language, which language would it be?

11 responses

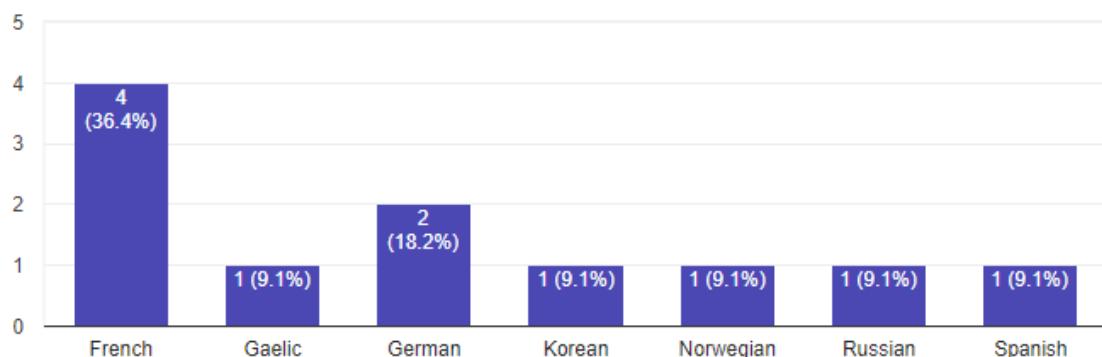


Figure 32 - Bar charts of results obtained from responses to the user requirements questionnaire.

Next, I asked the users which features they would like to see in a language learning website. This was undoubtedly one of the most important questions regarding gathering user requirements data. Because I wished for respondents to be thorough, I allowed long text answers. Responses here were very promising, and validated my early design ideas to include a translation guide. Some of the suggestions here were complex, but they were indeed features I had desired to include from the start, but their complexity prevented me from implementing these. If nothing else, this data can be used for additions to chapter 12, further work.

The main suggestions to takeaway here are the suggestion for the translation guide, filling in blanks and the fact that one user wanted the site to be free, which StanLearn will be.

What features would you like to see included in a language learning website?

11 responses

- Audio keys to help with pronunciation, informative video guidelines
- Just correct grammar
- Audio so the user understands how certain words are supposed to be said.
- SRS, games
- Maybe a translation guide along with sound bites
- Auto adjusting levels
- Voice recognition so it can correct your pronunciation.
- Recap of older lessons
- Flash cards that pop up on the screen and you have to type in the correct word.
- You learn by filling in the blanks in a sentence with a clue from a picture or video. It'll then say the sentence when you get it right so you know the pronunciation.
- Learn another language free!

Figure 33 - Results obtained from responses to the user requirements questionnaire.

Next, I asked for a simple yes or no on if the approach I desired to take to the site was an effective one. This approach being the “fill in the blank” method of testing whether the users memorised all the foreign words correctly. The results here were almost an even split with yes at 54.5%, and no at 45.5% but the results showed a positive reaction to the fill in the word approach. While I would like to include audio clips and visual flashcards as a method of learning, this is too complex for the current prototype state that StanLearn is in, and may result in the deadline being passed with incomplete work.

Do you feel a "fill in the word" approach to learning a language is an effective one?

11 responses

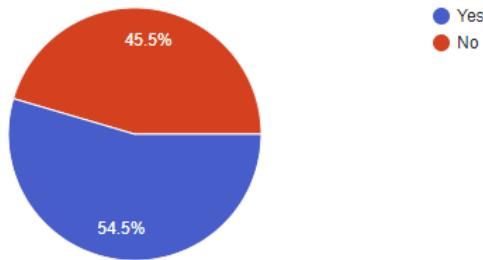


Figure 34 - Pie chart of results obtained from responses to the user requirements questionnaire.

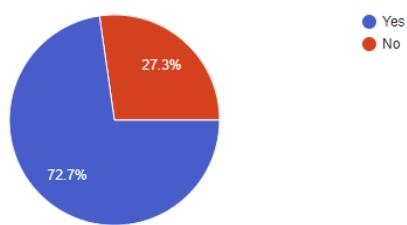
Next came questions asking whether users felt the languages I added to the site were worthwhile suggestions. First, Swedish was considered the least worthwhile by a small margin of 27.3% - not a deciding vote to have it removed from the site by any means, but

should the opportunity present itself to remove Swedish in favour of a different language (despite it being the one I know best), that is the course of action that will be taken.

German and Spanish are considered to be of the same worthiness, both only scoring one vote against them, but French is clearly considered the most worthwhile addition to the site, with 100% of respondents indicating it is worth keeping in the site. This result, in addition to French being the most desired language to learn has resulted in it being guaranteed inclusion in StanLearn.

Do you feel Swedish is a worthwhile language to include in the site?

11 responses



Do you feel French is a worthwhile language to include in the site?

11 responses

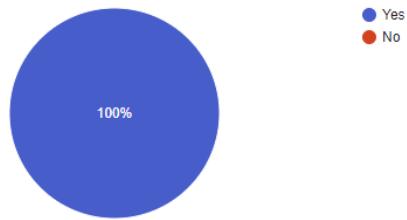
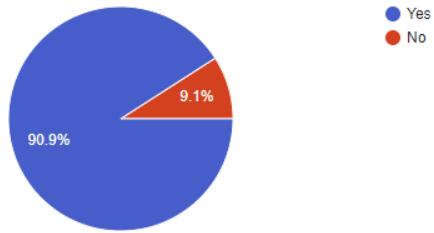


Figure 35 - Pie charts of results obtained from responses to the user requirements questionnaire.



Do you feel German is a worthwhile language to include in the site?

11 responses

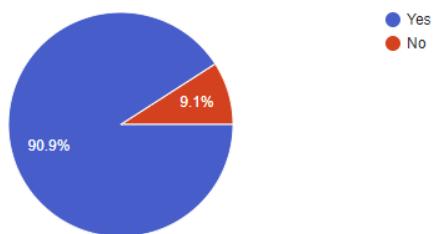


Figure 36 - Pie charts of results obtained from responses to the user requirements questionnaire.

Finally came the last two questions – the first being which languages would you suggest adding to the site, and the last asking the respondents if the topics I have suggested are suitable for first-time MFL learners.

The first question allowed users to submit their own answers, so as with previous large text answers, I will show the raw table and then show an ordered table I have created.

The first question produced some interesting results. While Latin and Ancient Greek are not MFLs, they will be included in the table for the sake of including all data, but they will be in black. There seems to be quite an interest in Asian languages being added to the site, with Chinese being the most popular pick. Perhaps when a line of communication is opened with a translator, this language may be implemented at a later date. Aside from the other Asian languages of Korean and Japanese, interest still remains in the usual most popular European language choices – German, French and Spanish, but now they are joined by suggestions of Norwegian and Finnish, alongside English. Given that the StanLearn site itself is in English, the latter seems a bit redundant. Trailing behind with only one vote on each is languages like Russian and Gaelic, that do not seem to be in high demand according to the results.

Which language(s) would you suggest adding to the site?

11 responses

Korean and Russian
Croatian
Italian
Japanese
Norwegian
Japanese, Korean, Chinese
English, French, German, Spanish.
Finnish, Norwegian etc. Related languages.
Latin, Gaelic, Welsh, Ancient Greek
English, French, German, Spanish, Mandarin Chinese, Russian.
Chinese

Figure 37 - Results obtained from responses to the user requirements questionnaire.

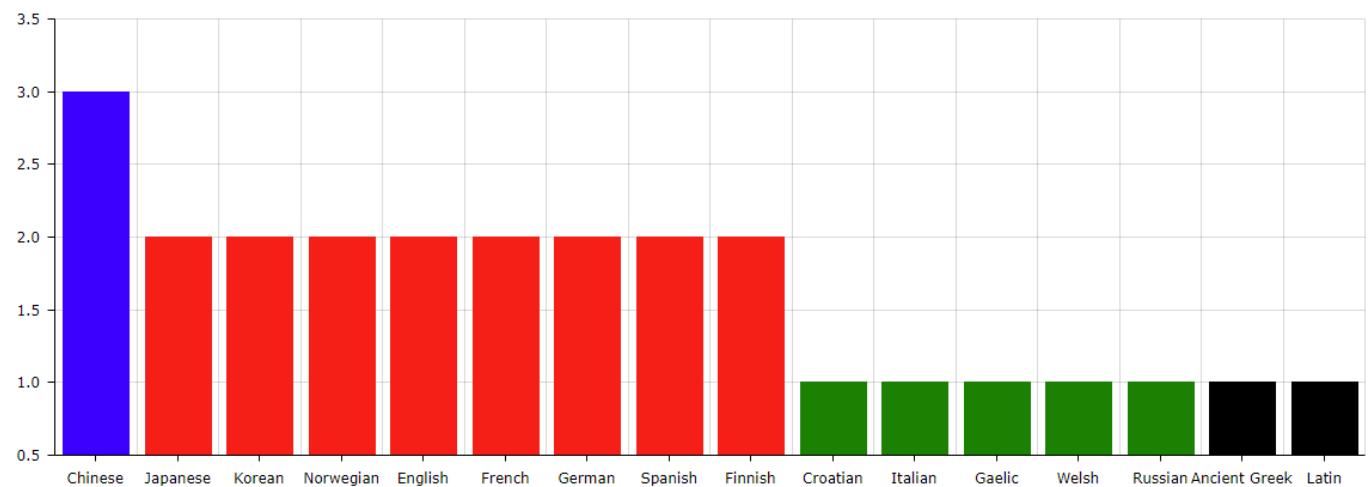


Figure 38 - Fixed bar charts of results obtained from responses to the user requirements questionnaire.

Finally, the last question is regarding whether or not the topics chosen for StanLearn are appropriate for beginners to MFL learning. All respondents indicated this is a good starting point, and proves my hypothesis that creating the site for beginner level users is the best approach.

Do you think Animals, Numbers, Colours and Food are good starter topics for people beginning to learn a language?

11 responses

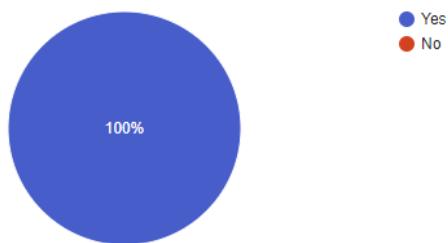


Figure 39 – Pie chart of results obtained from responses to the user requirements questionnaire.

6.3 Functional Requirements

For the site to fulfil what it has been created to do, it must first be noted what its functional requirements will be. Functional requirements are elements of the site that must be present for it to serve its purpose (or “function”). The site *must* contain these functions, or else it will be considered that it has not performed its purpose.

- Users must be able to navigate to the desired language area.
- Users must be able to access the quiz.
- Users must be able to attempt the quiz.
- Users must be able to select their answers to the quiz.
- Users must be able to submit their answers to the quiz for checking.
- Users must be able to see their results upon completing the quiz.

6.4 Non-Functional Requirements

In addition to its functional requirements, the site has requirements that are not related to functionality, but must still be present to be considered a success.

- Pages must remain gender and age neutral in their presentation.
- Pages must be suitable for people of all computing abilities.
- Pages must be somewhat accommodating for those hard of sight.
- The site must have a consistent theme to appear appropriately professional.
- The quiz must have questions suitable for those of no prior MFL experience.

6.5 Personas

This section will concern the different personas that are not just possible, but more likely to visit this site.

6.5.1 Young Teen / Child Persona

Jack is an 8-year-old boy who belongs to a new MFL class in a primary school. The teacher has decided to use StanLearn to introduce the children to French, as the site assumes the user has no prior experience with MFLs and has a clean, simple, and user-friendly layout the children will find easy to understand and navigate.

Jack is excited to be working on the computer instead of sitting at a desk and learning new French words from a worksheet by reading them and then covering them up and seeing if he remembers them. He finds the computer a more engaging way to test himself on these new words, and likes the quick feedback the quizzes give him when he gets a question wrong or

correct. He also finds the sections entertaining, and particularly likes testing himself on the names of different animals.

6.5.2 Teen Persona

Amy is an 18-year-old who wishes to learn German before she travels to study abroad there for a year, where she will be earning her degree in marketing. She has no experience with German and cannot read, write, understand or speak any other language apart from her own. She is an exceptionally social person and wishes to learn German so that it is easier for her to make friends when she lives there, as she is worried not everyone will know her language and therefore not able to have proper conversations.

Amy quickly searched up how to learn the basics of German quickly for free before she moves away. She finds StanLearn a more than adequate platform to begin building her skills and finds it a good base to advance her skills off when she learns conversational basics later on.

6.5.3 Adult Persona

Derek is a 35-year-old businessman, who has been in the finance sector for 10 years. During this time, he has risen up his company's ladder and now attends many high-level meetings regularly. His company wishes to branch out and conduct business with other companies whose headquarters are in Sweden, but for this to happen the company needs a translator who can create relations between the two companies, and begin discussing the benefits of dealing with one another.

Derek already knows one language and has been responsible for managing relations with foreign businesses before, and Derek has expressed willingness to learn another language to advance not only the company's standing, but also his own career. Derek's boss has decided to save money and not hire a translator, and is now instead requesting that Derek learn the basics of Swedish in his own time before moving to a more advanced course the boss will pay for. Derek stands to gain a lot if he is successful in this endeavour, but also may risk angering his boss and losing a significant position in the company should he fail.

Derek enjoys the new opportunity given to learn a new language, and can find all the basics of Swedish he needs on StanLearn before advancing to the paid course his boss has enrolled him on. He can learn these new words and test himself later on to see if he has remembered them, and imagines situations where they may be useful.

6.5.4 Senior Persona

Ethel is a 60-year-old woman who has reduced sight capabilities and little computer experience. She wishes to learn Spanish before she and her longtime partner retire abroad, where the language is widely spoken and in some cases necessary for comfortable life. However, she also worries that it is too late to start learning now, as she is in her later years.

She decides she needs a starting point, but finds trouble committing herself to regular learning. Instead of paying for a app, site, or service to learn Spanish, she decides to look for a free alternative to see if she is really interested, and if she believes she can commit to learning this new language. She searches for free language learning sites and sees StanLearn. As the interface and layout does not discriminate based on age, and as the content on the page is clearly spaced out with larger text sizes she has little trouble reading the questions and picking her answers. For Ethel, it is refreshing to see a website that has been made for her level of computing and foreign language ability, accommodates her impairment, and looks simple yet professional enough that she doesn't feel silly for using a learning website.

Because her experience was so positive, Ethel decides to purchase a full Spanish learning kit, and continues her lessons thanks to the introduction from StanLearn that showed her she was perfectly capable of learning a language, despite her preconceptions based on her age.

6.6 Use Case UML Diagram

The diagram below shows how users will “use” the site to gain access to different areas. Since only one user type will access the site (there is no account system), only one user type is needed.

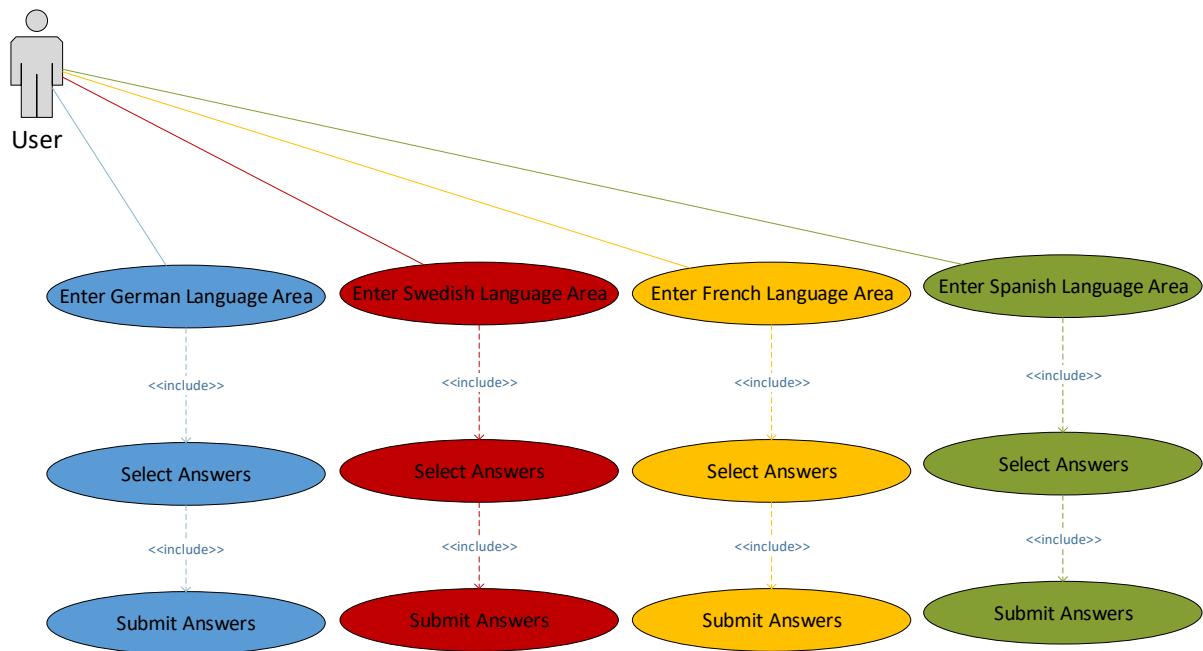


Figure 40 - Diagram depicting use cases.

Chapter 7. Project Design

This section will concern the stages of design the project entered, the final layout of the site, as well as the aesthetics of individual pages in the storyboarding section.

7.1 Site Mapping

The site map below shows how navigation between pages works in StanLearn. While the navigation bar to all areas of the site is available on every page (apart from the quiz pages) for usability's sake, this is the most efficient way of mapping out the site, and showing the logical steps users take to get to their desired areas.

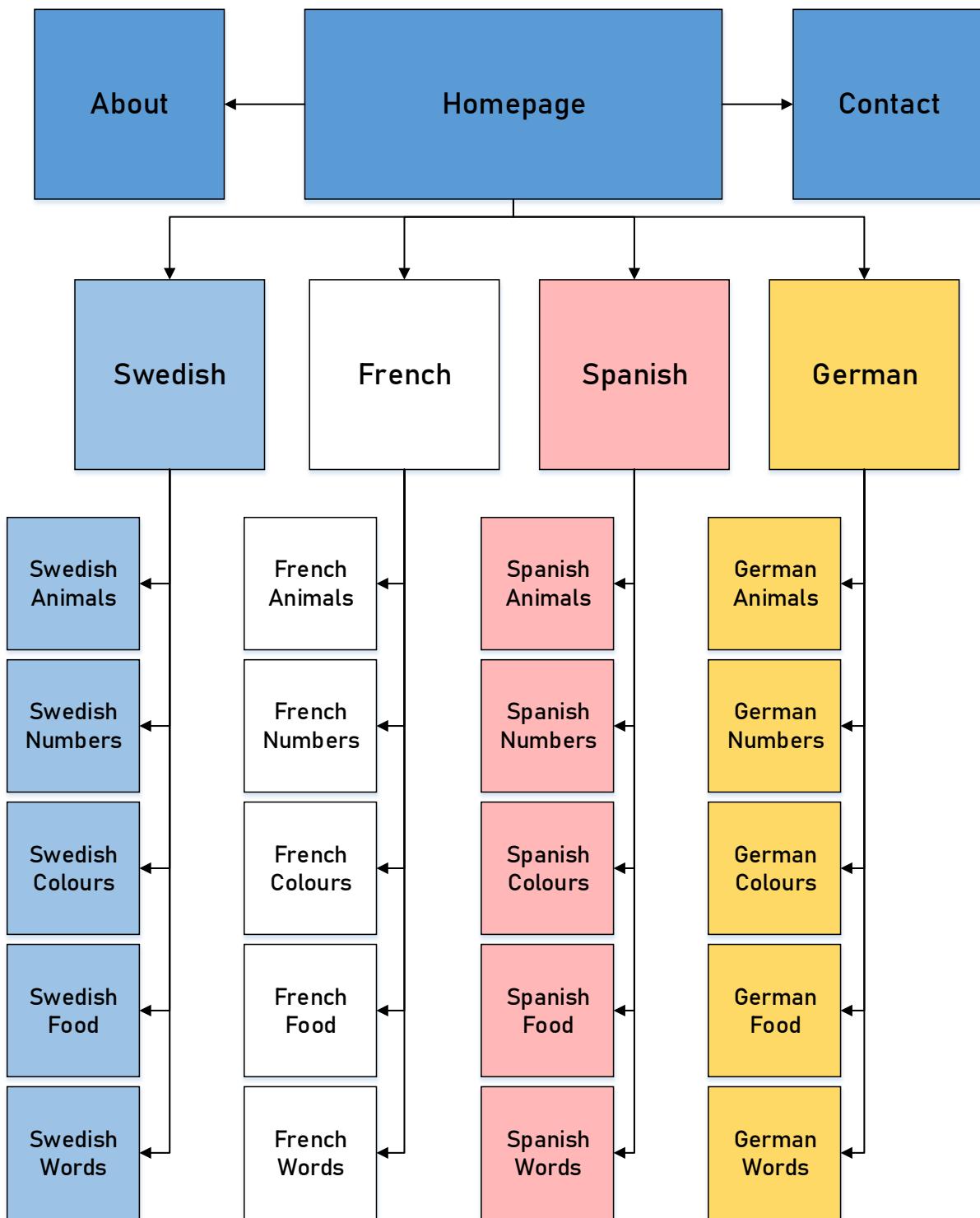


Figure 41 - Site mapping diagram.

As is visible in this diagram, once the user decides what language they want to learn, they need only click on the language in question, and they will be greeted with the options to navigate to their test of choice, or view the words so they can memorise them, and then test how well they remembered them.

7.2 Site Storyboarding

This subsection and the subsubsections that follow it will contain storyboarding images that were used to design and decide the site's layout early in development. The storyboards

should represent my desire to keep the theming consistent, along with emphasis on attention paid to small details which could affect the different user personas as described above. This section contains early storyboards of what I wanted the site pages to look like, as well as what the attributes are in the elements in the final design.

7.2.1 Homepage

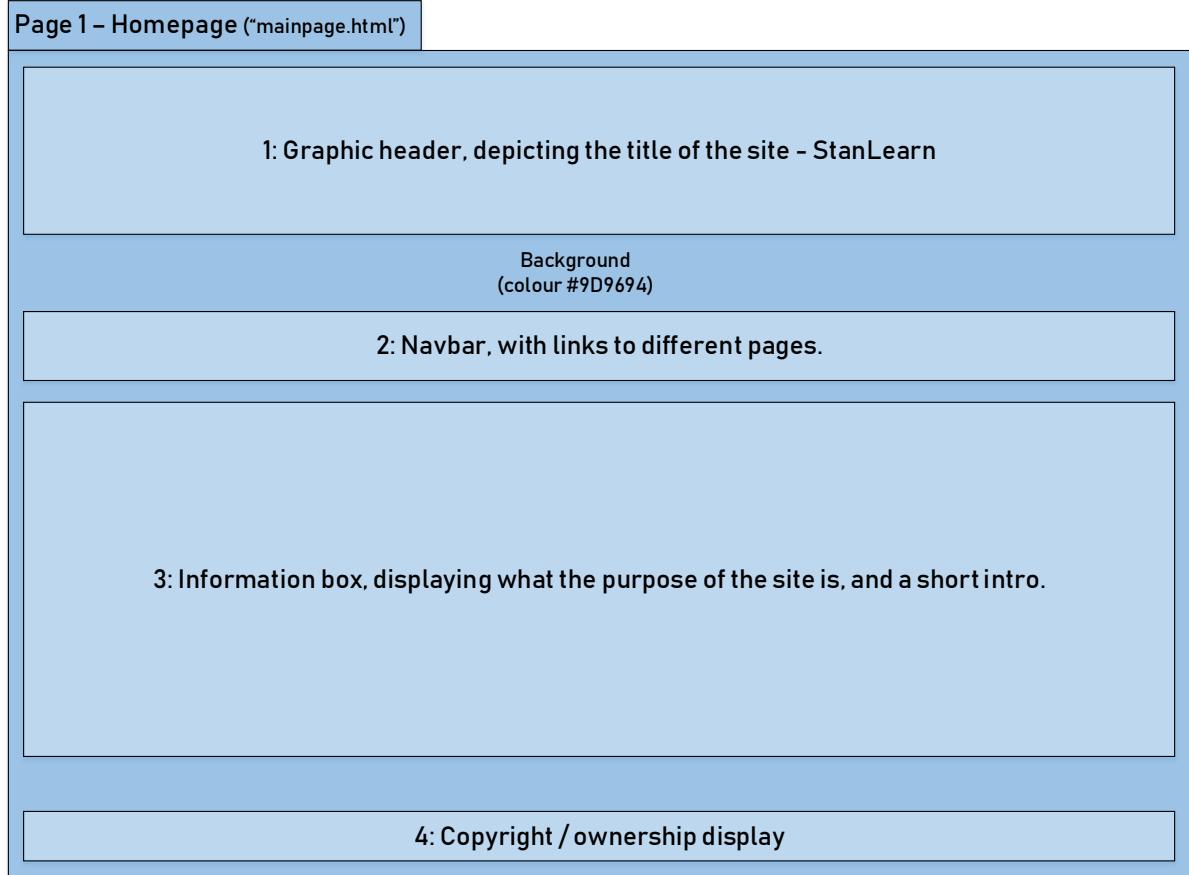


Figure 42 - Visio created object depicting site early designs.

1. Header

Height: 286px (100%), Width: 1333px (100%), (Will contain graphic image that adjusts to screen size.) Header graphic will link to homepage.

2. Navbar

Height: 19px (100%), Width: 1313px (100%), Padding: 20px 10px 20px 10px, Colour: solid black, Border: 1px solid black, Text-align: center, Background colour: #B0D7D7, Font-family: Bahnschrift, sans-serif.

3. Information Box

Height: 280px (100%), Width: 1313px (100%), Padding: 20px 10px 20px 10px, colour: solid black, border: 1px solid black, text-align: left, background-colour: #B0D7D7, font-size: 18px, Font-family: Bahnschrift Light, sans-serif.

4. Copyright / Ownership Display

Height: 28px (100%), Width: 98%, Font size: 15px, Position: Fixed, Bottom: 0%, Border: 1px solid black, Background colour: #B0D7D7, Padding 10px 0px 10px 10px, Font-family: Bahnschrift, sans-serif.

Graphic Header Design



Figure 43 - Image of site header.

The design of the graphic header is simplistic in design, which is good because it is one of the main themes I was aiming for. By using colours that are simple, gender-neutral and eye-catching but not too bright, I hope to set and maintain the atmosphere that the site is for all ages and sexes. The graphic header is informative too, as it displays instantly and without words which languages can currently be learned on the site.

7.2.2 About

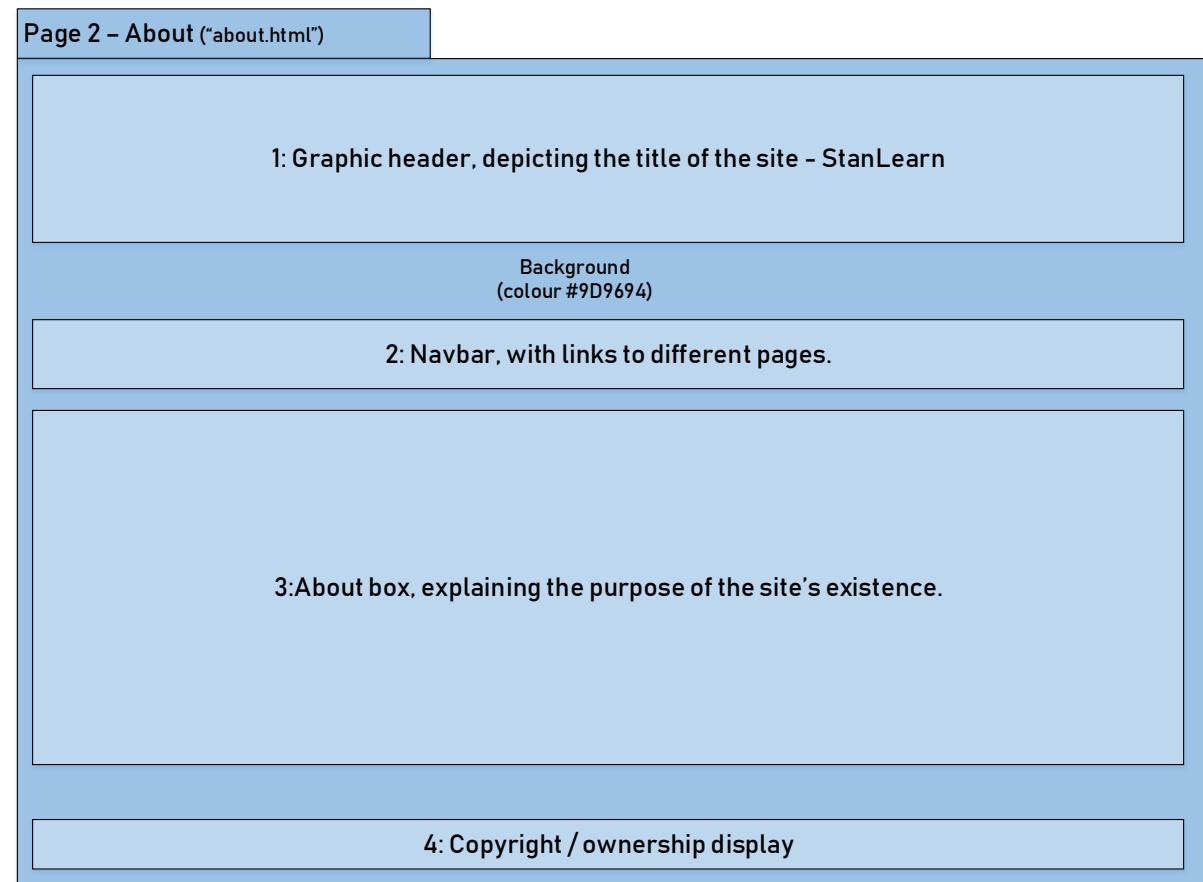


Figure 44 - Visio created object depicting site early designs.

1. Header

Height: 286px (100%), Width: 1333px (100%), (Will contain graphic image that adjusts to screen size.) Header graphic will link to homepage.

2. Navbar

Height: 19px (100%), Width: 1313px (100%), Padding: 20px 10px 20px 10px, Colour: solid black, Border: 1px solid black, Text-align: center, Background colour: #B0D7D7, Font-family: Bahnschrift, sans-serif.

3. About Box

Height: 91px (100%), Width: 1313px (100%), Padding: 20px 10px 20px 10px, colour: solid black, border: 1px solid black, text-align: left, background-colour: #B0D7D7, font-size: 18px, Font-family: Bahnschrift Light, sans-serif.

4. Copyright / Ownership Display

Height: 28px (100%), Width: 98%, Font size: 15px, Position: Fixed, Bottom: 0%, Border: 1px solid black, Background colour: #B0D7D7, Padding 10px 0px 10px 10px, Font-family: Bahnschrift, sans-serif.

7.2.3 Contact

Page 3 – Contact ("contact.html")

1: Graphic header, depicting the title of the site - StanLearn

Background
(colour #9D9694)

2: Navbar, with links to different pages.

3: Contact box, giving contact details for users to report problems like broken / missing content, or to suggest new languages for implementation.

5: Image with office
"location".

4: Copyright / ownership display

Figure 45 - Visio created object depicting site early designs.

1. Header

Height: 286px (100%), Width: 1333px (100%), (Will contain graphic image that adjusts to screen size.) Header graphic will link to homepage.

2. Navbar

Height: 19px (100%), Width: 1313px (100%), Padding: 20px 10px 20px 10px, Colour: solid black, Border: 1px solid black, Text-align: center, Background colour: #B0D7D7, Font-family: Bahnschrift, sans-serif.

3. About Box

Height: 91px (100%), Width: 1313px (100%), Padding: 20px 10px 20px 10px, colour: solid black, border: 1px solid black, text-align: left, background-colour: #B0D7D7, font-size: 18px, Font-family: Bahnschrift Light, sans-serif.

4. Copyright / Ownership Display

Height: 28px (100%), Width: 98%, Font size: 15px, Position: Fixed, Bottom: 0%, Border: 1px solid black, Background colour: #B0D7D7, Padding 10px 0px 10px 10px, Font-family: Bahnschrift, sans-serif.

5. Image

Height: 600px, Width: 600px.

“Office Location” Image

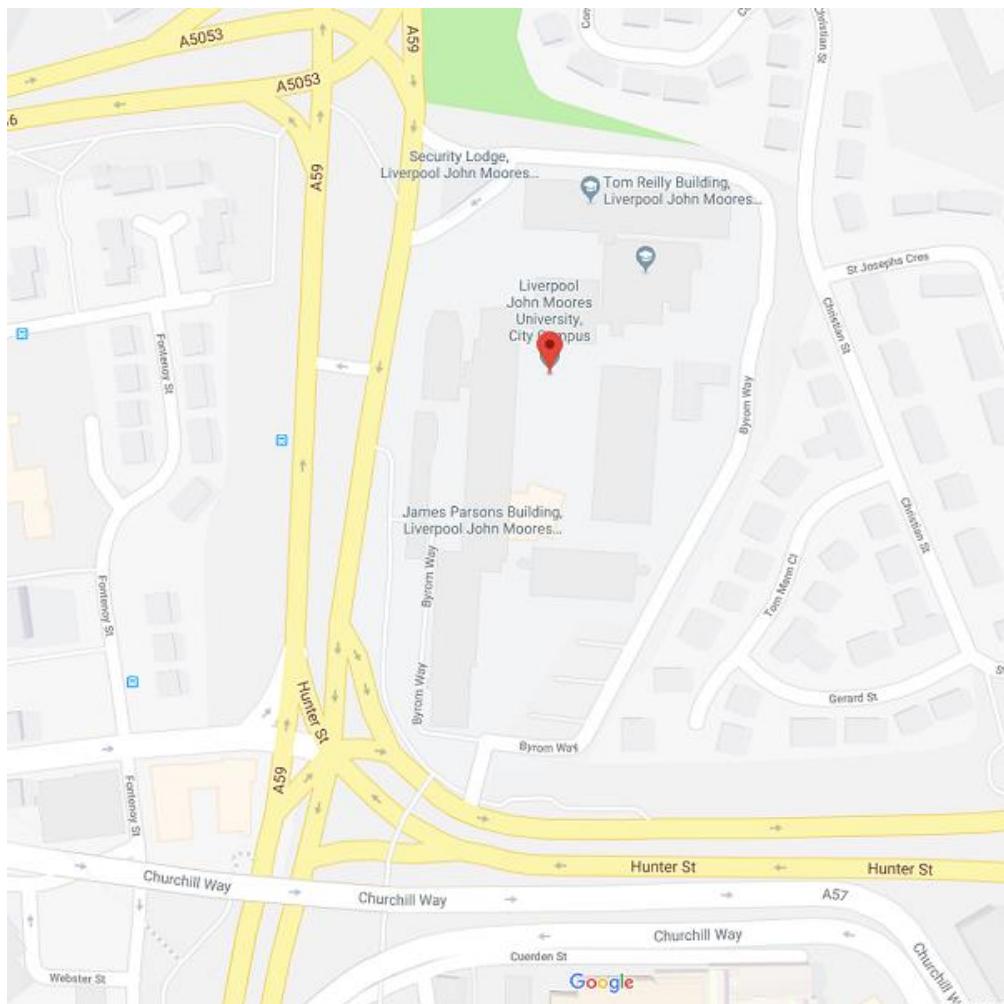


Figure 46 - Image used in the site as the StanLearn "office location".

The aforementioned “office location”, is simply a google maps image pointing to Liverpool John Moores University.

7.2.4 Swedish

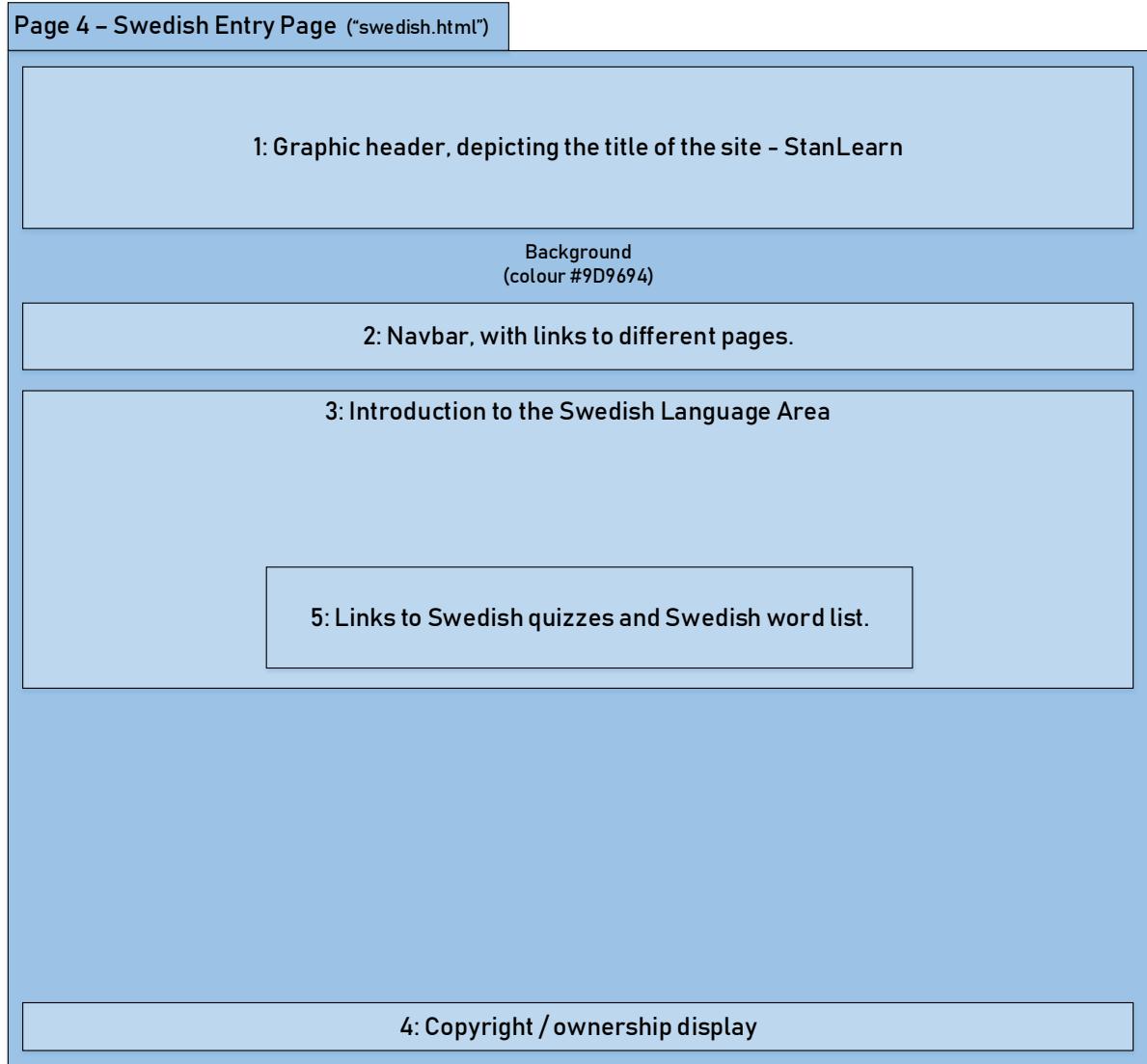


Figure 47 - Visio created object depicting site early designs.

1. Header

Height: 286px (100%), Width: 1333px (100%), (Will contain graphic image that adjusts to screen size.) Header graphic will link to homepage.

2. Navbar

Height: 19px (100%), Width: 1313px (100%), Padding: 20px 10px 20px 10px, Colour: solid black, Border: 1px solid black, Text-align: center, Background colour: #B0D7D7, Font-family: Bahnschrift, sans-serif.

3. Swedish Language Area Intro

Height: 91px (100%), Width: 1313px (100%), Padding: 20px 10px 20px 10px, colour: solid black, Border: 1px solid black, Text-align: center, Background-colour: #B0D7D7, Font-size: 19px, Background-image: url("swedishlanguagearea.png").

4. Copyright / Ownership Display

Height: 28px (100%), Width: 98%, Font size: 15px, Position: Fixed, Bottom: 0%, Border: 1px solid black, Background colour: #B0D7D7, Padding 10px 0px 10px 10px, Font-family: Bahnschrift, sans-serif.

5. Quiz and Wordlist Links

Height: 193px (100%), Width: 1313px (100%).

7.2.5 French

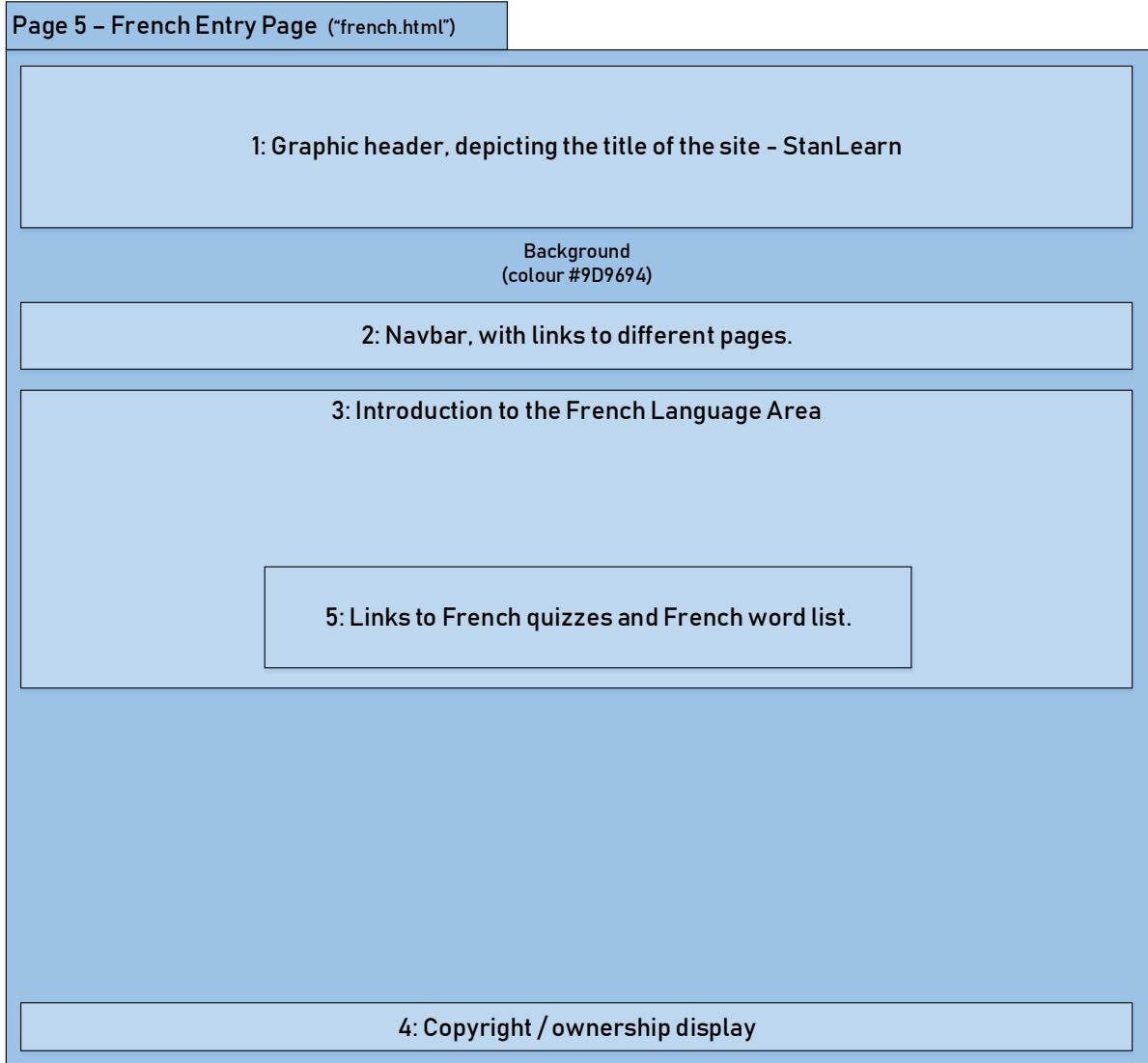


Figure 48 - Visio created object depicting site early designs.

1. Header

Height: 286px (100%), Width: 1333px (100%), (Will contain graphic image that adjusts to screen size.) Header graphic will link to homepage.

2. Navbar

Height: 19px (100%), Width: 1313px (100%), Padding: 20px 10px 20px 10px, Colour: solid black, Border: 1px solid black, Text-align: center, Background colour: #B0D7D7, Font-family: Bahnschrift, sans-serif.

3. French Language Area Intro

Height: 91px (100%), Width: 1313px (100%), Padding: 20px 10px 20px 10px, colour: solid black, Border: 1px solid black, Text-align: center, Background-colour: #B0D7D7, Font-size: 19px, Background-image: url("frenchlanguagearea.png").

4. Copyright / Ownership Display

Height: 28px (100%), Width: 98%, Font size: 15px, Position: Fixed, Bottom: 0%, Border: 1px solid black, Background colour: #B0D7D7, Padding 10px 0px 10px 10px, Font-family: Bahnschrift, sans-serif.

5. Quiz and Wordlist Links

Height: 193px (100%), Width: 1313px (100%).

7.2.6 Spanish

Page 6 – Spanish Entry Page (“spanish.html”)

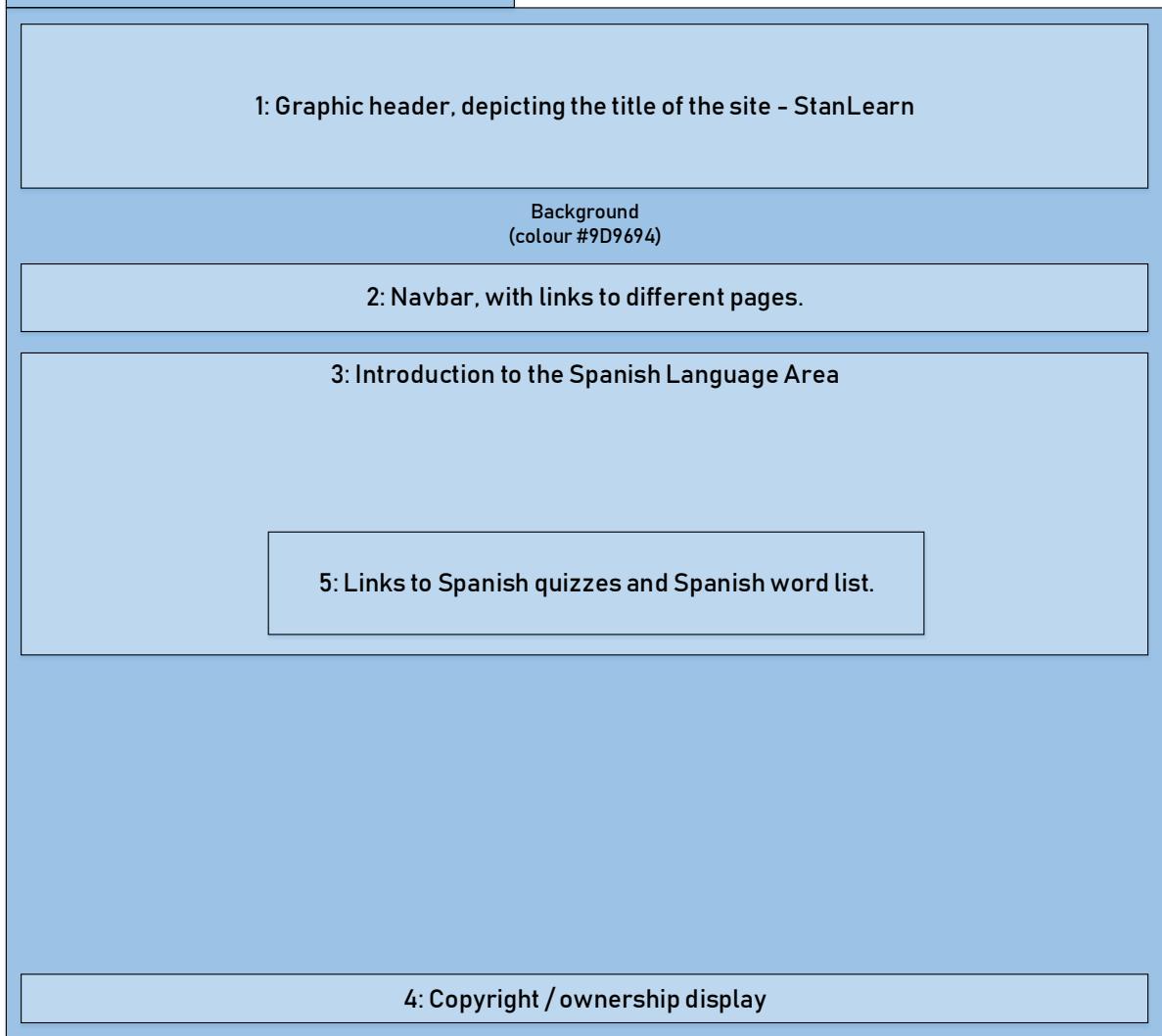


Figure 49 - Visio created object depicting site early designs.

1. Header

Height: 286px (100%), Width: 1333px (100%), (Will contain graphic image that adjusts to screen size.) Header graphic will link to homepage.

2. Navbar

Height: 19px (100%), Width: 1313px (100%), Padding: 20px 10px 20px 10px, Colour: solid black, Border: 1px solid black, Text-align: center, Background colour: #B0D7D7, Font-family: Bahnschrift, sans-serif.

3. Spanish Language Area Intro

Height: 91px (100%), Width: 1313px (100%), Padding: 20px 10px 20px 10px, colour: solid black, Border: 1px solid black, Text-align: center, Background-colour: #B0D7D7, Font-size: 19px, Background-image: url("spanishlanguagearea.png").

4. Copyright / Ownership Display

Height: 28px (100%), Width: 98%, Font size: 15px, Position: Fixed, Bottom: 0%, Border: 1px solid black, Background colour: #B0D7D7, Padding 10px 0px 10px 10px, Font-family: Bahnschrift, sans-serif.

5. Quiz and Wordlist Links

Height: 193px (100%), Width: 1313px (100%).

7.2.7 German

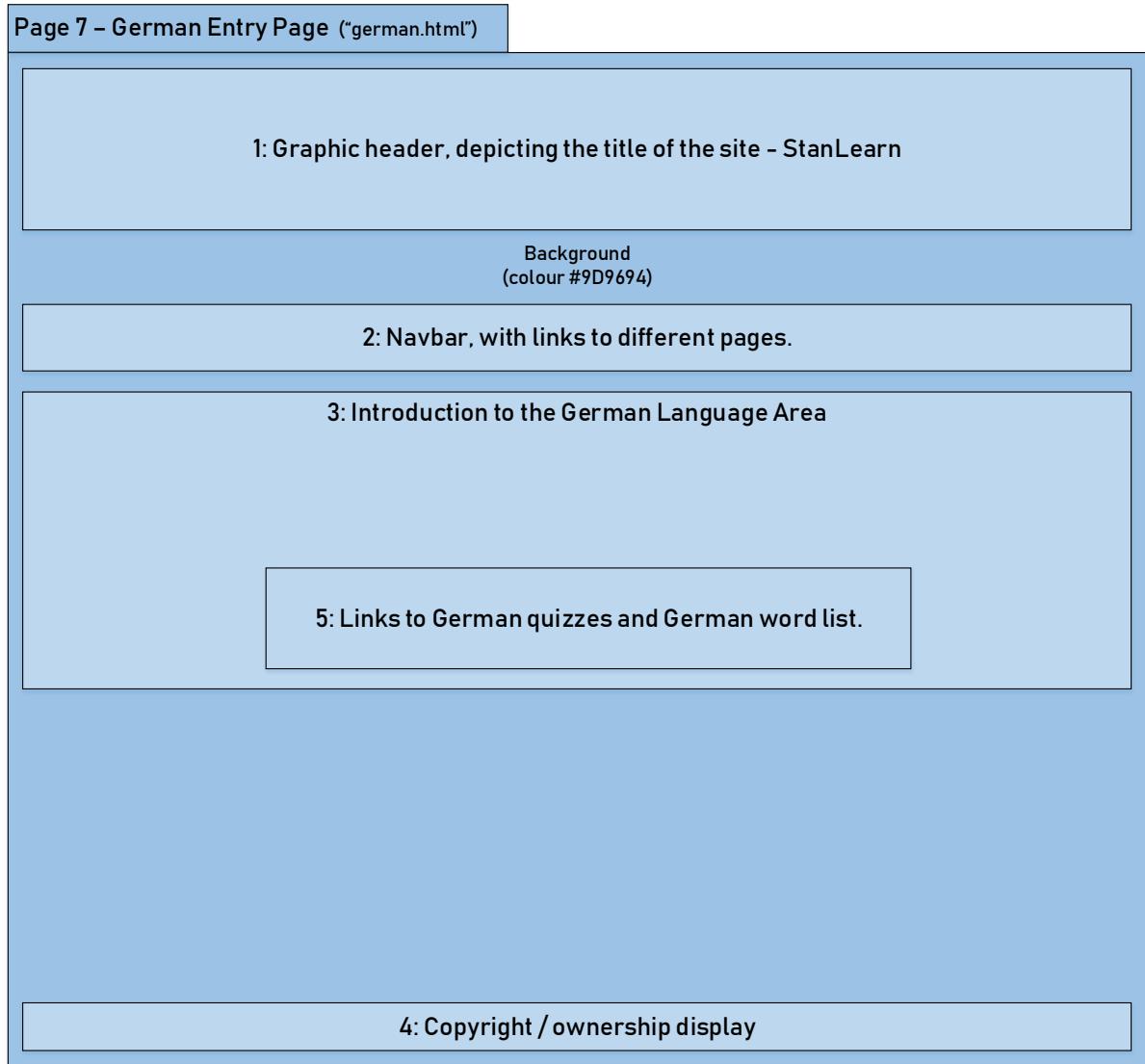


Figure 50 - Visio created object depicting site early designs.

1. Header

Height: 286px (100%), Width: 1333px (100%), (Will contain graphic image that adjusts to screen size.) Header graphic will link to homepage.

2. Navbar

Height: 19px (100%), Width: 1313px (100%), Padding: 20px 10px 20px 10px, Colour: solid black, Border: 1px solid black, Text-align: center, Background colour: #B0D7D7, Font-family: Bahnschrift, sans-serif.

3. German Language Area Intro

Height: 91px (100%), Width: 1313px (100%), Padding: 20px 10px 20px 10px, colour: solid black, Border: 1px solid black, Text-align: center, Background-colour: #B0D7D7, Font-size: 19px, Background-image: url("germanlanguagearea.png").

4. Copyright / Ownership Display

Height: 28px (100%), Width: 98%, Font size: 15px, Position: Fixed, Bottom: 0%, Border: 1px solid black, Background colour: #B0D7D7, Padding 10px 0px 10px 10px, Font-family: Bahnschrift, sans-serif.

5. Quiz and Wordlist Links

Height: 193px (100%), Width: 1313px (100%).

7.2.8 Quiz Pages

The quiz pages will each have their own individual CSS files for styling, but because they will be identical, for the sake of saving space I will condense them into one diagram. Just replace “language” with Swedish, French, Spanish or German.

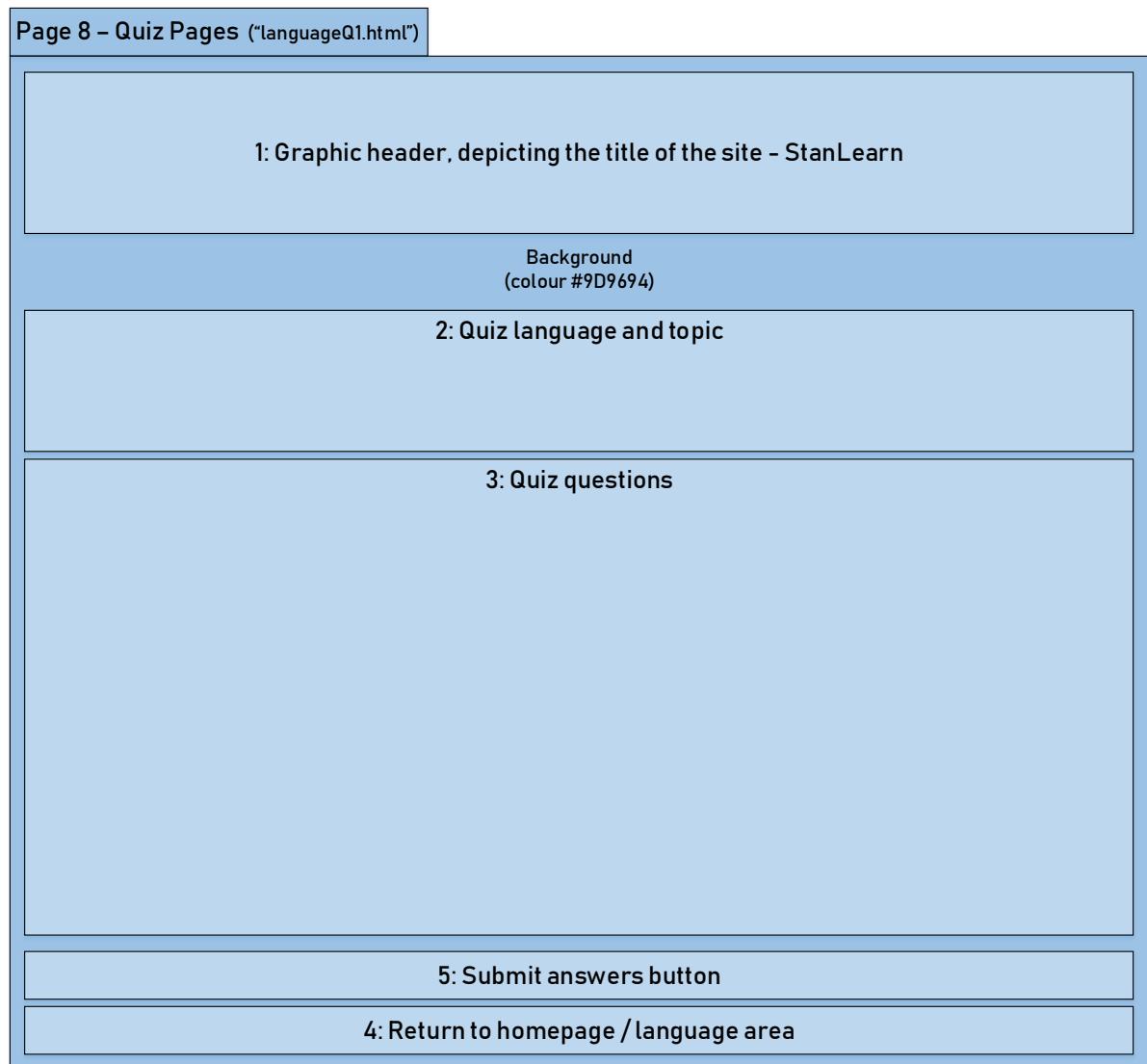


Figure 51 - Visio created object depicting site early designs.

1. Header

Height: 286px (100%), Width: 1333px (100%), (Will contain graphic image that adjusts to screen size.) Header graphic will link to homepage.

2. Quiz Language and Topic

Height: 170px (100%), Width: 1296px (100%), Padding: 20px 10px 20px 10px, Colour: solid black, Border: 1px solid black, Text-align: center, Background colour: #B0D7D7, Font size: 19px.

3. Quiz Questions

Height: 525px (100%), Width: 1316px (100%), Font-family: Bahnschrift Light, sans-serif, Colour: solid black, Text-align: left, Vertical-align: middle, Border: 1px solid black, Font-size: 17px, Letter-spacing: 1px.

4. Return to Homepage / Language Area

Height: 28px (100%), Width: 98%, Font size: 15px, Position: Fixed, Bottom: 0%, Border: 1px solid black, Background colour: #B0D7D7, Padding 10px 0px 10px 10px, Font-family: Bahnschrift, sans-serif.

5. Submit Answers Button

Height: 24px (100%), Width: 1296px (100%).

7.2.9 Word Lists

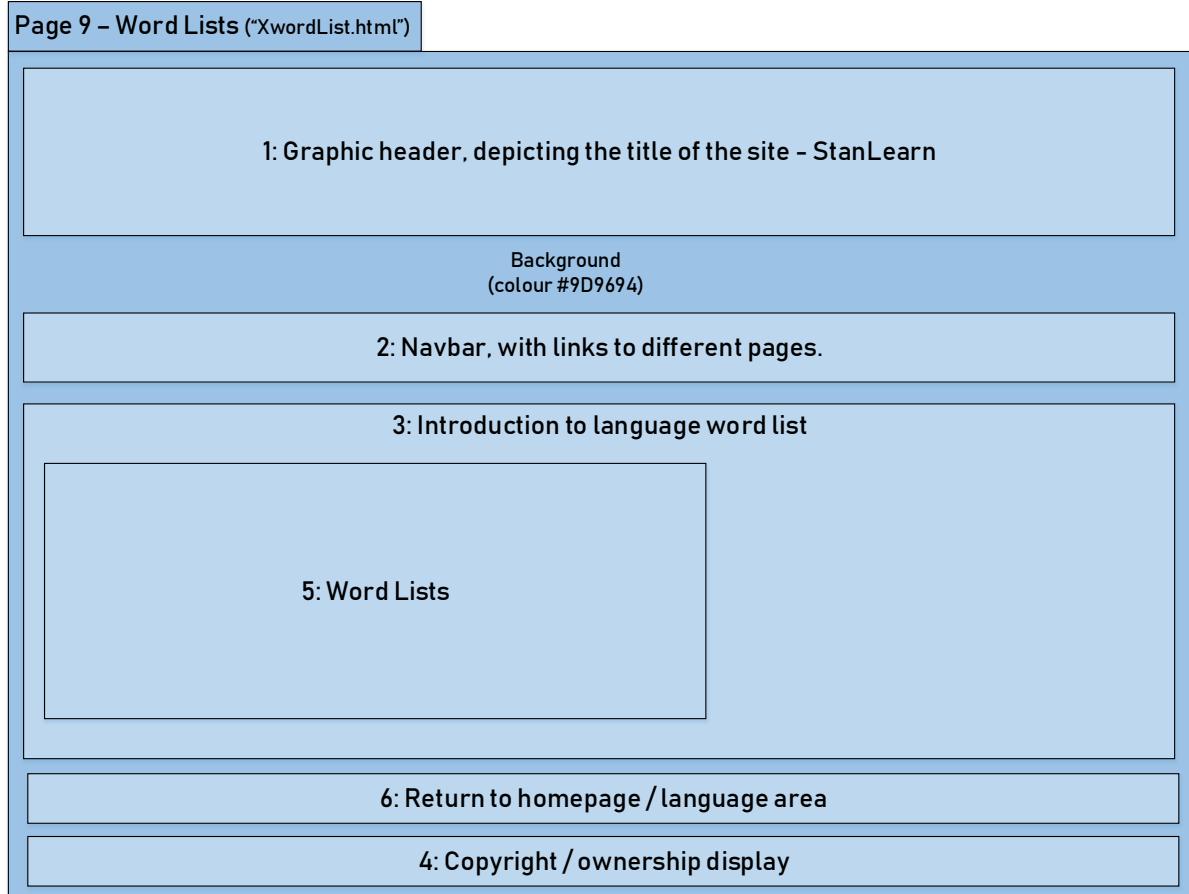


Figure 52 - Visio created object depicting site early designs.

1. Header

Height: 286px (100%), Width: 1333px (100%), (Will contain graphic image that adjusts to screen size.) Header graphic will link to homepage.

2. Navbar

Height: 19px (100%), Width: 1313px (100%), Padding: 20px 10px 20px 10px, Colour: solid black, Border: 1px solid black, Text-align: center, Background colour: #B0D7D7, Font-family: Bahnschrift, sans-serif.

3. Introduction to Language Word List

Height: 850px, Width: 1313px (100%), Padding: 20px 10px 20px 10px, Colour: solid black, Border: 1px solid black, Text-align: left, Background-colour: #B0D7D7, Font-size: 18px.

4. Copyright / Ownership Display

Height: 28px (100%), Width: 98%, Font size: 15px, Position: Fixed, Bottom: 0%, Border: 1px solid black, Background colour: #B0D7D7, Padding 10px 0px 10px 10px, Font-family: Bahnschrift, sans-serif.

5. Word Lists

Height: 850px, Width: 1296px (100%).

6. Return to Homepage / Language Area

Height: 28px (100%), Width: 98%, Font size: 15px, Position: Fixed, Bottom: 0%, Border: 1px solid black, Background colour: #B0D7D7, Padding 10px 0px 10px 10px, Font-family: Bahnschrift, sans-serif.

7.3 Activity UML Diagram

This UML diagram should assist in explaining how the process behind the user answer submission works.

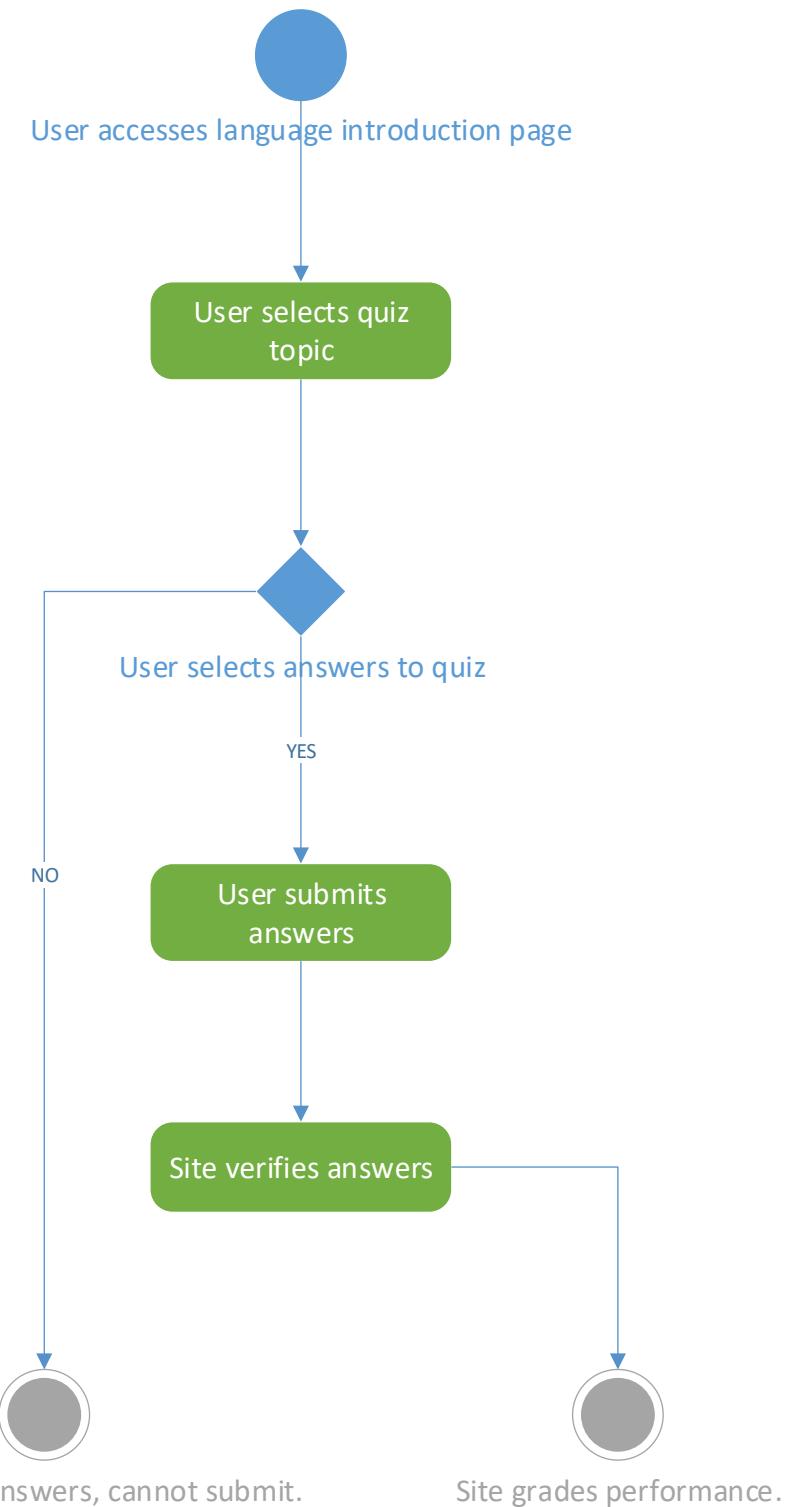


Figure 53 - UML activity diagram.

Chapter 8. Development of the Language Learning Quiz

This chapter focuses on the development of the quiz. It delves into precisely how the quiz works, and how the answers the user gives are checked and confirmed.

8.1 How Will the Quiz Work?

The quiz process itself will work by users visiting the language area for the language they want to learn, then going to the wordlist to memorise the words in the sections they want to

test themselves in, and then navigating to the relevant quiz and testing themselves. For example, a user wanting to test their knowledge of animals in German would go to the German language area, then the wordlist, then memorise the animals, then go back to the German language area and into the animals quiz.

The idea I had in mind before creating the quiz was to create one in HTML using radio buttons to answer multiple choice questions. I had the idea of full modularity, so I had to make sure questions and answers could easily be added, changed, or removed in the HTML, CSS and Java. In keeping of this theme of modularity, every topic in every language area has its own HTML page, under the header “languageQ1”, with Q2, Q3 and Q4 also being created to represent the animals, numbers, colours and food quizzes respectively.

The work on the HTML and CSS was done using Notepad++. This software was chosen because I have years of experience with it from my three years at University. Other software considered can be found in Chapter 9 subsection 9.1, along with justification for choosing Notepad++ in Chapter 9, subsection 9.2.

The quiz will work by having the user submit an answer via a radio button. The JavaScript will read the answers submitted by the user and check if they are the same as the answers it has stored inside it. If the user has gotten any questions wrong, it will push forward a div that tells them which question they got wrong. For this to happen, there will be 5 divs in the HTML created, divs 1-4 will contain the correct question answer, and div 5 will display a message telling the user they got all the answers correct. By using the CSS, divs 1-5 will be hidden until the JavaScript tells them to display. The JavaScript will tell divs to display based on submitted user answers.

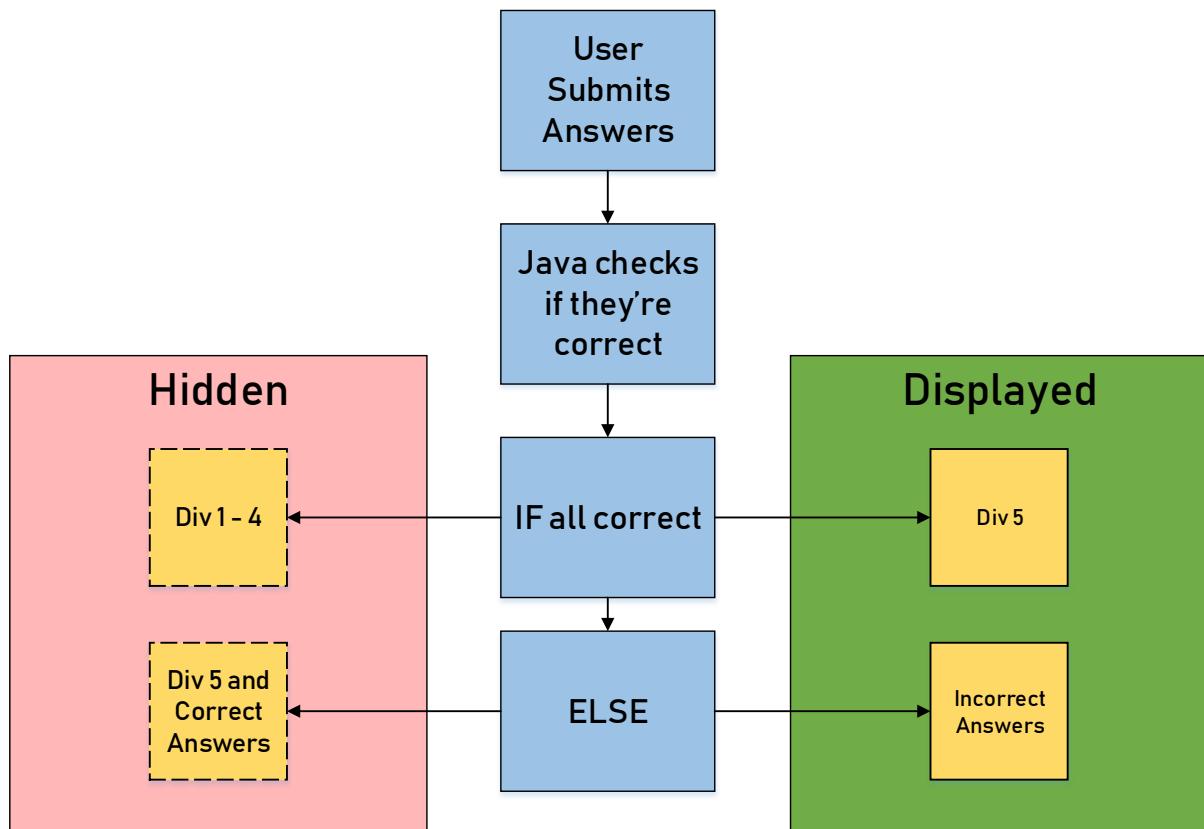


Figure 54 - Diagram depicting how the site will display user results.

The diagram above explains when divs will be shown, and when they will stay hidden. For example, if the user gets all answers incorrect, divs 1-4 will display, telling them what the correct answers were. If the user gets all answers correct, only the one div will display that tells the user they have gotten all answers correct, instead of 4 individual divs.

8.2 HTML Base

Before developing the quiz with Java, I had to create the basic design and layout so I knew what to do. I created a rough layout of how the quiz would work with Visio so I had a visual aid to write the HTML from.

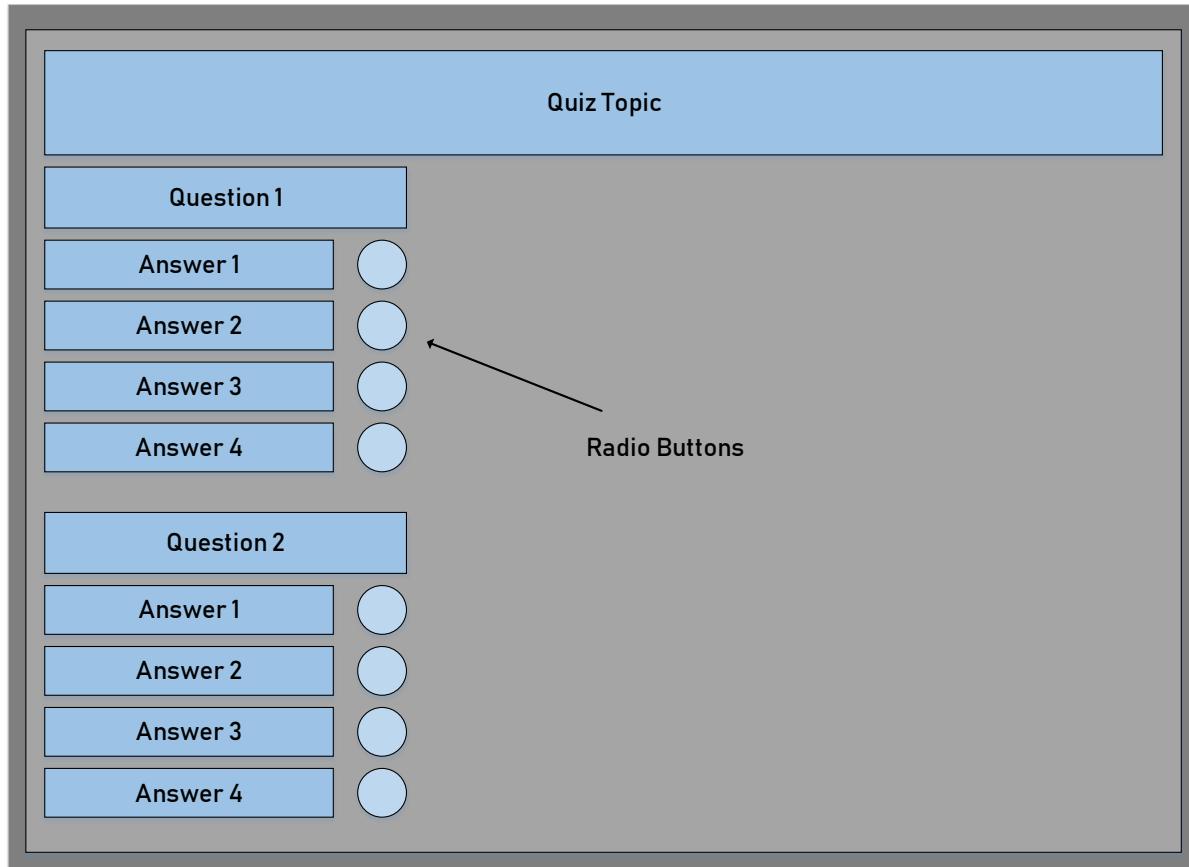


Figure 55 - Visio object depicting final design of quiz pages.

The example above shows a very early version that the final design was based off. With this early design in mind, I could easily create a modular HTML framework that could be built on later.

To begin the layout of the page, I included the name of the quiz in a header, along with a short line to explain to a user what quiz they are on and in what language. These elements were placed inside a div known as the "main_body".

```
<div id="main_body">
  <h2> Swedish Quiz 1 - Animals </h2>
  <br><br>
  <p> Welcome to the Swedish animals quiz! A sentence will be given with a blank, and you must select the correct answer. </p>
</div>
```

Figure 56 - Code snippet depicting the header and intro to the language learning page for Swedish.

To start laying out the questions and answers, I created one single div (with the div id "questions") that would hold the question with the use of a paragraph (), and then a

list tag (` `) to hold the radio buttons and the answers alongside them. I used labels to put the answers in the lists, and set the input type to radio to make users able to see multiple answers but only select one. This can all be seen below.

```
<div id="questions">
<p class="question"><b> Question 1. What is the Swedish word for "animal"? . </b></p>
<ul class="answers">
<label>Djur.</label>
<input type="radio" name="swedques1" value="1"><br>
<label>Katt.</label>
<input type="radio" name="swedques1" value="2"><br>
<label>Hund.</label>
<input type="radio" name="swedques1" value="3"><br>
<label>Fågel.</label>
<input type="radio" name="swedques1" value="4"><br>
</ul>

<p class="question"><b> Question 2. What is the Swedish word for "cat"? . </b></p>
<ul class="answers">
<label>Mus.</label>
<input type="radio" name="swedques2" value="1"><br>
<label>Häst.</label>
<input type="radio" name="swedques2" value="2"><br>
<label>Spindel.</label>
<input type="radio" name="swedques2" value="3"><br>
<label>Katt.</label>
<input type="radio" name="swedques2" value="4"><br>
</ul>

<p class="question"><b> Question 3. What is the Swedish word for "the moose"? . </b></p>
<ul class="answers">
<label>Älgen.</label>
<input type="radio" name="swedques3" value="1"><br>
<label>Hundarna.</label>
<input type="radio" name="swedques3" value="2"><br>
<label>Hästen.</label>
<input type="radio" name="swedques3" value="3"><br>
<label>Björnen.</label>
<input type="radio" name="swedques3" value="4"><br>
</ul>

<p class="question"><b> Question 4. What is the Swedish word for "bird"? . </b></p>
<ul class="answers">
<label>Djur.</label>
<input type="radio" name="swedques4" value="1"><br>
<label>Katt.</label>
<input type="radio" name="swedques4" value="2"><br>
<label>Fågel.</label>
<input type="radio" name="swedques4" value="3"><br>
<label>Hund.</label>
<input type="radio" name="swedques4" value="4"><br>
</ul>
<br>
</div>
```

Figure 57 - Code snippet depicting the question in the Swedish animals quiz.

Instead of using extensive formatting techniques in the CSS relating to padding, simple line breaks (`
`) were used to space lines out from each other in the HTML, and make the questions a bit clearer. As is visible in these lines of code, the questions and the answers are totally modular, meaning that questions can be added, removed and edited at the developers will, as long as the Java is edited appropriately later. In addition to this, extra answers can be added into existing questions, again, if the Java is edited appropriately with more values being added. For example, to add more questions I simply copied the first set of questions and pasted them four times to get all the questions, and changed the “name” attribute to having a different number on the end.

Initially, I was going to name the questions different names depending on the language, however in the end I simply left it as “swedques” which is shorthand for “Swedish question”, for every language. The “value” setting is simply the number of the answer, and is quoted by the Java to decide which answer is correct.

To submit the answers, a div was created named “userResults” which would simply serve as a hover button.

```
<div id="userResults">  
    Submit answers!  
</div>
```

Figure 58 - Image depicting the submit button.

So that the user knows when they got an answer incorrect, or when they get all answers correct, I have included 5 divs at the bottom that contain the answers to questions one to four, and the fifth div tells them they have got all answers correct.

```
<div id="answer1">  
    <p> Question 1: The correct answer was 1 - Djur. </p><br>  
</div>  
<div id="answer2">  
    <p> Question 2: The correct answer was 4 - Katt. </p><br>  
</div>  
<div id="answer3">  
    <p> Question 3: The correct answer was 1 - Älgen. </p><br>  
</div>  
<div id="answer4">  
    <p> Question 4: The correct answer was 3 - Fågel. </p><br>  
</div>  
<div id="answer5">  
    <p> You answered all Swedish animal questions answered correctly, well done! </p><br>  
</div>
```

Figure 59 - Code snippet depicting correct answers that display upon submitting answers.

These divs are not displayed until the user submits their answers, and unless they are relevant. For example, if a user gets all questions beside question 3 correct, only div “answer3” will appear. Once the user gets all answers correct, div “answer5” appears. I considered naming the last div something along the lines of “answerVerifier” similar to what it is called in the Java, but decided to keep its name as shown as it reduced confusion when styling.

8.3 CSS Styling

After the HTML framework was created, I made use of Cascading Styling Sheets (CSS) files to assist in styling my pages and adding content to them.

Using CSS allowed me to perform more complex features such as hiding divs that did not need to be seen until such a time when they are called. It also allowed me to create borders on certain interesting items, change the way text appears with the use of new fonts and bolding, and insert background images to fulfil stylistic desires.

To start, I styled the #main_body div. This div contains the topic of the quiz, as well as the language the quiz is testing the user in. To make this clear, I created small graphics of flags as a background image and inserted it with the CSS. I also changed the font size to make it more readable, as well as adding padding around it to separate it from the header and

questions. Since the size of the div has been set with the CSS and the line breaks in the HTML, the image will stay fitted.

```
#main_body {  
    padding: 20px 10px 20px 10px;  
    font-size: 19px;  
    color: solid black;  
    border: 1px solid black;  
    text-align: center;  
    background-color: #B0D7D7;  
    background-image: url("swedishlanguagearea.png");  
}
```

Figure 60 - CSS code snippet.

Next, the questions were styled. The questions were easy to style, as they did not require much interference. A different font style was used to increase readability and to improve the aesthetic look of the site, as well as having the font size itself changed so users could tell they were on the correct section.

```
#questions {  
    font-family: Bahnschrift Light, sans-serif;  
    color: solid black;  
    text-align: left;  
    vertical-align: middle;  
    border: 1px solid black;  
    font-size: 17px;  
    letter-spacing: 1px;  
    background-color: #B0D7D7;  
}
```

Figure 61 - CSS code snippet.

After this came the styling of the user submitting results button. To make it clear to the user that this is a button to submit the answers, I made use of the hover function, so that the cursor changes shape when the button is highlighted, as well as having it change colour. The user results div is called twice, as it must be styled in two different states – one when the user is hovering over it, and one when the user is not. Comments were added to assist myself, so I would remember which elements to manipulate when making minute adjustments in future.

```
#userResults {  
    text-align: center; /* this was done to center the text, when the button is out of hover mode. */  
    background-color: #9D9694;  
    border: 1px solid black;  
    padding: 10px 10px 10px 10px;  
}  
#userResults:hover {  
    background-color: #B0D7D7;  
    text-align: center;  
    cursor: pointer; /* sets the pointer to be a cursor, so the user knows it's clickable */  
    border: 1px solid black;  
    padding: 10px 10px 10px 10px;  
}
```

Figure 62 - CSS code snippet.

Next, the answer divs were styled. These are important to the quiz, as they tell the user where they went wrong. However, if these divs appear too early or not at all, they will prove detrimental to the user and either tell them the answer prematurely, or not help them at all.

The image below will show how I told the divs not to display themselves, and how I had to explicitly set these divs to not have a border, as the setting in an earlier div applied borders to all divs. The image below will also show the “closeAnswers” div, which is also responsible for hiding the answers, which also possesses properties which hide its borders, and also the div entirely.

```
#answer1, #answer2, #answer3, #answer4, #answer5 {
    display: none;
    border: none;
}
#closeAnswers {
    display: none;
    border: none;
}
```

Figure 63 - CSS code snippet.

Across all of this styling, I tried to make sure the text was of appropriate font, and of appropriate size to draw attention to where it was needed. I also made sure the colour scheme stayed consistent, as well as making use of paddings to space out the divs from each other to create and maintain the professional atmosphere I was aiming to create.

8.4 Java

Finally, I had to create and insert the JavaScript into the HTML framework that had been styled by the CSS for users to be able to submit the answers they picked. This was clearly the most important part, but luckily as the HTML and CSS had been purpose-built to include this Java code, the migration process was fairly simple. This subsection will talk about how the Java was created, how it functions, and how it interacts with the page. The code in the screenshots has been commented on by myself, to explain what each part is doing in case I became confused when revisiting later.

The first step was to create a name for the function, in this case it has been named “userResults”, coinciding with the name of the div that the users press to submit their results. The code functions when the user clicks the submit answers button on the website, represented by the “.on(“click”). In addition to the function creating and type, the first part of the verification code is present. What these lines of code do is check each one of the user inputs, to see if any are missing. The user will not be able to submit if there are inputs missing.

```
1 $(document).ready(function()
2 {   $("#userResults").on("click", function() {
3     /* above signifies the beginning of and names of
4     if (!$("input[name=swedques1]:checked").val() ||
5     !$("input[name=swedques2]:checked").val() || /* t
6     !$("input[name=swedques3]:checked").val() ||
7     !$("input[name=swedques4]:checked").val()
```

Figure 64 - Java code snippet.

Next, was the creation of variables. The “quesname” and “quesverifiername” values simply link to their answer number. For example, all questions will have 4 answers, and each of these answers will have a value that the code reads.

Question 1	Name and Value
Answer 1	swedques1, 1
Answer 2	swedques1, 2
Answer 3	swedques1, 3
Answer 4	swedques1, 4
Question 2	Name and Value
Answer 1	swedques2, 1
Answer 2	swedques2, 2
Answer 3	swedques2, 3
Answer 4	swedques2, 4
Question 3	Name and Value
Answer 1	swedques3, 1
Answer 2	swedques3, 2
Answer 3	swedques3, 3
Answer 4	swedques3, 4
Question 4	Name and Value
Answer 1	swedques4, 1
Answer 2	swedques4, 2
Answer 3	swedques4, 3
Answer 4	swedques4, 4

Figure 65 - Table depicting how the Java reads the questions in the HTML.

There are also the variables that check whether questions 1-4 have been answered correctly. The numbers stored at the end of the four “quesx” strings are the correct answers. So, for example, in the first Swedish quiz, the answers are the 1, 4, 1, 3 for each question respectively. The final line is the quesverifier, which checks all the answers given from left to right by using the && logical operator. If a question is wrong, the correct answer will be pushed forward (as shown in a later image). If all questions are correct, none of the lines will be shown that say a user has got a question wrong. Also included is the creation of an alert, which triggers if the user has not selected an answer for one or more questions.

```

8 } {
9 alert("You have not answered all of the questions!"); /* this line simply adds
10 }
11 else {
12 var ques1name = "1"; /* these lines will be the lines that display if a user
13 var ques2name = "2";
14 var ques3name = "3";
15 var ques4name = "4";
16 var quesverifiername = "5"; /* this line is the question verifier. this line
17 var ques1 = ($("#input[name=swedques1]:checked").val() != "1"); /* these lines
18 var ques2 = ($("#input[name=swedques2]:checked").val() != "4");
19 var ques3 = ($("#input[name=swedques3]:checked").val() != "1");
20 var ques4 = ($("#input[name=swedques4]:checked").val() != "3");
21 var quesverifier = (!ques1 && !ques2 && !ques3 && !ques4); var answers = [];

```

Figure 66 - Java code snippet.

The lines shown below are the lines responsible for pushing the correct answer forward if the user selected and submitted the wrong answer. It takes the answers submitted from the quesverifier which has confirmed they are wrong, and pushes out the correct answer.

```
24
25 if (ques1) {answers.push(ques1name)};
26 if (ques2) {answers.push(ques2name)};|
27 if (ques3) {answers.push(ques3name)};
28 if (ques4) {answers.push(ques4name)};
29 if (quesverifier) {answers.push(ques1name)};
30
```

Figure 67 - Java code snippet.

These last few lines control what divs are shown after the user submits their answers. They correspond to the answer 1, 2, 3, 4, and 5 (5 displays if all questions are correct) divs in the HTML and CSS. These divs are hidden by default, and will only appear after the user has submitted their answers.

```
--|
33 if (ques1) ${"#answer1").show();};
34 if (ques2) ${"#answer2").show();};
35 if (ques3) ${"#answer3").show();};
36 if (ques4) ${"#answer4").show();};
37 if (quesverifier) ${"#answer5").show();}; /* "
38 ${"#closeAnswers").show();}; /* process for cl
```

Figure 68 - Java code snippet.

For example, if question 3 is wrong, the code shown in figure 67 will push ques3, and then the code in figure 68 will show the answer3 div, as it contains the correct answer.

For now, all of the answers to the quizzes are the same. This means that for every quiz on every topic on the website, every answer is 1, 4, 1 ,3. This however, can be easily changed, all I need to do is change which numbers are in the ques1 checker variables, and then save the edited java code as a new file with a new name, for example quizCorrect2. Then, I just need to change the line in the html that reads **<script src="quizCorrect.js"> </script>** to **<script src="quizCorrect2.js"> </script>**. This will then treat the quiz as having an entirely different set of answers and questions, and since all the quizzes are fully modular this can be done very quickly. Because this website is more of a proof of concept and a prototype, all the answers are the same but as stated above and shown by example making changes would take minutes.

8.4.1 Java Class Diagram

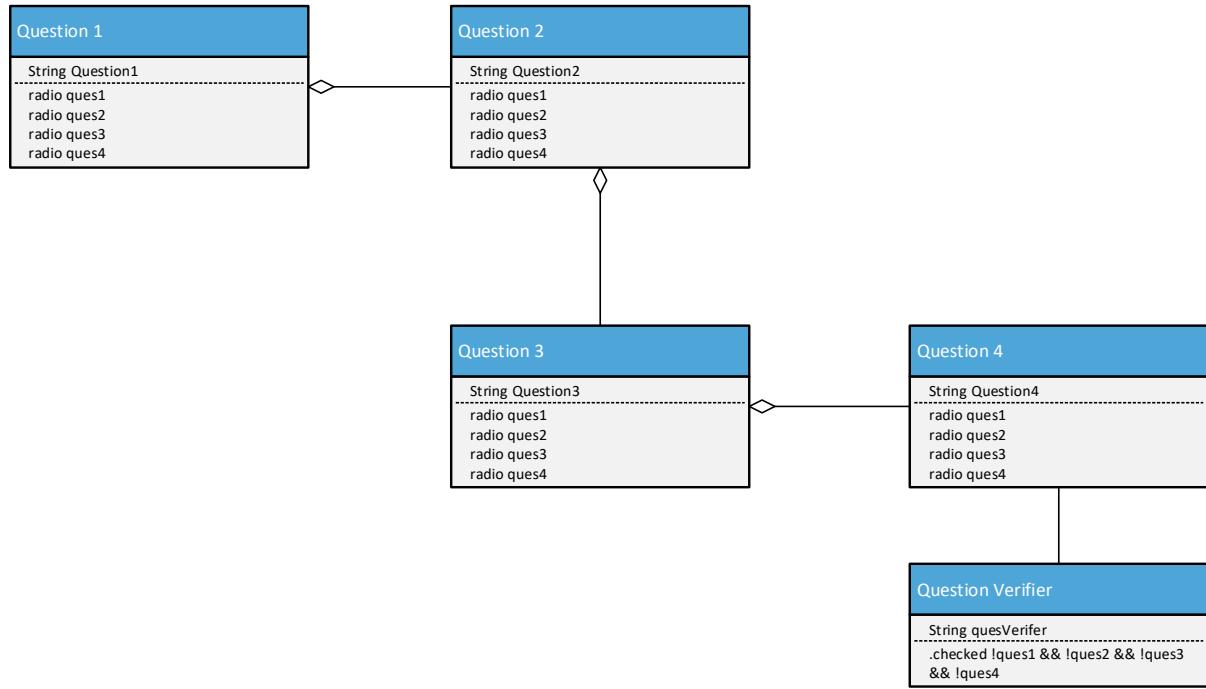


Figure 69 – UML Class Diagram.

Chapter 9. Implementation

The purpose of this chapter is to explain what software was used to create the website, why the selected software was used, and how the pages were designed.

9.1 Consideration of Software

Before creating my website, I had to decide which software I was going to use. I needed software that was capable of creating and manipulating HTML and CSS files, as well as being able to implement jQuery and JavaScript.

9.1.1 Notepad++

```
</header>
</div>
<div id="main_body">
    <h2> Swedish Quiz 1 - Animals </h2>
    <br><br>
    <p> Welcome to the Swedish animals quiz! A sentence will be given with a blank, and you must select the correct answer. <br><br>
</div>
<div id="questions">
    <p class="question"><b> Question 1. What is the Swedish word for "animal"? </b></p>
    <ul class="answers">
        <label>djur.</label>
        <input type="radio" name="swedques1" value="1"><br>
        <label>Katt.</label>
        <input type="radio" name="swedques1" value="2"><br>
        <label>Hund.</label>
        <input type="radio" name="swedques1" value="3"><br>
        <label>Fågel.</label>
        <input type="radio" name="swedques1" value="4"><br>
    </ul>
    <p class="question"><b> Question 2. What is the Swedish word for "cat"? </b></p>
    <ul class="answers">
        <label>Mus.</label>
        <input type="radio" name="swedques2" value="1"><br>
        <label>Häst.</label>
        <input type="radio" name="swedques2" value="2"><br>
        <label>Spindel.</label>
        <input type="radio" name="swedques2" value="3"><br>
        <label>Katt.</label>
        <input type="radio" name="swedques2" value="4"><br>
    </ul>
    <p class="question"><b> Question 3. What is the Swedish word for "the moose"? </b></p>
    <ul class="answers">
        <label>Algen.</label>
        <input type="radio" name="swedques3" value="1"><br>
        <label>Hundarna.</label>
        <input type="radio" name="swedques3" value="2"><br>
        <label>Hästen.</label>
        <input type="radio" name="swedques3" value="3"><br>
        <label>Björnen.</label>
        <input type="radio" name="swedques3" value="4"><br>
    </ul>
    <p class="question"><b> Question 4. What is the Swedish word for "bird"? </b></p>
    <ul class="answers">
        <label>Djur.</label>
        <input type="radio" name="swedques4" value="1"><br>
        <label>Katt.</label>
        <input type="radio" name="swedques4" value="2"><br>
        <label>Fågel.</label>
    </ul>
</div>
```

Figure 70 - Screenshot of Notepad++

First, I considered using Notepad++. I considered this software first because I have been using it throughout my time in university. I am familiar with its structure, its UI, its functions and what types of code it can interact with and create. I knew from experience that it was easily capable of creating and manipulating HTML files, as well as creating CSS files for it to interact with. I also knew that it could take these HTML and CSS files and easily have jQuery and JavaScript injected into them to create functional webpages.

It is regularly updated, fully supported, and most importantly its completely free. Its UI allows for editing of multiple files at once, which is especially useful for comparing documents and fine-tuning certain parts, or checking for discrepancies between the two if they share similar parts of code (which my project undoubtedly does). From experience, I can testify to having used the tabbed editing function and saved myself a lot of time and effort re-writing lines of code. Expanding on usability and user-friendly UI, Notepad++ features “syntax highlighting”, which is an automatic process that highlights and changes the font appearance of certain parts of code to assist the user. This assists the user with readability, and helps point out parts of code that aren’t working. For example, if the user has used a delimiter to contain text in a line of code, but one of these delimiters is missing, it will colour all text after the delimiter, which easily displays to the user that something is wrong or missing.

However, this software relies on donations to keep itself running, but while the concern of freeware suddenly ceasing being updated is always present, Notepad++ has been supported for 14 whole years and is still seeing regular updates. At the time of writing (April 2018), the last update to Notepad++ was March 19th, 2018. As well as this, the software itself is very much “entry level” and does not offer the usability benefits or additional functions that paid software would.

Advantages

- Free
- Contains many important features
- Regularly updated
- Personal experience
- Many usability considerations that assist the user

Disadvantages

- Entry level software
- Does not offer same level of quality as paid software
- Unreliable, relies on donations

9.1.2 Sublime Text 2

The other software I considered was Sublime Text 2.

Sublime text 2 is an incredibly easy to use and user-friendly piece of free software that was originally created in 2008. It is like Notepad++ free to download and use, and has an impressive feature list, with a significant amount of these designed to enhance user experience. For example, it contains many different options for syntax highlighting. In the screenshot provided below, I have the “html” syntax highlighting selected, meaning it will carefully select which elements it highlights to accommodate for HTML code more appropriately.

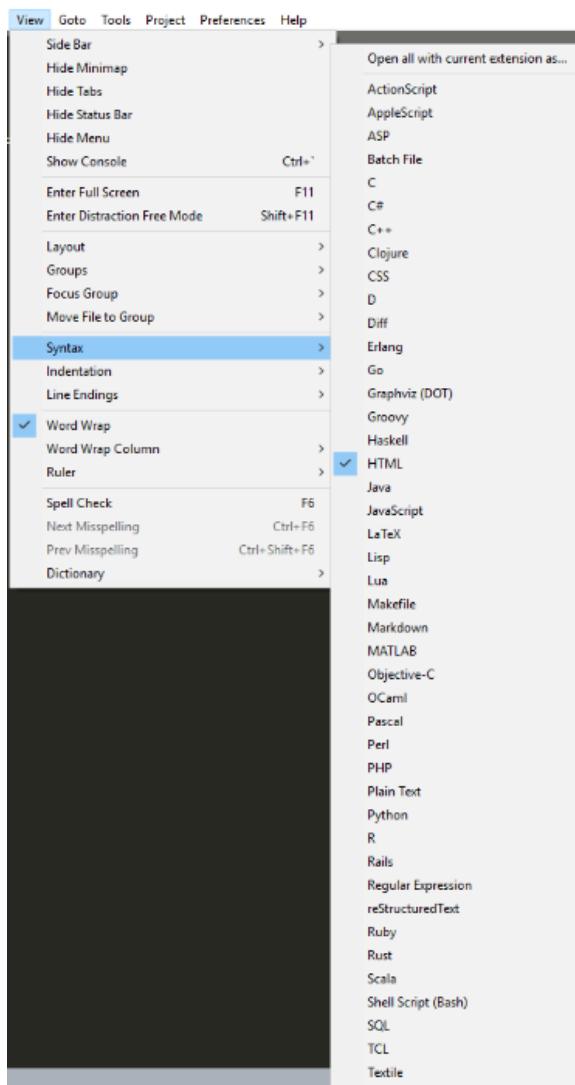


Figure 72 - Screenshot showing functionality of Sublime Text 2.



Hello world, Nathan here!

Figure 71 - Screenshot showing functionality of Sublime Text 2.

Also like Notepad++, Sublime Text supports the use of HTML with CSS, as well as

supporting jQuery and JavaScript integration and injection, meaning it satisfies all my needs for this project. Regarding the mentioned feature list, ST2 contains many usability optimisations which enhance the user experience. Examples of these being its “GoTo Anything” function, which allows users to open new files and find lines, words or symbols with the press of only a few keys. It works by pressing CTRL + P and then type the file name, symbol or colon with a number to jump to a line number. Another feature is multiple selections. This means that if a user makes a mistake, instead of editing every line where that mistake occurs, they can instead select every instance of it at once, and then add to, modify or remove it. The GoTo definition feature included provides on-the-fly definitions for methods and commands the user implements. For example, if the user writes some code and then later forgets what it does when revisiting, they can use this feature to check what that function does and remind themselves what the purpose of that line is for.

However, like N++, ST2 is a free software that is not reliably financially supported, and relies on user generosity and donations. While its development is active now, there is always the risk of development being ceased in future for any period of time, or indefinitely.

To conclude, ST2 is a more sophisticated, feature-rich free software that enhances the user experiences with a number of shortcuts and optimisations. While ST2 is clearly a step-up from N++, I have no prior experience with it and this could create issues during the development cycle.

Advantages

- Free
- Multitude of usability enhancements
- Supports many integrations with different softwares
- Split editing support
- GoTo feature has broad possibilities

Disadvantages

- Personal unfamiliarity
- Too many enhancements could make simple changes needlessly complex for new users
- Possibility for enhancements to degrade user experience

9.2 Chosen Software

This subsection will detail which software was chosen and why.

Ultimately, I decided to “play it safe” and use Notepad++ for this project. The reasoning for this is that I am already familiar with exactly what N++ can do. I know what software integrations it supports, I know what it looks like, I know what usability features it must make my experience streamlined, and I know that there are no surprises waiting for me when I create my project.

ST2 was a very tempting choice, and took some real weighing of options before I decided on N++. This was because while ST2 is free and undoubtedly feature-rich, I am not familiar with it, and its many functions could've caused problems for me later in development, which would've impacted particularly fiercely if those problems arose towards the end of my site's development. There are too many variables of ST2 that could trip me up, and could make seemingly simple fixes or adjustments too complex and take up too much time that would've been better spent using N++.

Finally, StanLearn is simple site that will need very little in the way of complex operations besides the implementation of the Java. Notepad++ fulfils all the requests of the StanLearn project and more, while ST2 undoubtedly does this as well, it goes above and beyond what is required for this project.

9.3 Website Creation

With the appropriate software chosen, the development of the website began. The following subsubsections will explain the process behind developing each page, as well as their individual complexities.

Before developing my project, I always had the idea of complete modularity in mind. This means that I wanted every page to be individually modifiable, and so there are a lot more HTML pages than may have been required. This was done on purpose, as if I wished to make minute, individual edits to only some pages later, I could easily do that without risk of affecting the content on any other pages. However, a lot of these pages are identical besides small changes such as changing the word “Swedish” to “German”, as well as a single background image. Because of this, I will not include every individual page as a “9.3.X”, but will instead talk about the layout of those pages and include all examples in image format.

9.3.1 Homepage

I found the first logical place to start at would be the homepage, as it is the first page that the user sees. The homepage in my opinion must link to every area of the site (apart from specifics such as quizzes etc) and so made a conscious effort to make the layout of the page readable and to make the content it links to easily findable and accessible.

The homepage is called “mainpage.html”, and its corresponding CSS is “main.css”. The use of CSS (cascading style sheets) greatly assisted in styling the site, and make certain functions available that otherwise would not have been. The HTML and CSS were linked together via this line.

```
<link rel="stylesheet" href="main.css" type="text/css"/>
```

```
<!DOCTYPE html>
<html>
  <head>
    <title>Home - StanLearn</title> <!--Decides what the page's name will be displayed as in browser.-->
    <link rel="stylesheet" href="main.css" type="text/css" media="screen" /> <!-- The CSS that the site will draw it's reference from
  </head>
  <body>
    <div id="headerContainer">
      <header>
        <a href="mainpage.html"> 
      </header>
    </div>
    <!-- Creation of navbar under header. -->
    <div id="navbar">
      <a href="about.html" class="navButton">About</a>
      <a href="contact.html" class="navButton">Contact</a>
      <a href="swedish.html" class="navButton">Swedish</a>
      <a href="french.html" class="navButton">French</a>
      <a href="spanish.html" class="navButton">Spanish</a>
      <a href="german.html" class="navButton">German</a>
      <a href="mainpage.html" class="navButton">Back Home</a>
    </div>
    <div id="main_body">
      <h2> StanLearn Language Learning </h2>
      <p> Welcome to StanLearn Language Learning! This site aims to be the number one resource for users of all ages and all abilities.
      <p> Our language choices currently stand at - </p>
      <p> Swedish </p>
      <p> French </p>
      <p> Spanish </p>
      <p> German </p>
      <p> We offer four quizzes in each of these subjects with the topics animals, numbers, colours and food! </p>
      <p> Don't forget to check the word sheets before you start learning to remind yourself of words you may have forgotten! </p>
    </div>
    <div id="copyrightDiv">
      <p> © Copyright Nathan Stanley 2017-2018 </p>
    </div>
  </body>
```

Figure 73 - HTML snippet of final code used in site.

```

* {
    padding: 0;
    margin: 5px;
}

#headerContainer {
    border: 1px solid black;
    padding: 0px;
    background-color: #B0D7D7;
    max-width:100%;
    max-height:100%;
    overflow: hidden;
}

body {
    /* font-family: Arial, Verdana, sans-serif; */
    font-family: Bahnschrift, sans-serif;
    font-style: normal;
    font-size: 100%;
    background-color: #9D9694;
}

#main_body {
    font-family: Bahnschrift Light, sans-serif;
    padding: 20px 10px 20px 10px;
    color: solid black;
    border: 1px solid black;
    text-align: left;
    background-color: #B0D7D7;
    font-size: 18px;
}

#navbar {
    padding: 20px 10px 20px 10px;
    color: solid black;
    border: 1px solid black;
    text-align: center;
    background-color: #B0D7D7; /* this section relates to the styling of the nav bar */
}

#copyrightDiv {
    font-size: 16px;
    position: fixed;
    border: 1px solid black;
    bottom: 0;
    width: 98%; /* this style section serves to place the copyright div at the bottom */
    background-color: #B0D7D7;
    padding: 10px 0px 10px 10px;
}

.navButton {
    border: 1px solid black;
    padding: 8px 8px 8px 8px;
    color: black;
    font: bold 100px;
    text-decoration: none;
    background-color: #97BDC9;
}

```

Figure 74 - CSS snippet of final code used in site.

The images above showcase the HTML and then the CSS in that order. One of the problems I was having was the image in the header. It is a href that links to the homepage, but for some reason on some pages the header image would overflow over the div it was supposed to be contained in (the header div). To fix this, and maintain an air of consistency, I gave all CSS files the overflow: hidden attribute in the header. This cuts off the part of the header image that is overflowing, if it is, and makes the image position appear the same for all pages.

The spacing between the elements was achieved by setting padding values, and the styling of the buttons on the navbar was done because leaving the navigation buttons laid out as text was a practice I felt unprofessional. By creating each of the navigation buttons with the **class="navButton"** at the end of the href, I was able to then call the buttons in the CSS with **.navButton** and style them from there. All this code is visible in the screenshots above. Below is how the site appears on different browsers. I included the navbar on all pages except for the quiz pages. This was done to aid the user in usability, and make it clear to them they could return to the homepage at any time if they felt confused. The only pages the navbar is not present on is the quiz pages, so that the user did not accidentally navigate elsewhere.

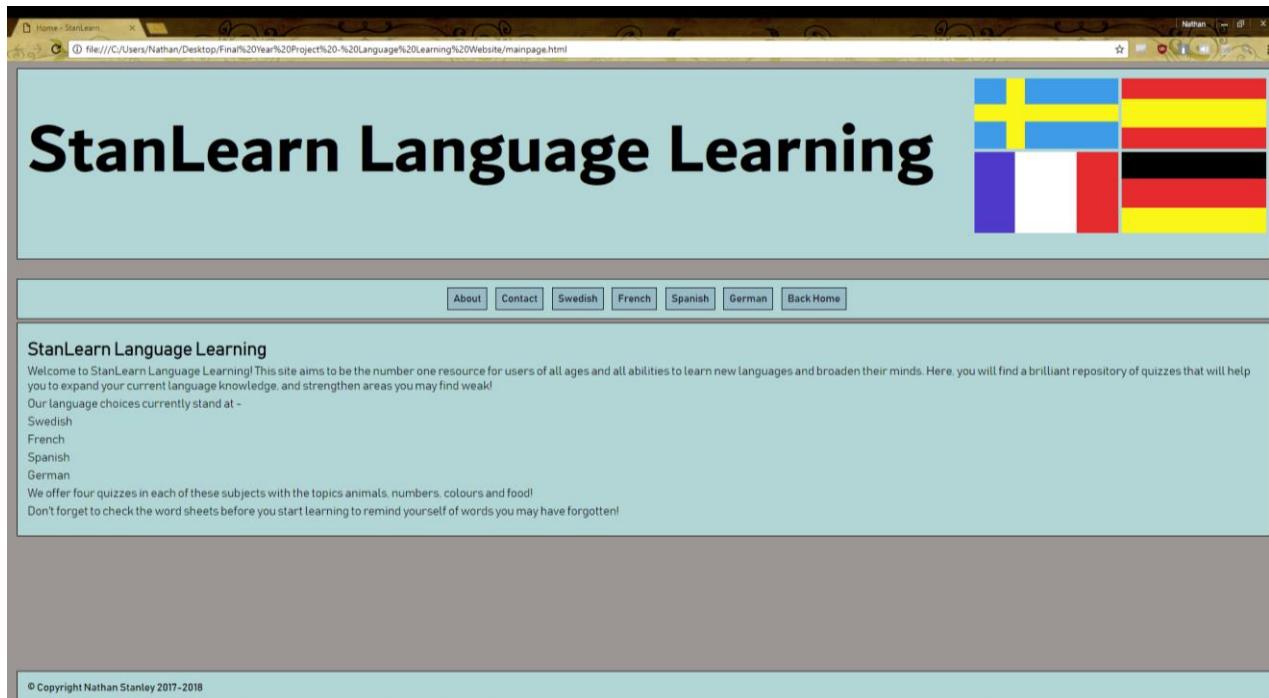


Figure 75 - Homepage in Google Chrome

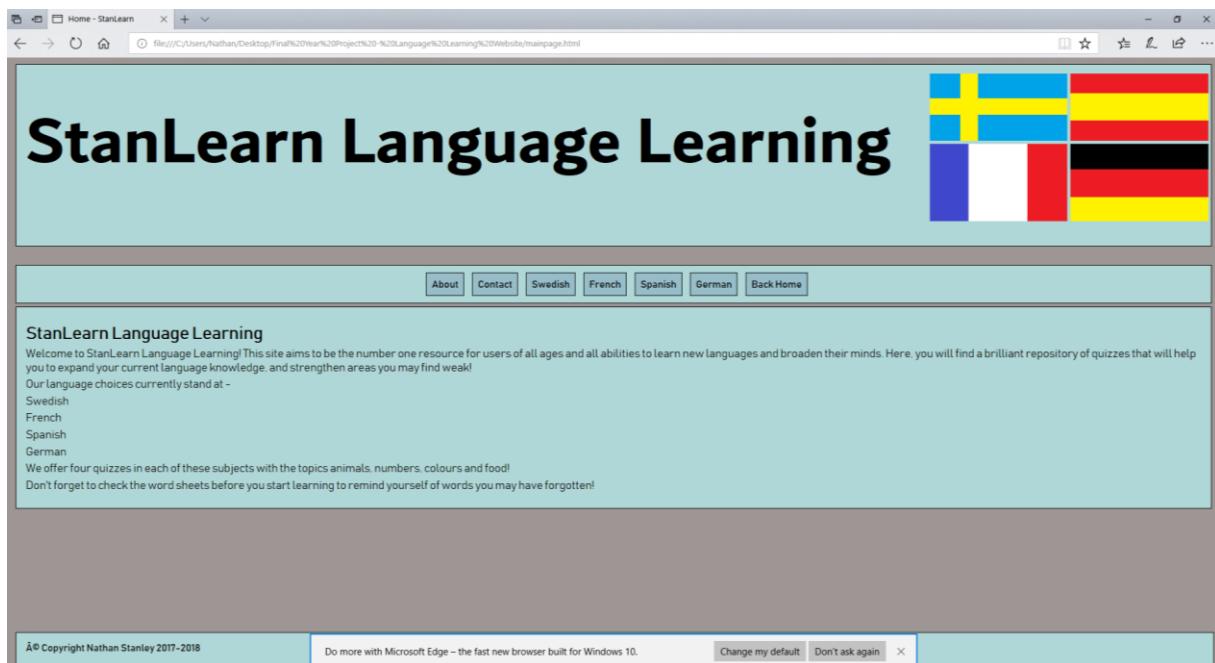


Figure 76 - Homepage in Microsoft Edge

9.3.2 About Page

After creating the homepage, I felt it appropriate to create an “about” page. This page’s purpose was simply to inform the user what StanLearn was about. As its design did not need to differ much from the homepage apart from a few lines of text, it uses the same CSS that the mainpage does – **main.css**.

```

<!DOCTYPE html>
<html>
  <head>
    <title>About Us</title> <!--Decides what the page's name will be displayed as in browser.-->
    <link rel="stylesheet" href="main.css" type="text/css" media="screen" /> <!-- The CSS that the site will
  </head>
  <body>
    <div id="headerContainer">
      <header>
        <a href="mainpage.html"> 
      </header>
    </div>
    <!-- Creation of navbar under header. -->
    <div id="navbar">
      <a href="about.html" class="navButton">About</a>
      <a href="contact.html" class="navButton">Contact</a>
      <a href="swedish.html" class="navButton">Swedish</a>
      <a href="french.html" class="navButton">French</a>
      <a href="spanish.html" class="navButton">Spanish</a>
      <a href="german.html" class="navButton">German</a>
      <a href="mainpage.html" class="navButton">Back Home</a>
    </div>
    <div id="main_body">
      <h2> About StanLearn </h2>
      <p><b> What is StanLearn? </b></p><br>
      <p> StanLearn is the next big thing in language learning! In it's current state, StanLearn aims to assist users as best as possible in learning a new European language! </p>
      <p> Currently, there are four languages you can choose to start learning, each having four different topics and their own wordlist! These languages are: </p>
      <ul style="list-style-type: none; padding-left: 0; margin: 0; font-size: small; color: #666; font-family: sans-serif; font-weight: bold;">
        <li>French</li>
        <li>German</li>
        <li>Spanish</li>
        <li>Swedish</li>
      </ul>
      <p> The different topics you can choose to start learning are animals, numbers, colours and food! these languages are popular first choices for people who wish to learn a second language in Europe, and so I felt it appropriate to allow you to start right here! StanLearn was created from the years 2017-2018 by Nathan Stanley. </p>
    </div>
    <div id="copyrightDiv">
      <p> © Copyright Nathan Stanley 2017-2018 </p>
    </div>
  </body>

```

Figure 77 - HTML snippet of final code used in site.

```

* {
    padding: 0;
    margin: 5px;
}

#headerContainer {
    border: 1px solid black;
    padding: 0px;
    background-color: #B0D7D7;
    max-width:100%;
    max-height:100%;
    overflow: hidden;
}

body {
    /* font-family: Arial, Verdana, sans-serif; */
    font-family: Bahnschrift, sans-serif;
    font-style: normal;
    font-size: 100%;
    background-color: #9D9694;
}

#main_body {
    font-family: Bahnschrift Light, sans-serif;
    padding: 20px 10px 20px 10px;
    color: solid black;
    border: 1px solid black;
    text-align: left;
    background-color: #B0D7D7;
    font-size: 18px;
    line-height: 30px;
}

#navbar {
    padding: 20px 10px 20px 10px;
    color: solid black;
    border: 1px solid black;
    text-align: center;
    background-color: #B0D7D7; /* this section relates to t
}

#copyrightDiv {
    font-size: 16px;
    position: fixed;
    border: 1px solid black;
    bottom: 0;
    width: 98%; /* this style section serves to place the c
    background-color: #B0D7D7;
    padding: 10px 0px 10px 10px;
}

.navButton {
    border: 1px solid black;
    padding: 8px 8px 8px 8px;
    color: black;
    font: bold 100px;
    text-decoration: none;
    background-color: #97BDC9;
}

```

Figure 78 - CSS snippet of final code used in site.

A problem I encountered this page when writing the short brief was that the text appeared too close together, and ended up being hard to read and looked aesthetically displeasing. To fix this, I inserted a line of code into the CSS – **line-height: 30px;** - into the main_body, where the brief is located, and it made the site appear much more presentable and readable.

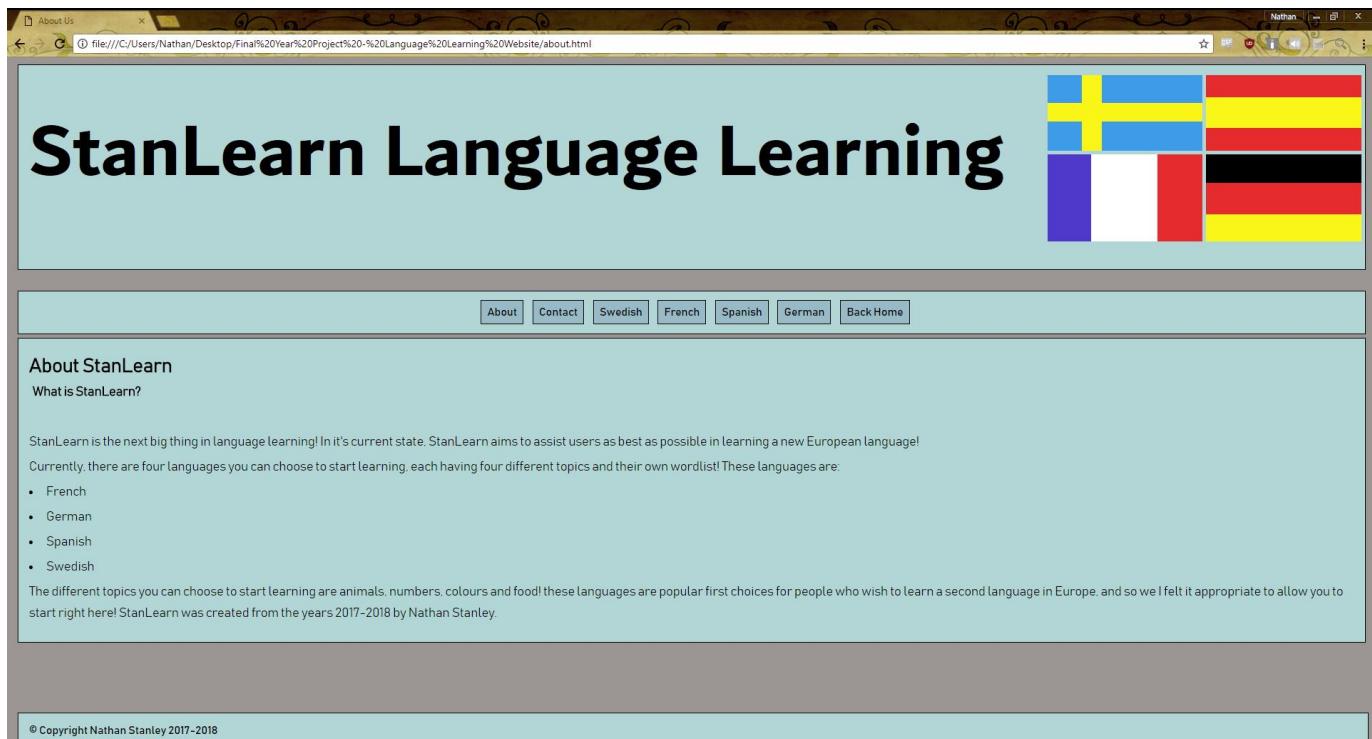


Figure 79 - About page in Google Chrome.

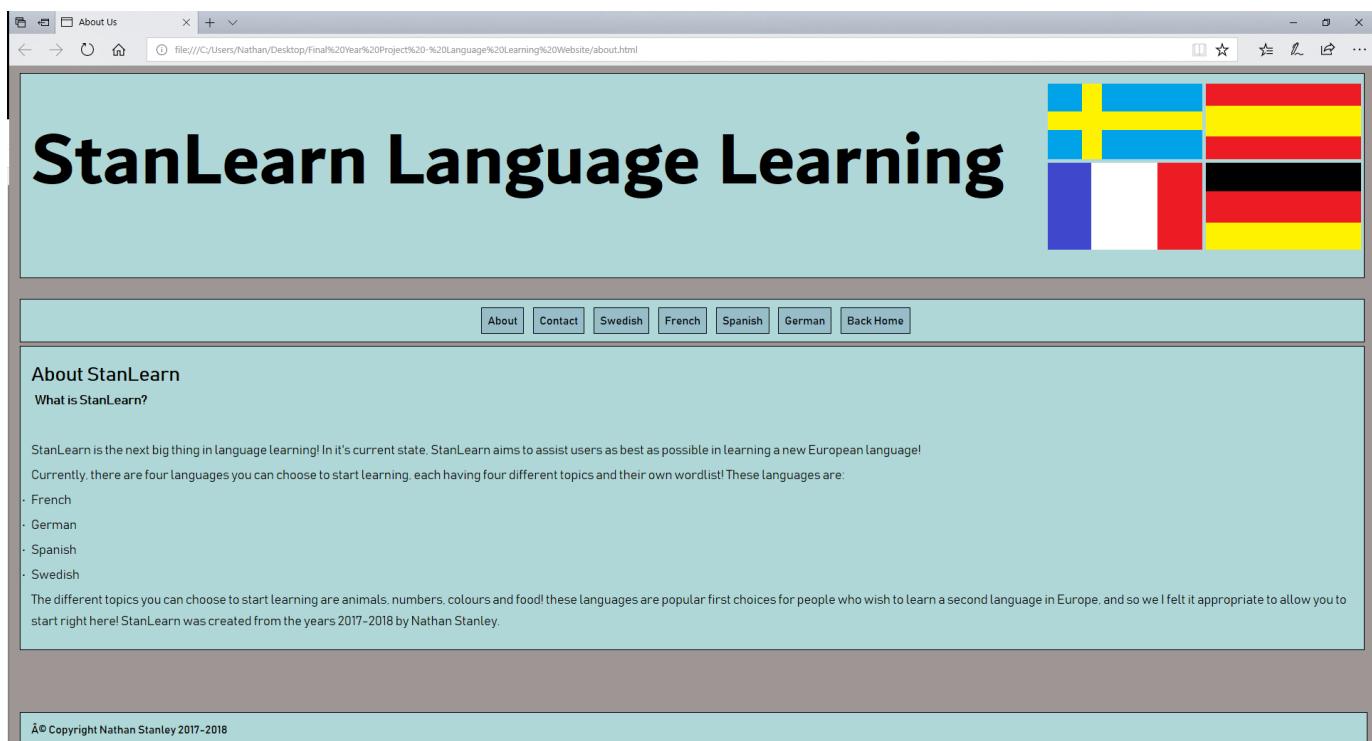


Figure 80 - About page in Microsoft Edge.

9.3.3 Contact Page

The next page I created was the contact page. This page was created to create the atmosphere of professionalism and realism I wanted the site to have. It is a short page that features an email address for reporting problems, and email address for suggesting new languages, and a fake postcode and office location. Because the contact page contains an image, I used a different CSS for it, named **main2.css**.

```

<!DOCTYPE html>
<html>
  <head>
    <title>Contact Us</title> <!--Decides what the page's name will be displayed as in browser.-->
    <link rel="stylesheet" href="main2.css" type="text/css" media="screen" /> <!-- The CSS that the site will draw it's reference from. -->
  </head>
  <body>
    <div id="headerContainer">
      <header>
        <a href="mainpage.html"> 
      </header>
    </div>
    <!-- Creation of navbar under header. -->
    <div id="navbar">
      <a href="about.html" class="navButton">About</a>
      <a href="contact.html" class="navButton">Contact</a>
      <a href="swedish.html" class="navButton">Swedish</a>
      <a href="french.html" class="navButton">French</a>
      <a href="spanish.html" class="navButton">Spanish</a>
      <a href="german.html" class="navButton">German</a>
      <a href="mainpage.html" class="navButton">Back Home</a>
    </div>
    <div id="main_body">
      <h2> Contact StanLearn </h2>
      <p> This is the page where you will find our contact details! This page allows you to contact us regarding any trouble you may have using the site, or allows you to suggest new languages for us to implement!</p><br>
      <p><strong> Problem? Email us here: </strong></p><br>
      <p><strong> customerissues@stanLearn.co.uk </strong></p><br>
      <p><strong> Suggestions? Suggest them here: </strong></p><br>
      <p><strong> customersuggestions@stanLearn.co.uk </strong></p><br>
      <p><strong> Or, you can reach us by post! Our building is in the Google Maps image below, and our postcode is as follows: </strong></p><br>
      <p> L3 4AY </p>
      
    </div>
    <div id="copyrightDiv">
      <p> © Copyright Nathan Stanley 2017-2018 </p>
    </div>
  </body>

```

Figure 81 - HTML snippet of final code used in site.

```

* {
    padding: 0;
    margin: 5px;
}

#headerContainer {
    border: 1px solid black;
    padding: 0px;
    background-color: #B0D7D7;
    max-width:100%;
    max-height:100%;
    overflow: hidden;
}

body {
    /* font-family: Arial, Verdana, sans-serif; */
    font-family: Bahnschrift, sans-serif;
    font-style: normal;
    font-size: 100%;
    background-color: #9D9694;
}

#main_body {
    font-family: Bahnschrift Light, sans-serif;
    padding: 20px 10px 20px 10px;
    color: solid black;
    border: 1px solid black;
    text-align: left;
    background-color: #B0D7D7;
    font-size: 16px;
    overflow: auto;
}

img {
    border-radius: 8px;
    float: left;
}

#navbar {
    padding: 20px 10px 20px 10px;
    color: solid black;
    border: 1px solid black;
    text-align: center;
    background-color: #B0D7D7; /* this section relates to the s */
}

#copyrightDiv {
    font-size: 16px;
    position: relative;
    border: 1px solid black;
    bottom: 0;
    width: 98%; /* this style section serves to place the copyr
    background-color: #B0D7D7;
    padding: 10px 0px 10px 10px;
}

.navButton {
    border: 1px solid black;
    padding: 8px 8px 8px 8px;
    color: black;
    font: bold 100px;
    text-decoration: none;
    background-color: #97BDC9;
}

```

Figure 82 - CSS snippet of final code used in site.

During this page's creation, I began having trouble placing the image inside the div. I first wanted it to float right, inside the div, but it left the div and messed up the page layout. To combat this, I made the image float left instead, and set the overflow on the div it was inside to auto, so it would be able to float but not leave the div. Once I was able to make it float to the right, I decided it looked better on the left as it was. I also gave the image a border radius of 8px, to make the edges rounder and make it look more aesthetically pleasing. The images below only contain the content added in the contact page, but the navbar, header and copyright bar are all still present.

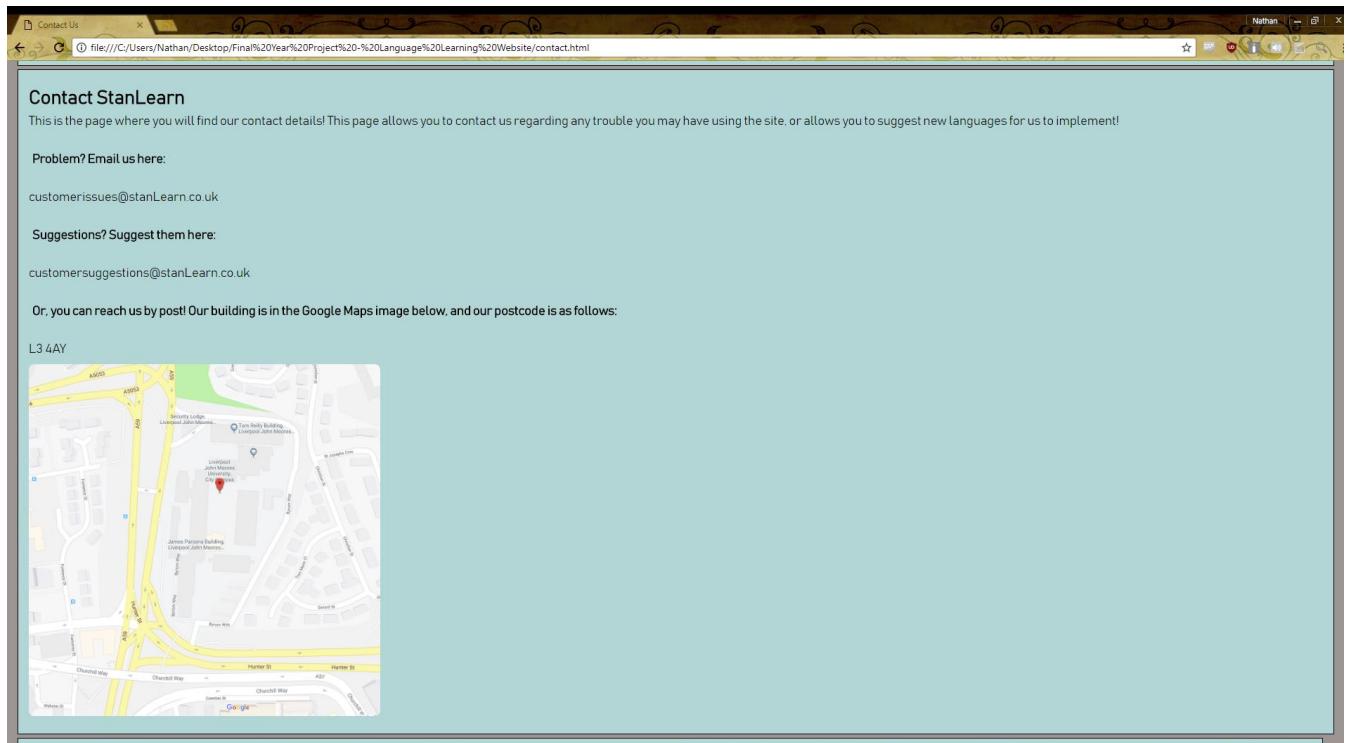


Figure 83 - Contact page in Google Chrome.

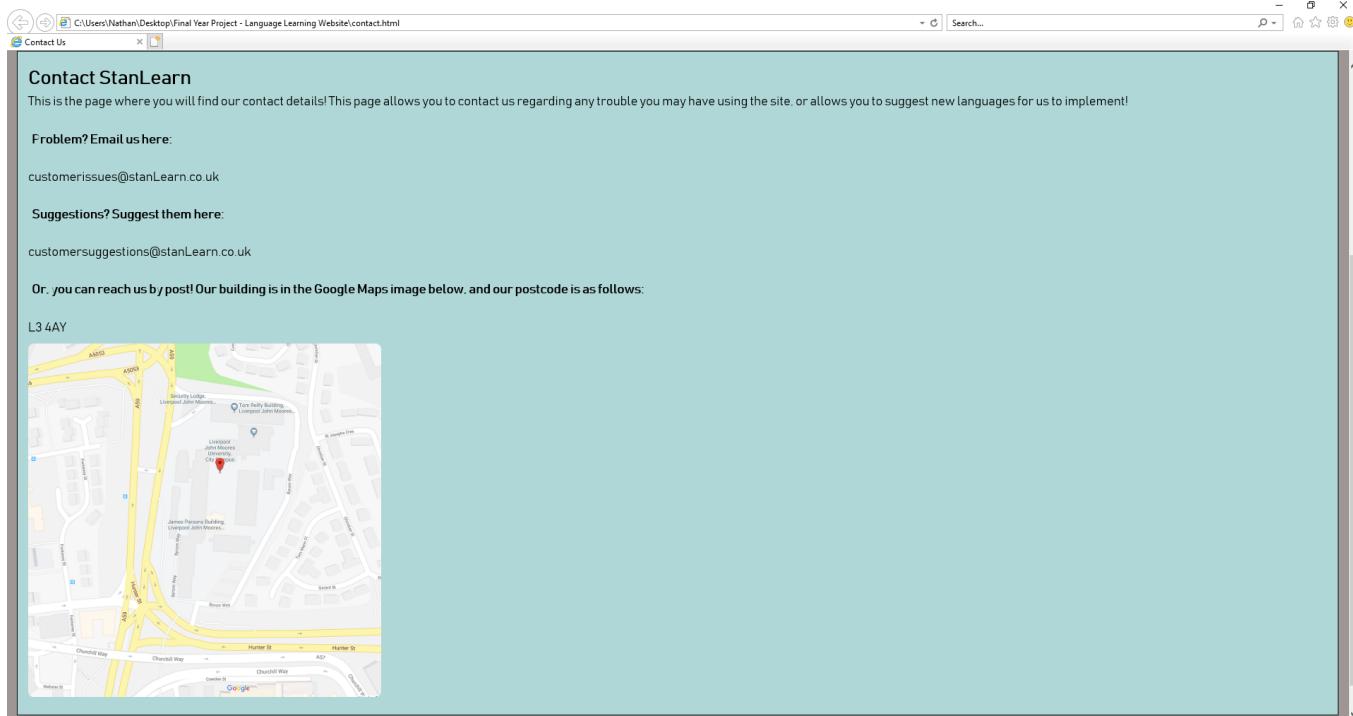


Figure 84 - Contact page in Microsoft Edge

9.3.4 Language Introduction / Entry Pages

Next, was the language introduction pages. What these pages do is tell the user what language area they are about to enter, before they begin taking and completing quizzes. This was done to provide a buffer zone between the user being on the normal areas of the site (homepage, contact etc) and the quiz section, to help them establish where they are. These language introduction pages present the gateway to the quizzes and wordlist, with the

different topics and the wordlist being shown in the main body along with a title and short explanation, as well as a greeting in said language. The main body also contains a background image with a flag of the nation who created that language. For example, the Swedish language area will display a small greeting in Swedish, a title that tells the user they are in the Swedish language area as well as the Swedish flag as the background. All flag images were made by me in MS Paint. All language entry pages are shown in the images below.



Figure 85 - Swedish language area banner graphic.

The HTML and CSS of these pages was fairly easy to create. I was able to reuse the HTML from the mainpage, and take out the text in the main body and replace it with the background image, title, small paragraph and the 5 buttons shown in the example above. Each language entry page has its own CSS as well, named appropriately for their language. For example, the Swedish entry zone (**swedish.html**) and the French entry zone (**french.html**) use the style sheets **mainSwedish.css** and **mainFrench.css** respectively. The German and Spanish areas maintain the same name styles for their HTML and CSS too (**german.html**, **spanish.html**, **mainGerman.css**, **mainSpanish.css**). The screenshots below show the HTML for the Swedish, French, German and Spanish zones in that order.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Swedish Language</title> <!--Decides what the page's name will be displayed as in browser.-->
    <link rel="stylesheet" href="mainSwedish.css" type="text/css" media="screen" /> <!-- The CSS that the site will draw it's reference from. -->
  </head>
  <body>
    <div id="headerContainer">
      <header>
        <a href="mainpage.html"> 
      </header>
    </div>
    <!-- Creation of navbar under header. -->
    <div id="navbar">
      <a href="about.html" class="navButton">About</a>
      <a href="contact.html" class="navButton">Contact</a>
      <a href="swedish.html" class="navButton">Swedish</a>
      <a href="french.html" class="navButton">French</a>
      <a href="spanish.html" class="navButton">Spanish</a>
      <a href="german.html" class="navButton">German</a>
      <a href="mainpage.html" class="navButton">Back Home</a>
    </div>
    <div id="main_body">
      <h2> Swedish Language Area </h2>
      <br><br>
      <p> Hallå! This is the area specialising in the Swedish language. Please select from the categories below which area you would like to be tested on. </p>
      <br><br>
      <!-- <br><br> -->
      <a href="swedishQ1.html" class="quizSelect">Animals</a>
      <a href="swedishQ2.html" class="quizSelect">Numbers</a>
      <a href="swedishQ3.html" class="quizSelect">Colours</a>
      <a href="swedishQ4.html" class="quizSelect">Food</a>
      <a href="SWBwordlist.html" class="quizSelect">Wordlist</a>
    </div>
    <div id="copyrightDiv">
      <p> © Copyright Nathan Stanley 2017-2018 </p>
    </div>
  </body>
```

Figure 86 - HTML snippet of final code used in site.

```

<!DOCTYPE html>
<html>
  <head>
    <title>French Language</title> <!--Decides what the page's name will be displayed as in browser.-->
    <link rel="stylesheet" href="mainFrench.css" type="text/css" media="screen" /> <!-- The CSS that the site will draw it's reference from. -->
  </head>
  <body>
    <div id="headerContainer">
      <header>
        |   <a href="mainpage.html"> 
      </header>
    </div>
    <!-- Creation of navbar under header. -->
    <div id="navbar">
      <a href="about.html" class="navButton">About</a>
      <a href="contact.html" class="navButton">Contact</a>
      <a href="swedish.html" class="navButton">Swedish</a>
      <a href="french.html" class="navButton">French</a>
      <a href="spanish.html" class="navButton">Spanish</a>
      <a href="german.html" class="navButton">German</a>
      <a href="mainpage.html" class="navButton">Back Home</a>
    </div>
    <div id="main_body">
      <h2> French Language Area </h2>
      <br><br>
      <p> Bonjour! This is the area specialising in the French language. Please select from the categories below which area you would like to be tested on. </p>
      <br><br>
      <!-- <br><br> -->
      <a href="frenchQ1.html" class="quizSelect"> Animals</a>
      <a href="frenchQ2.html" class="quizSelect">Numbers</a>
      <a href="frenchQ3.html" class="quizSelect">Colours</a>
      <a href="frenchQ4.html" class="quizSelect">Food</a>
      <a href="FRAwordList.html" class="quizSelect">Wordlist</a>
    </div>
    <div id="copyrightDiv">
      <p> © Copyright Nathan Stanley 2017-2018 </p>
    </div>
  </body>

```

Figure 87 - HTML snippet of final code used in site.

```

<!DOCTYPE html>
<html>
  <head>
    <title>German Language</title> <!--Decides what the page's name will be displayed as in browser.-->
    <link rel="stylesheet" href="mainGerman.css" type="text/css" media="screen" /> <!-- The CSS that the site will draw it's reference from. -->
  </head>
  <body>
    <div id="headerContainer">
      <header>
        |   <a href="mainpage.html"> 
      </header>
    </div>
    <!-- Creation of navbar under header. -->
    <div id="navbar">
      <a href="about.html" class="navButton">About</a>
      <a href="contact.html" class="navButton">Contact</a>
      <a href="swedish.html" class="navButton">Swedish</a>
      <a href="french.html" class="navButton">French</a>
      <a href="spanish.html" class="navButton">Spanish</a>
      <a href="german.html" class="navButton">German</a>
      <a href="mainpage.html" class="navButton">Back Home</a>
    </div>
    <div id="main_body">
      <h2> German Language Area </h2>
      <br><br>
      <p> Hallo! This is the area specialising in the German language. Please select from the categories below which area you would like to be tested on. </p>
      <br><br>
      <!-- <br><br> -->
      <a href="germanQ1.html" class="quizSelect">Animals</a>
      <a href="germanQ2.html" class="quizSelect">Numbers</a>
      <a href="germanQ3.html" class="quizSelect">Colours</a>
      <a href="germanQ4.html" class="quizSelect">Food</a>
      <a href="GERwordList.html" class="quizSelect">Wordlist</a>
    </div>
    <div id="copyrightDiv">
      <p> © Copyright Nathan Stanley 2017-2018 </p>
    </div>
  </body>

```

Figure 88 - HTML snippet of final code used in site.

```

<!DOCTYPE html>
<html>
<head>
    <title>Spanish Language</title> <!--Decides what the page's name will be displayed as in browser.-->
    <link rel="stylesheet" href="mainSpanish.css" type="text/css" media="screen" /> <!-- The CSS that the site will draw it's reference from. -->
</head>
<body>
    <div id="headerContainer">
        <header>
            |   <a href="mainpage.html"> 
        </header>
    </div>
    <!-- Creation of navbar under header. -->
    <div id="navbar">
        <a href="about.html" class="navButton">About</a>
        <a href="contact.html" class="navButton">Contact</a>
        <a href="swedish.html" class="navButton">Swedish</a>
        <a href="french.html" class="navButton">French</a>
        <a href="spanish.html" class="navButton">Spanish</a>
        <a href="german.html" class="navButton">German</a>
        <a href="mainpage.html" class="navButton">Back Home</a>
    </div>
    <div id="main_body">
        <h2> Spanish Language Area </h2>
        <br><br>
        <p> Hola! This is the area specialising in the Spanish language. Please select from the categories below which area you would like to be tested on. </p>
        <br><br>
        <!-- <br><br> -->
        <a href="spanishQ1.html" class="quizSelect">Animals</a>
        <a href="spanishQ2.html" class="quizSelect">Numbers</a>
        <a href="spanishQ3.html" class="quizSelect">Colours</a>
        <a href="spanishQ4.html" class="quizSelect">Food</a>
        <a href="SPAwordlist.html" class="quizSelect">Wordlist</a>
    </div>
    <div id="copyrightDiv">
        <p> © Copyright Nathan Stanley 2017-2018 </p>
    </div>
</body>

```

Figure 89 - HTML snippet of final code used in site.

And these screenshots show the CSS, in that same order. Because of the policy of modularity, I self-implemented at the beginning of the project's creation, I was easily able to make small adjustments to the colour of the text on each of the flags, to both enhance readability and to add a stylistic flair which would appeal visually to all users.

```
* {
    padding: 0;
    margin: 5px;
}

body {
    /* font-family: Arial, Verdana, sans-serif; */
    font-family: Bahnschrift, sans-serif;
    font-style: normal;
    font-size: 100%;
    background-color: #9D9694;
}

#headerContainer {
    border: 1px solid black;
    padding: 0px;
    background-color: #B0D7D7;
    max-width: 100%;
    max-height: 100%;
    overflow: hidden;
}

#main_body {
    font-size: 19px;
    padding: 20px 10px 20px 10px;
    color: solid black;
    border: 1px solid black;
    text-align: center;
    background-color: #B0D7D7;
    background-image: url("swedishlanguagearea.png");
}

#navbar {
    padding: 20px 10px 20px 10px;
    color: solid black;
    border: 1px solid black;
    text-align: center;
    background-color: #B0D7D7; /* this section relates to the
}

#copyrightDiv {
    position: fixed;
    border: 1px solid black;
    bottom: 0;
    width: 98%; /* this style section serves to place the co
    background-color: #B0D7D7;
    padding: 10px 0px 10px 10px;
}

.quizSelect {
    border: 1px solid black;
    padding: 6px 6px 6px 6px;
    color: black;
    font: bold 100px;
    text-decoration: none;
}

.navButton {
    border: 1px solid black;
    padding: 8px 8px 8px 8px;
    color: black;
    font: bold 100px;
    text-decoration: none;
    background-color: #97BDC9;
```

Figure 90 - CSS snippet of final code used in site.

The screenshot shows a CSS editor interface with several tabs at the top: 'sh.html', 'mainSwedish.css', 'mainFrench.css' (which is the active tab), 'mainGerman.css', and 'mainSpanish.css'. The main area displays the following CSS code:

```
* { padding: 0; margin: 5px; }
body {
    /* font-family: Arial, Verdana, sans-serif; */
    font-family: Bahnschrift, sans-serif;
    font-style: normal;
    font-size: 100%;
    background-color: #9D9694;
}
#headerContainer {
    border: 1px solid black;
    padding: 0px;
    background-color: #B0D7D7;
    max-width: 100%;
    max-height: 100%;
    overflow: hidden;
}
#main_body {
    font-size: 19px;
    padding: 20px 10px 20px 10px;
    color: solid black;
    border: 1px solid black;
    text-align: center;
    background-color: #B0D7D7;
    background-image: url("frenchlanguagearea.png");
}
#navbar {
    padding: 20px 10px 20px 10px;
    color: solid black;
    border: 1px solid black;
    text-align: center;
    background-color: #B0D7D7; /* this section relates to the sidebar */
}
#copyrightDiv {
    position: fixed;
    border: 1px solid black;
    bottom: 0;
    width: 98%; /* this style section serves to place the copyright notice at the bottom */
    background-color: #B0D7D7;
    padding: 10px 0px 10px 10px;
}
.quizSelect {
    border: 1px solid red;
    padding: 6px 6px 6px 6px;
    color: red;
    font: bold 100px;
    text-decoration: none;
}
.navButton {
    border: 1px solid black;
    padding: 8px 8px 8px 8px;
    color: black;
    font: bold 100px;
    text-decoration: none;
    background-color: #97BDC9;
}
```

Figure 91 - CSS snippet of final code used in site.

```
* {
    padding: 0;
    margin: 5px;
}

body {
    /* font-family: Arial, Verdana, sans-serif; */
    font-family: Bahnschrift, sans-serif;
    font-style: normal;
    font-size: 100%;
    background-color: #9D9694;
}

#headerContainer {
    border: 1px solid black;
    padding: 0px;
    background-color: #B0D7D7;
    max-width: 100%;
    max-height: 100%;
    overflow: hidden;
}

#main_body {
    font-size: 19px;
    padding: 20px 10px 20px 10px;
    color: white;
    border: 1px solid black;
    text-align: center;
    background-color: #B0D7D7;
    background-image: url("germanlanguagearea.png");
}

#navbar {
    padding: 20px 10px 20px 10px;
    color: solid black;
    border: 1px solid black;
    text-align: center;
    background-color: #B0D7D7; /* this section relates to the site */
}

#copyrightDiv {
    position: fixed;
    border: 1px solid black;
    bottom: 0;
    width: 98%; /* this style section serves to place the copyright notice at the bottom of the page */
    background-color: #B0D7D7;
    padding: 10px 0px 10px 10px;
}

.quizSelect {
    border: 1px solid black;
    padding: 6px 6px 6px 6px;
    color: black;
    font: bold 100px;
    text-decoration: none;
}

.navButton {
    border: 1px solid black;
    padding: 8px 8px 8px 8px;
    color: black;
    font: bold 100px;
    text-decoration: none;
    background-color: #97BDC9;
```

Figure 92 - CSS snippet of final code used in site.

```
* {
    padding: 0;
    margin: 5px;
}

body {
    /* font-family: Arial, Verdana, sans-serif; */
    font-family: Bahnschrift, sans-serif;
    font-style: normal;
    font-size: 100%;
    background-color: #9D9694;
}

#headerContainer {
    border: 1px solid black;
    padding: 0px;
    background-color: #B0D7D7;
    max-width: 100%;
    max-height: 100%;
    overflow: hidden;
}

#main_body {
    font-size: 19px;
    padding: 20px 10px 20px 10px;
    color: solid black;
    border: 1px solid black;
    text-align: center;
    background-color: #B0D7D7;
    background-image: url("spanishlanguagearea.png");
}

#navbar {
    padding: 20px 10px 20px 10px;
    color: solid black;
    border: 1px solid black;
    text-align: center;
    background-color: #B0D7D7; /* this section relates to the st
}

#copyrightDiv {
    position: fixed;
    border: 1px solid black;
    bottom: 0;
    width: 98%; /* this style section serves to place the copyri
    background-color: #B0D7D7;
    padding: 10px 0px 10px 10px;
}

.quizSelect {
    border: 1px solid yellow;
    padding: 6px 6px 6px 6px;
    color: yellow;
    font: bold 100px;
    text-decoration: none;
}

.navButton {
    border: 1px solid black;
    padding: 8px 8px 8px 8px;
    color: black;
    font: bold 100px;
    text-decoration: none;
    background-color: #97BDC9;
}
```

Figure 93 - CSS snippet of final code used in site.

And below is how the language introduction pages look. Again, they are in the same order.

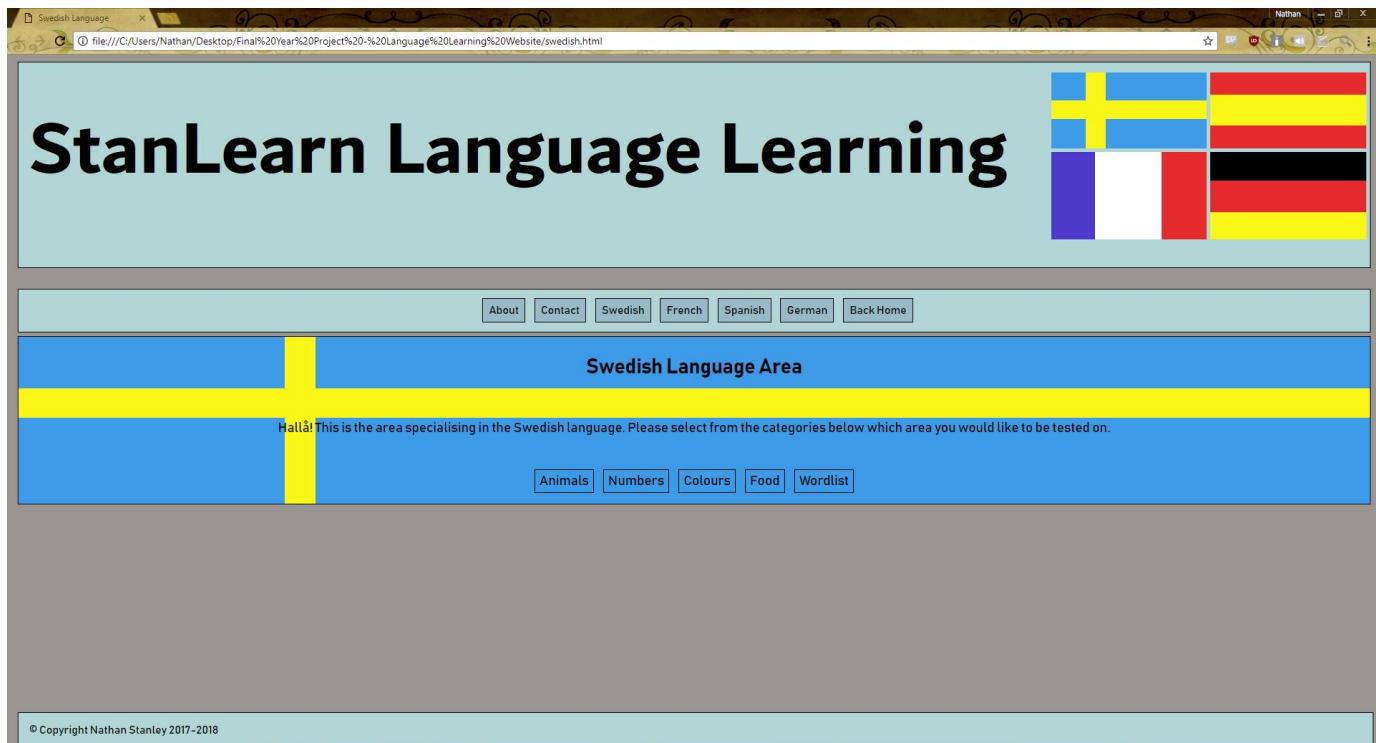


Figure 94 - Swedish language area in Google Chrome.

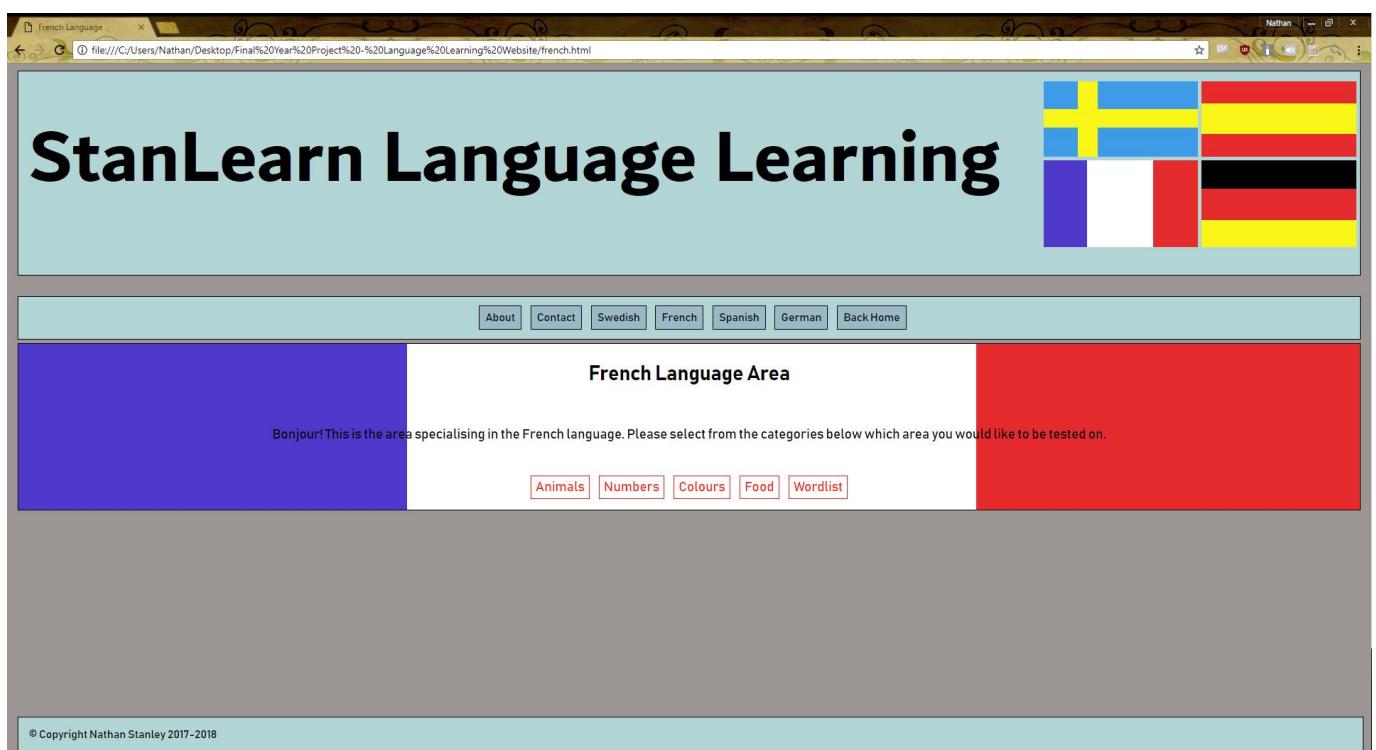


Figure 95 - French language area in Google Chrome.

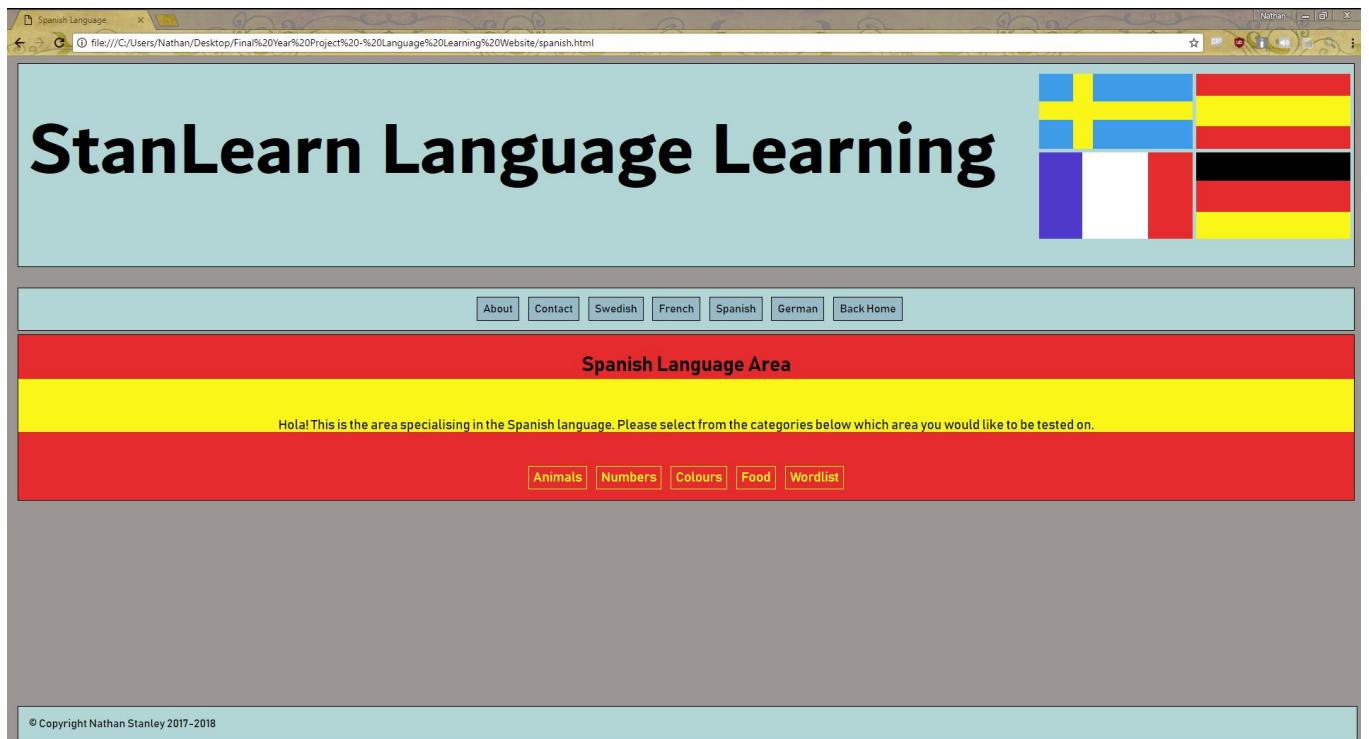


Figure 96 - Spanish language area in Google Chrome.



Figure 97 - German language area in Google Chrome.

9.3.5 Language Quiz Pages

After creating the quiz entry pages, I had to create the quizzes themselves. So that I could purpose-build these pages from the ground up, I started from scratch, but kept the div names when it came time to create them. Every question page follows the exact same layout, except the questions and answers have been changed, along with the topic name, language name, and page title at the top. For the sake of convenience and not to bloat out

this entire section, only one example question page will be shown. This is because every language has 4 question HTMLs (e.g. French has frenchQ1.html, frenchQ2.html, frenchQ3.html and frenchQ4.html), so I would need to show individually the code for 16 different pages. These pages in their entirety will be shown in the Appendices section. In the examples below, the first French quiz will serve as the example.

The structure of the header and main body stayed the same, except that the href in the header image that linked back to the homepage was removed, and of course the content in the main body changed according to the content of the quiz. For example, in the main body of each of the quiz pages is a `<h>` tag, which contains the language of the quiz and the topic, and a paragraph tag below it which explains how the quiz works.

```
<div id="main_body">
    <h2> French Quiz 1 - Animals </h2>
    <br></br>
    <p> Welcome to the French animals quiz! You will be asked a question, and you must select the correct answer. </p>
    <br></br>
```

Figure 98 - Code snippet of HTML used in final site.

Then, I had to create a new div to store the questions. I decided the best way to create the quiz would be to use radio buttons, as it is the best option when creating quizzes with multiple choices for answers. I wanted to create 4 different questions, and have 4 answers for each question. To do this, I created a div with the id “questions”, and then used the paragraph tag with the class “questions”, and used an unordered list tag with the tag “answers” for the answers. This was done so it was easier for me to style the individual elements later within the CSS, by calling upon the `.answers` and `.questions` classes, although in the end I didn’t need to do anything with them, but they are of more use kept as classes so that in the event I *do* need to manipulate them, I can.

The answers inside the unordered list were kept with labels. I considered using a class for the labels, but seeing as I found no use in classing the questions `<p>` tag and the answers `` tag, I left it. If I did need to class the labels for individual manipulation in the CSS however, this would be easily accomplished by copy pasting a new class with a new name besides the label tag.

```
<div id="questions">
    <p class="question"><b> Question 1. What is the French word for "cat"? . </b></p>
    <ul class="answers">
```

Figure 99 - Code snippet of HTML used in final site.

Next, I had to create names for each of the radio button answers so they could be read and manipulated by the JavaScript later. As I worked on and completed the Swedish questions first (it is a language I know personally, and so made the most sense), the name of the answers were “swedques1”, “swedques2”, “swedques3” and “swedques4” respectively. However, because I did not see a need to change the answers for the rest of the languages as this site is only a prototype, the other language quizzes share this common name. As was mentioned earlier however, the question name can be changed in the HTML and the JavaScript file, and by simply saving this new `.js` file and quoting it in the HTML, the existing answers can be easily changed and removed, and new answers can be created.

The final value included in each of the radio button answers is the “value” tag. This was also explained earlier, as each answer to the question needing to be tagged as so. So, for example, all “1” values are the first answers to each question, but each have different

“names”. These are used for deciding answers in the JavaScript. For more information on how this works, view the table in chapter 8, subsection 8.3.

```
<div id="questions">
<p class="question"><b> Question 1. What is the French word for "cat"? . </b></p>
<ul class="answers">
<label>Chat.</label>
<input type="radio" name="swedques1" value="1"><br>
<label>Chien.</label>
<input type="radio" name="swedques1" value="2"><br>
<label>Lion.</label>
<input type="radio" name="swedques1" value="3"><br>
<label>Serpent.</label>
<input type="radio" name="swedques1" value="4"><br>
</ul>

<p class="question"><b> Question 2. What is the French word for "dog"? . </b></p>
<ul class="answers">
<label>Araignee.</label>
<input type="radio" name="swedques2" value="1"><br>
<label>Serpent.</label>
<input type="radio" name="swedques2" value="2"><br>
<label>Dauphin.</label>
<input type="radio" name="swedques2" value="3"><br>
<label>Chien.</label>
<input type="radio" name="swedques2" value="4"><br>
</ul>

<p class="question"><b> Question 3. What is the French word for "tiger"? . </b></p>
<ul class="answers">
<label>Tigre.</label>
<input type="radio" name="swedques3" value="1"><br>
<label>Chien.</label>
<input type="radio" name="swedques3" value="2"><br>
<label>Chat.</label>
<input type="radio" name="swedques3" value="3"><br>
<label>Abeille.</label>
<input type="radio" name="swedques3" value="4"><br>
</ul>

<p class="question"><b> Question 4. What is the French word for "bird"? . </b></p>
<ul class="answers">
<label>Lapin.</label>
<input type="radio" name="swedques4" value="1"><br>
<label>Serpent.</label>
<input type="radio" name="swedques4" value="2"><br>
<label>Oiseau.</label>
<input type="radio" name="swedques4" value="3"><br>
<label>Chien.</label>
<input type="radio" name="swedques4" value="4"><br>
</ul>
<br>
</div>
```

Figure 100 - Code snippet of HTML used in final site for quiz questions.

After the quiz answers were created, I had to create the element the user would press to submit their answer. This was done by creating a simple div with a paragraph that read “Submit answers” under the div tag “userResults”. This div was made to hover by using CSS, shown later. This div is also the .onclick function that is quoted in the JavaScript, and is responsible for initiating the answer checking process.

```

<div id="userResults">
    Submit answers!
</div>

```

Figure 101 - Code snippet of HTML used in final site for submit button.

After this, I made the divs that would display the correct answers if the user submitted the wrong answer. This was done by creating 5 different divs at the end of the page, and having them all hide themselves by calling on them in the CSS with `#answer1`, `#answer2` etc. Answer divs 1-4 contained the correct answer, and div 5 contained a message that displayed if all answers were correct. Refer to the diagram created in Chapter 8.1 for more information on how this works.

```

<div id="answer1">
<p> Question 1: The correct answer was 1 - Chat. </p><br>
</div>
<div id="answer2">
<p> Question 2: The correct answer was 4 - Chien. </p><br>
</div>
<div id="answer3">
<p> Question 3: The correct answer was 1 - Tigre. </p><br>
</div>
<div id="answer4">
<p> Question 4: The correct answer was 3 - Oiseau. </p><br>
</div>
<div id="answer5">
<p> You answered all French animal questions answered correctly, well done! </p><br>
</div>

```

Figure 102 - Code snippet of HTML used in final site for question answers.

Finally, since the quiz pages have no navbar, and since the header doesn't link back to the homepage, had to create some type of way for the user to leave the page if they suddenly decided they didn't want to take the test without clicking the back button. This was done by introducing two buttons in a div at the bottom of the page. The div is called "returnToHomepage", and features two buttons which use the same `.navButton` class that the buttons in the navbar do. Of these two buttons, one will link to the homepage for user convenience, and the other will link back to the language entry page for the language the quiz is in. For example, the French quiz will have a button leading to the homepage, and one button leading to the French language entry page. This can be seen at the bottom of the HTML code below.

```

<div id="returnToHomepage">
    <a href="mainpage.html" class="navButton">Back to Homepage.</a>
    <a href="french.html" class="navButton">Back to French language area.</a>
</div>

```

Figure 103 - Code snippet of HTML used in final site to return user to different pages.

After the HTML came the CSS. Like the language entry pages, each quiz language has their own CSS file (e.g. `frenchQuiz.css`, `swedishQuiz.css`, `germanQuiz.css` and `spanishQuiz.css`). This allowed for more customisation in each individual language quiz. This advanced customisation ability was used to insert the same flag images used before in the main body as background images again, to easily display to the user what section of the site they were on, as well as providing page style.

```
#main_body {  
    padding: 20px 10px 20px 10px;  
    font-size: 19px;  
    color: solid black;  
    border: 1px solid black;  
    text-align: center;  
    background-color: #B0D7D7;  
    background-image: url("frenchlanguagearea.png");  
}
```

Figure 104 - Code snippet of CSS used in final site.

After this came the styling of the questions. For the questions, I used a lighter version of the font that was used for the main body, to provide a bit of style and emphasise the difference between the questions div and the rest of the page. The font that is used throughout the page is Bahnschrift sans-serif, and the font for the questions is Bahnschrift Light, sans-serif. I also used **text-align** to set the questions to display from left to right, and used letter spacing to make the questions easier to read when displayed. Font size was also manually set to be a bit smaller than the body text (19px body, 17px questions) and also used vertical align to format the questions.

```
#questions {  
    font-family: Bahnschrift Light, sans-serif;  
    color: solid black;  
    text-align: left;  
    vertical-align: middle;  
    border: 1px solid black;  
    font-size: 17px;  
    letter-spacing: 1px;  
    background-color: #B0D7D7;  
}
```

Figure 105 - Code snippet of CSS used in final site.

After the questions were styled, I moved onto the returnToHomepage and userResults divs. The first div contained the return to homepage and language area navbuttons, so I simply aligned the buttons to the center of the div, gave it a small border, added some padding to make sure users know it's a different div, and gave it a background colour that fit in with the rest of the site.

The userResults div was a bit more complex. Because it's a submit button, and because I wanted to draw extra attention to it I made it a **hover** element. What this means is that the div will change how it looks when the user mouses over it. I did this first by styling the div on how it would look when the user is not hovering over it – gave it a small border, background colour the same as the page, aligned the text to the center and gave it some padding. Then, I moved on to how it would look when hovered over. Here, I added the same border and padding but with a different background colour, and used the **cursor** tag to make the cursor change to a pointer, instead of the default cursor. This means now that when the user hovers over the userResults div (which is the submit button), the div will turn blue and the cursor will change.

Finally, I had to style the answer 1-5 divs and the “closeAnswers” div. Divs 1-5 were easy; they all had to explicitly not show until the user submitted their answers, at which point the JavaScript would push out (read: display) the appropriate divs. To do this, I simply used the **display** tag and set it to “none”, meaning the div would not display. I also set the div to have no border, or else it would just appear as a blank div with a small 1px black border. The closeAnswer div does not work as intended – it is supposed to hide the answers when the user adjusts their answers and resubmits to get all answers correct. It has been left in so I can try and get it working before the deadline, but if it does not work it will simply be left in, as it does not affect the code in its current state – but it’s removal may.

While the navbar does not exist on the question pages, buttons still do (in the returnToHomepage div), so the .navButton lines must stay in to style the buttons.

```
#userResults {  
    text-align: center; /* this was done to center t!  
    background-color: #9D9694;  
    border: 1px solid black;  
    padding: 10px 10px 10px 10px;  
}  
#userResults:hover {  
    background-color: #B0D7D7;  
    text-align: center;  
    cursor: pointer; /* sets the pointer to be a cur:  
    border: 1px solid black;  
    padding: 10px 10px 10px 10px;  
}  
#answer1, #answer2, #answer3, #answer4, #answer5 {  
    display: none;  
    border: none;  
}  
#closeAnswers {  
    display: none;  
    border: none;  
}  
.navButton {  
    border: 1px solid black;  
    padding: 6px 8px 8px 8px;  
    color: black;  
    font: bold 100px;  
    text-decoration: none;  
    background-color: #97BDC9;  
}
```

Figure 106 - Code snippet of HTML used in final site.

Lastly, here is what the quiz pages look like in browser.

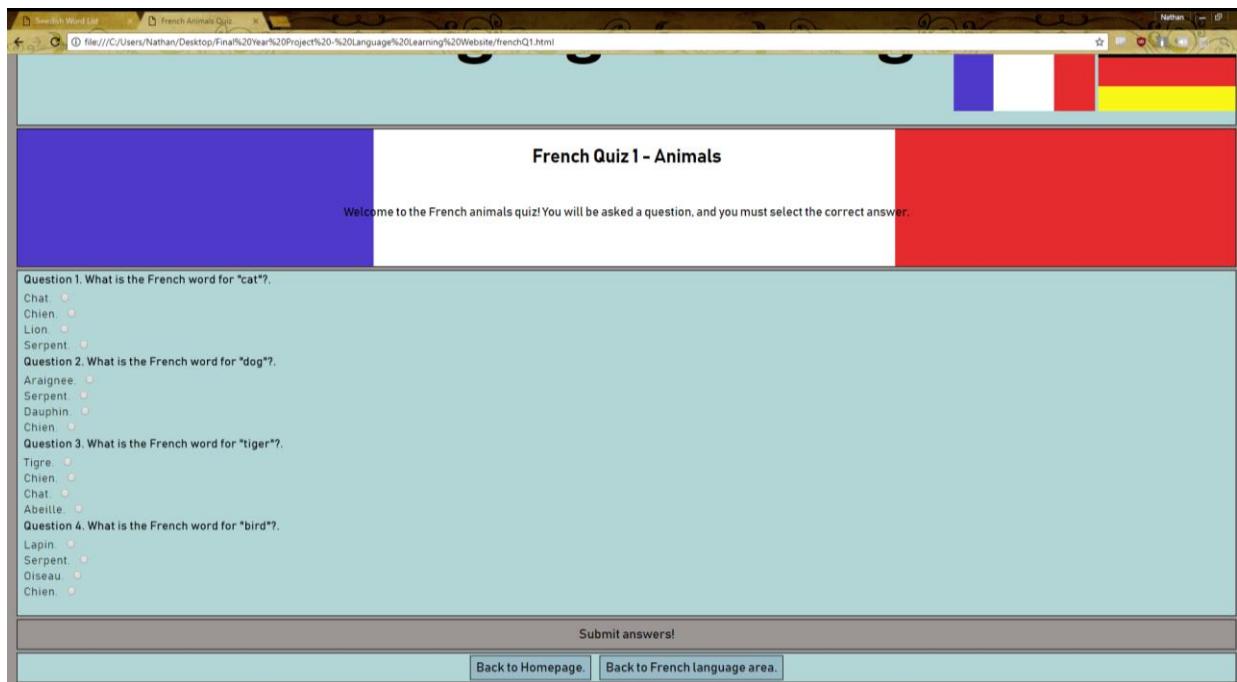


Figure 107 - Submit button not highlighted in Google Chrome.

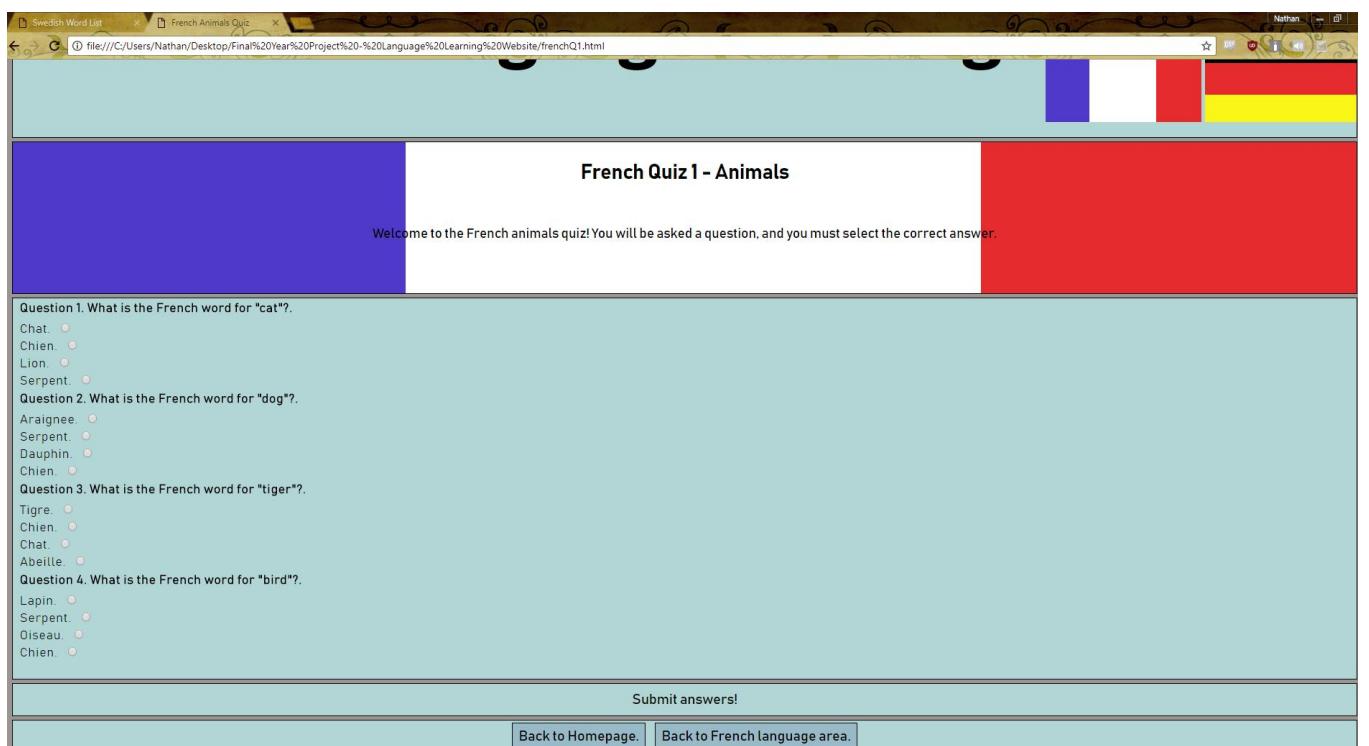


Figure 108 - Submit button highlighted in Google Chrome.

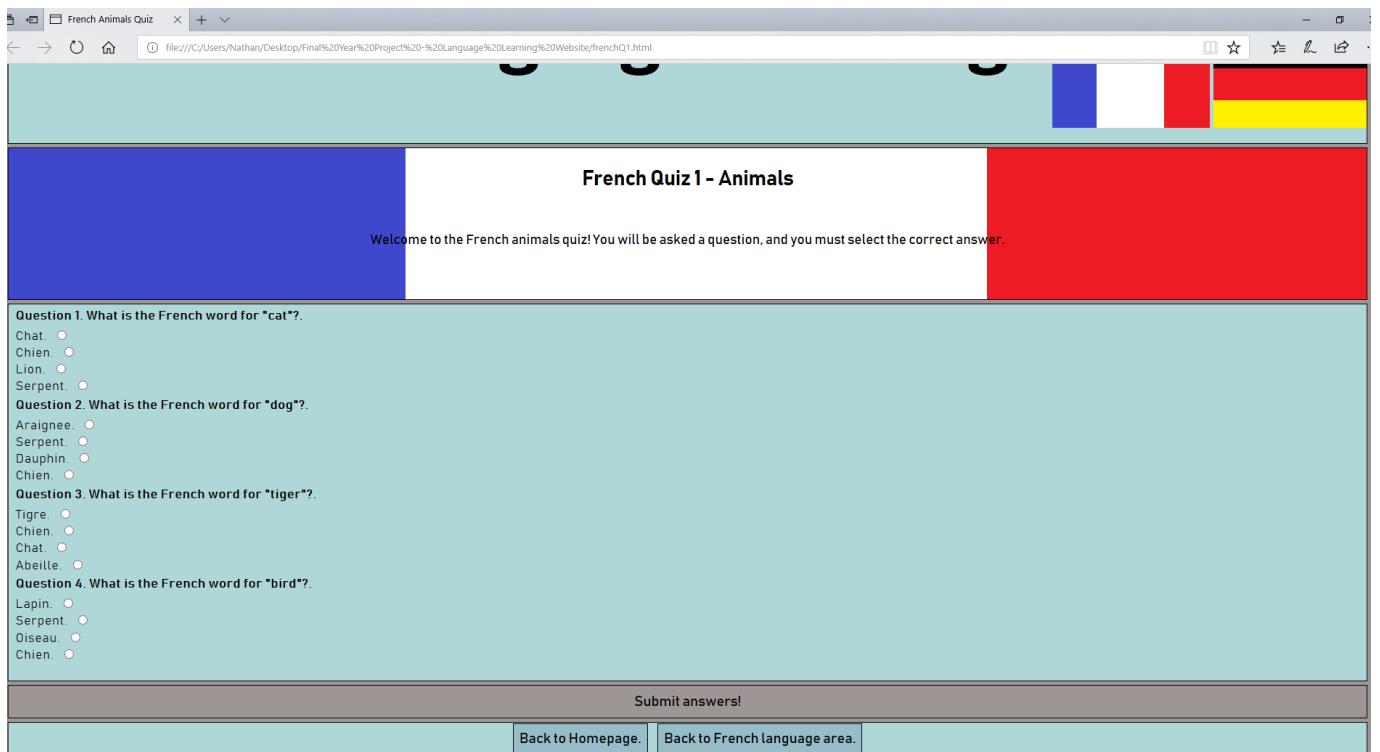


Figure 109 - Submit button not highlighted in Microsoft Edge.

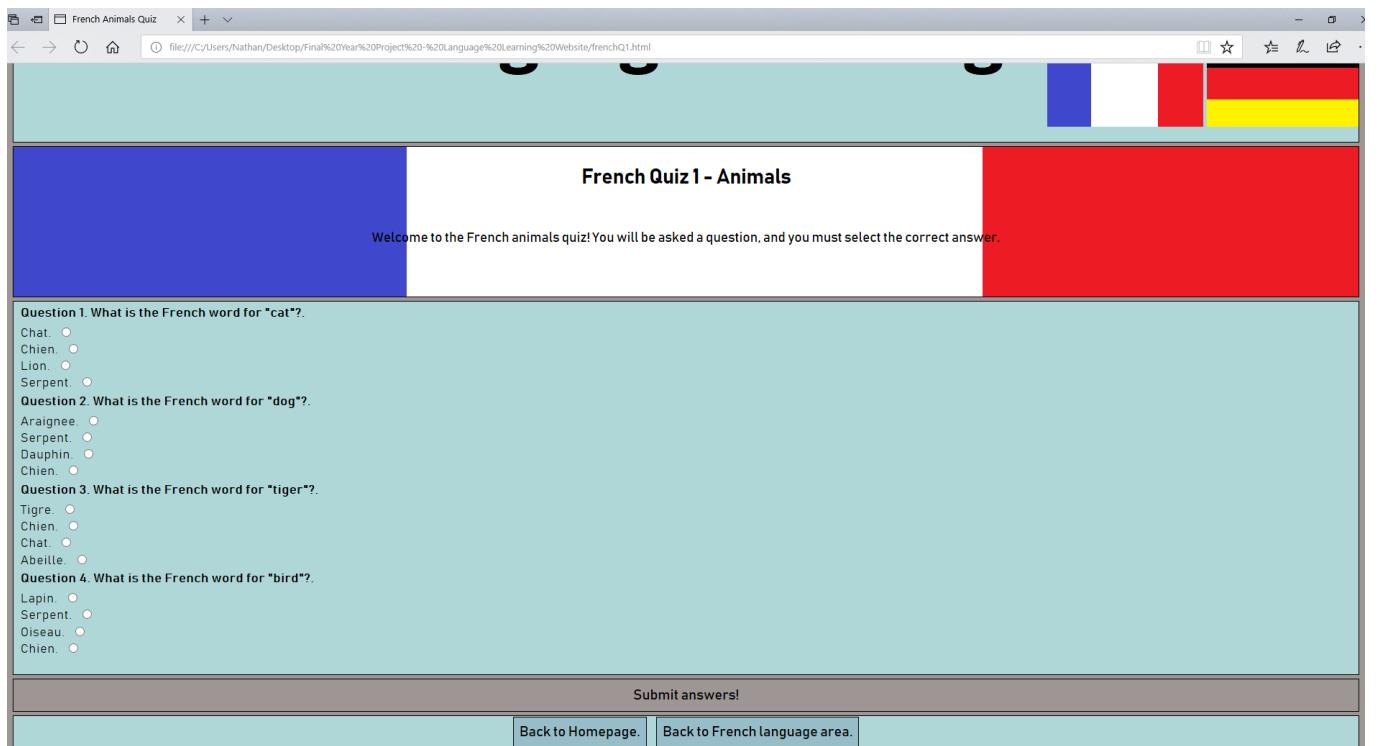


Figure 110 - Submit button highlighted in Microsoft Edge.

Microsoft Edge (submit button highlighted).

9.3.6 Word List Pages

Finally, I had to complete the word list pages. These pages exist so that users can first be aware of what words they need to know before completing the test. They are tasked with memorising them, and then testing their skills in the quiz.

First came the HTML. Similarly to the language areas that have come before, each language has its own wordlist rather than having all languages on one giant set of lists. I used the table tag extensively in the HTML to create four different tables that contained every English word along with their foreign translation counterpart. The tables were created in the order that the topics appear on the language entry page. The HTML code was taken from the homepage.html, but had the returnToHomepage div at the bottom, and had the main body completely repurposed to store tables.

The tables are created by first creating the two table headers using the tag <th>, for example in the Swedish wordlist these are

<th> English Animals </th>

<th> Swedish Translation </th>

After this, comes the creation of the row, and the cells inside them, shown below with the animals table.

```
</div>
<div id="main_body">
    <h2> Swedish Word List </h2>
    <p> Remember, if you want to say "the cat" instead of "a cat", you must end the word in "en"! In this example, the word "Katt" would become "Katten". </p>
    <table style="float: left;">
        <tr>
            <th> English Animals </th>
            <th> Swedish Translation </th>
        </tr>
        <tr>
            <td> Cat </td>
            <td> Katt </td>
        </tr>
        <tr>
            <td> Mouse </td>
            <td> Mus </td>
        </tr>
        <tr>
            <td> Animal </td>
            <td> Djur </td>
        </tr>
        <tr>
            <td> Dog </td>
            <td> Hund </td>
        </tr>
        <tr>
            <td> Bird </td>
            <td> Fågel </td>
        </tr>
        <tr>
            <td> Spider </td>
            <td> Spindel </td>
        </tr>
        <tr>
            <td> Horse </td>
            <td> Häst </td>
        </tr>
        <tr>
            <td> Bear </td>
            <td> Björn </td>
        </tr>
        <tr>
            <td> Elk </td>
            <td> Älg </td>
        </tr>
    </table>
```

Figure 111 - HTML depicting wordlists with Swedish words and English translations.

As is visible in the above image, also created was a single paragraph line designed to help the user with grammar-centred answers. This single line is also present in the French and Spanish translation lists, as they are needed for the users to be able to guess correctly. All tables look like the one above, except with the translations and table header tags changed according to the topic.

Also present in the tables was my decision to make the style the tables so they floated left. Without this styling in the table HTML, the translations displayed as a long list that stretched the page out to inefficient proportions. I justified this choice by asserting that it appeared better both stylistically and in terms of usability to have the word lists displayed in a row, from left to right, much like the topics are in the language entry pages.

```
<table style="float:left;">
:
:
```

Figure 112 - Code snippet relating to table styling.

Next, came the work with the CSS. In contrast to the modularity I had decided on earlier, I decided to only have one CSS file for the wordlist, aptly named the **wordlist.css**. This was because the pages did not need much in the way of individual styling, and it became much easier to modify every table on every page at once, instead of creating four times the workload for myself for unnecessary reasons.

The styling of the wordlist is largely the same as the main.css for the most part, except for in the main body, where I had to manually set the divs height to 850px, for it to keep all the tables inside it without any of them overflowing due to the tables floating. I feel this styling kept the tables very user friendly, and aesthetically pleasing as a bonus.

```
#main_body {
    font-family: Bahnschrift Light, sans-serif;
    padding: 20px 10px 20px 10px;
    color: solid black;
    border: 1px solid black;
    text-align: left;
    background-color: #B0D7D7;
    font-size: 18px;
    height:850px;
}
```

Figure 113 - CSS code used in final site.

After this, I styled the tables. To keep the theme consistent, I changed the font-type to the one used throughout the site (Bahnschrift). I also styled the text in the table headers and table data cells to align to the left, and collapsed the border to make the border lines split the translations and the actual English word, to increase readability. Without this attribute, the tables did not have line down the Y axis, and made the translations hard to read. I individually styled the table rows to have a height of 35 px, and styled the table headers to have a darker background colour, 30px height and 160px width.

```

table, td, th {
    font-family: Bahnschrift, sans-serif;
    text-align: center-left;
    border: 1px solid black;
    border-collapse: collapse;
}
tr {
    height: 35px;
}
th {
    height: 30px;
    width: 160px;
    background-color: #97BDC9;
}

```

Figure 114 - CSS code used to style table headers, rows and columns.

Below is what the word list pages look like in browser, they are displayed in the order of; Swedish, French, Spanish, German. All screenshots are taken in Google Chrome.

Swedish Word List

Remember, if you want to say "the cat" instead of "a cat", you must end the word in "en"! In this example, the word "Katt" would become "Katten".

English Animals	Swedish Translation	English Numbers	Swedish Translation	English Colours	Swedish Translation	English Food	Swedish Translation
Cat	Katt	One	En/Ett	Yellow	Gul	Fish	Fisk
Mouse	Mus	Two	Två	Red	Röd	Egg	Ägg
Animal	Djur	Three	Tre	White	Vit	Chicken	Kyckling
Dog	Hund	Four	Fyra	Orange	Orange	Orange	Apelsin
Bird	Fågel	Five	Fem	Purple	Lila	Coffee	Kaffe
Spider	Spindel	Ten	Tio	Blue	Blå	Fruit	Frukt
Horse	Häst	Eleven	Elv	Brown	Brun	Breakfast	Frukost
Bear	Björn	Twelve	Tolv	Green	Grön	Sugar	Socker
Elk	Alg	Thirteen	Tretton	Grey	Grå	Beef	Nötkött
		Fourteen	Fjorton	Black	Svart	Strawberry	Jordgubbe
		Fifteen	Femton			Lemon	Citron
		Twenty	Tjugo				
		Thirty	Trettio				
		Fourty	Fyrto				
		Fourty One	Fyrtoett				
		Fourty Two	Fyrtoetvå				
		Fifty	Femtio				
		Sixty	Sextio				
		Seventy	Sjuttio				
		Eighty	Åttio				
		Eighty Nine	Åttionio				

[Back to Homepage](#) | [Back to Swedish language area.](#)

Figure 115 – Swedish wordlist in Google Chrome.

French Word List

In the colours section, the masculine word is translated first, followed by the feminine one.

English Animals	French Translation	English Numbers	French Translation	English Colours	French Translation	English Food	French Translation
Cat	Chat	One	Un	Yellow	Jaune	Tomato	Tomate
Lion	Lion	Two	Deux	Red	Rouge	Baguette	Baguette
Snake	Serpente	Three	Trois	White	Blanc / Blanche	Onion	Oignon
Dog	Chien	Four	Quatre	Orange	Orange	Egg	Oeuf
Bird	Oiseau	Five	Cinq	Purple	Violet / Violette	Chocolate	Chocolat
Spider	Araignee	Ten	Dix	Blue	Bleu / Bleue	Salad	Salade
Bee	Abelille	Eleven	Onze	Brown	Marron	Water	Eau
Dolphin	Dauphin	Twelve	Douze	Green	Vert / Verte	Coffee	Cafe
Tiger	Tigre	Thirteen	Treize	Grey	Gris / Grise	Banana	Banane
Rabbit	Lapin	Fourteen	Quatorze	Black	Noir / Noire	Tea	The
		Fifteen	Quinze			Cheese	Fromage
		Sixteen	Seize			Crepe	Crepe
		Seventeen	Dix-sept				
		Eighteen	Dix-huit				
		Nineteen	Dix-neuf				
		Twenty	Vingt				
		Thirty	Trente				
		Fourty	Quarante				
		Fifty	Cinquante				
		Sixty	Soixante				
		Sixty One	Soixante et un				

[Back to Homepage.](#) [Back to French language area.](#)

Figure 116 - French wordlist in Google Chrome.

Spanish Word List

In the colours section, the masculine word is translated first, followed by the feminine one.

English Animals	Spanish Translation	English Numbers	Spanish Translation	English Colours	Spanish Translation	English Food	Spanish Translation
Cat	Gato	One	Uno	Yellow	Amarillo / Amarilla	Rice	Arroz
Tiger	Tigre	Two	Dos	Red	Rojo / Roja	Orange	Naranja
Animal	Animal	Three	Tres	White	Blanco / Blanca	Chicken	Pollo
Dog	Perro	Four	Cuatro	Orange	Naranja	Meat	Carne
Butterfly	Mariposa	Five	Cinco	Purple	Morado / Morada	Cabbage	Col
Elephante	Elefante	Ten	Diez	Blue	Azul	Apple	Manzana
Horse	Caballo	Eleven	Once	Green	Verde	Lemon	Limon
Bear	Oso	Twelve	Doce	Grey	Gris	Coffee	Cafe
Giraffe	Jirafa	Thirteen	Trece	Black	Negro / Negra	Pear	Pera
Snake	Serpiente	Fourteen	Catorce			Grapefruit	Pomelo
Lizard	Lagartija	Fifteen	Quince				
		Twenty	Veinte				
		Twenty Five	Veinticinco				
		Twenty Six	Veintiseis				
		Twenty Seven	Veintisiete				
		Twenty Eight	Veintiocho				
		Thirty	Treinta				
		Fourty	Cuarenta				
		Fifty	Cincuenta				
		Sixty	Sesenta				
		Seventy	Setenta				

[Back to Homepage.](#) [Back to Spanish language area.](#)

Figure 117 - Spanish wordlist in Google Chrome.

English Animals	German Translation	English Numbers	German Translation	English Colours	German Translation	English Food	German Translation
Cat	Katze	One	Eins	Yellow	Gold	Fish	Fisch
Mouse	Maus	Two	Zwei	Red	Rot	Apple	Apfel
Rabbit	Kaninchen	Three	Drei	White	Weiß	Chicken	Hähnchen
Dog	Hund	Four	Vier	Orange	Orange	Ham	Schinken
Monkey	Affe	Five	Fünf	Purple	Violett	Banana	Banane
Snake	Schlange	Ten	Zehn	Blue	Blau	Meat	Fleisch
Hamster	Hamster	Eleven	Elf	Pink	Rosa	Onion	Zwiebel
Fish	Fisch	Twelve	Zwölf	Green	Grün	Rice	Reis
Chicken	Huhn	Thirteen	Dreizehn	Brown	Braun	Sausage	Wurst
Horse	Pferd	Fourteen	Vierzehn	Black	Schwarz	Peas	Erbse
Goat	Ziege	Fifteen	Fünfzehn	Grey	Grau	Cheese	Käse
		Twenty	Zwanzig			Raspberry	Himbeere
		Thirty	DreiBzig				
		Fourty	Vierzig				
		Fourty One	Einundvierzig				
		Fourty Two	Zweiundvierzig				
		Fourty Three	Dreiundvierzig				
		Fifty	Fünfzig				
		Sixty	Sechzig				
		Seventy	Siebzig				

[Back to Homepage.](#) [Back to German language area.](#)

Figure 118 - German wordlist in Google Chrome.

Chapter 10. Testing

The purpose of this chapter is to test individual functions on the site to see if everything is working, and working as intended. This will be done using a test plan, which will take the form of a large table full of test cases. The table will have 5 sections – the test number, a description of the feature / function, the relevant test data or element being manipulated, and the actual outcome.

10.1 Test Cases

Test No.	Test Description	Manipulated Element	Expected Outcome	Real Outcome
1	“Back home” button takes user to homepage.	Back home navButton	Navigate to homepage	Navigated to homepage
2	Header image takes user to homepage.	Image header	Navigate to homepage	Navigated to homepage
3	“About” button takes user to about page.	About navbutton.	Navigate to about page.	Navigated to about page.
4	“Contact” button takes user to contact page.	Contact navbutton.	Navigate to contact page.	Navigated to contact page.
5	“Swedish” button takes user to Swedish language entry page.	Swedish navbutton.	Navigate to Swedish language area page.	Navigated to Swedish language area page.

6	“French” button takes user to French language entry page.	French navbutton.	Navigate to French language area page.	Navigated to French language area page.
7	“Spanish” button takes user to Spanish language entry page.	Spanish navbutton.	Navigate to Spanish language area page.	Navigated to Spanish language area page.
8	“German” button takes user to German language entry page.	German navbutton.	Navigate to German language area page.	Navigated to German language area page.
9	In all language pages, “Animals” button takes user to animals quiz.	Animals navbutton.	Navigate to animals quiz.	Navigated to animals quiz.
10	In all language pages, “Numbers” button takes user to numbers quiz.	Numbers navbutton.	Navigate to numbers quiz.	Navigated to numbers quiz.
11	In all language pages, “Colours” button takes user to colours quiz.	Colours navbutton.	Navigate to colours quiz.	Navigated to colours quiz.
12	In all language pages, “Food” button takes user to food quiz.	Food navbutton.	Navigate to food quiz.	Navigated to food quiz.
13	In language page, “Wordlist” button takes user to wordlist.	Wordlist navbutton.	Navigate to wordlist.	Navigated to wordlist.
14	Able to submit answers to all animals quizzes.	Submit answer button.	Selected answers submitted.	Selected answers submitted.
15	Able to submit answers to all numbers quizzes.	Submit answer button.	Selected answers submitted.	Selected answers submitted.
16	Able to submit answers to all colours quizzes.	Submit answer button.	Selected answers submitted.	Selected answers submitted.

17	Able to submit answers to all food quizzes.	Submit answer button.	Selected answers submitted.	Selected answers submitted.
18	Back to homepage button at bottom of screen works.	Back to homepage navbutton.	Navigate to homepage.	Navigated to homepage.
19	Back to language area button at bottom of screen works.	Back to target language area navbutton.	Navigate to target language area.	Navigated to target language area.
20	Quiz radio buttons can be selected.	All radio buttons in all quizzes.	All radio buttons can be selected.	All radio buttons can be selected
21	Error message appears if submit button is pressed with missing inputs.	Submit answers button in all quizzes.	Error message appears on page if input is missing.	Error message appears on page when input is missing.
22	Correct divs display according to answers user got correct / incorrect.			
23	Incorrect answer divs are closed when user corrects answers.	Radio buttons, submit button.	Incorrect divs are closed when answer is correct, message displays telling user all answers correct.	Message displays telling user all answers are correct, incorrect answer divs still present.

Figure 119 - Table of use cases.

Chapter 11. Evaluation

This chapter will serve as a summary of the project, and will evaluate whether StanLearn achieved the goal it was created for. It will evaluate the sites aesthetic, usability and detail whether or not the way the quiz and site overall functions to serve its purpose.

11.1 Site Aesthetics and Usability

The overall aesthetics of StanLearn are not highly stylised. While the project itself is and always has been created to be a prototype, this was no excuse to skimp on the way the site was styled. One reason behind this was because I felt there was a considerable link between site aesthetics and site usability. Too many cluttered and brightly coloured elements I felt would impact the way the site appears professionally and to other users.

Important elements are displayed clearly, the navigation bar does not include too many options and is not so expansive to the point the user would become too confused. There are no pages that are too long and difficult to scroll through, there are no elements overflowing or moving to places I have not otherwise told them to move to. I was exceptionally

considerate with my use of site graphics and colours, with the idea of making it apparent to users what area of the site they were on without having to read text.

For example, for each of the language entry areas I included the flag associated with the nation where that language is the majority / official language. This results in users of StanLearn only having to look at the graphic behind the text to find out what area of the site they are on without reading sections of text. While this feature is not particularly considerate to those with visual impairments such as colour-blindness, I felt it would be impossible to create an at least basic-styled site and accommodate for all impairments, so I sacrificed some usability for aesthetics in that regard.



Figure 120 - Swedish language area graphic.

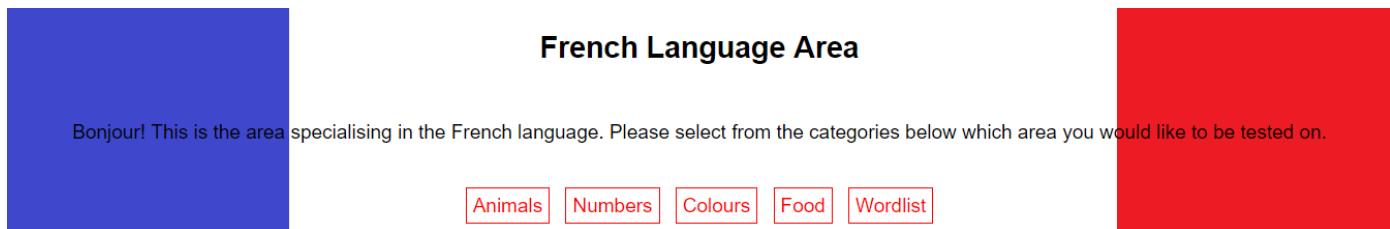


Figure 121 - French language area graphic.

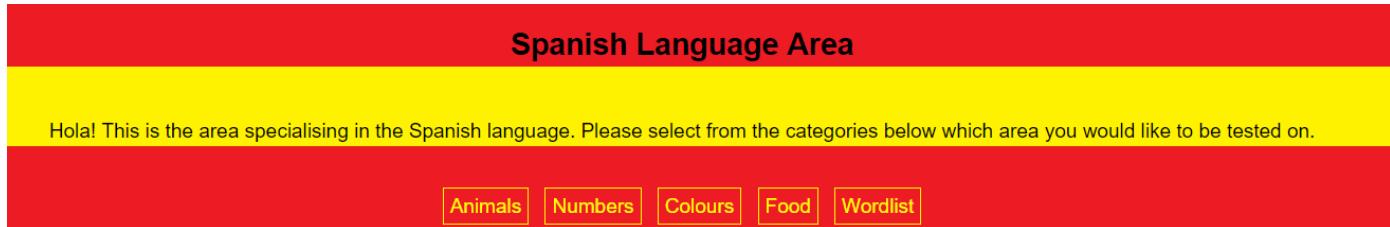


Figure 122 - Spanish language area graphic.

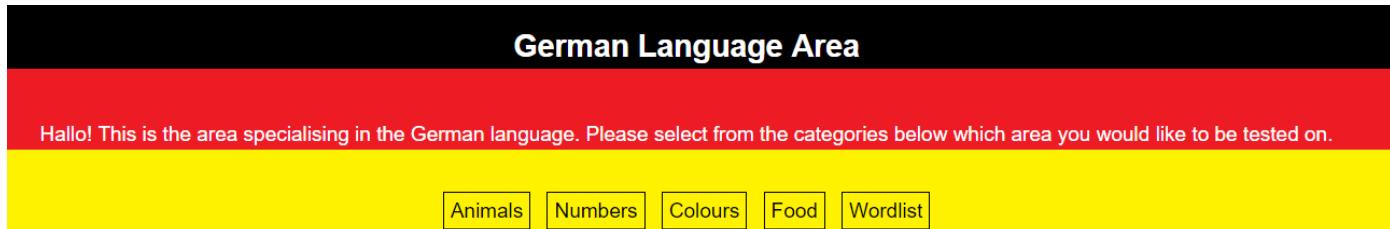


Figure 123 - German language area graphic.

In addition, all text on the site is displayed in a clear, readable font at different sizes. The gap between font sizes in areas is only several pixels, with the smallest being in the 16-17 pixels range, and the largest around 19pixels. While this does of course aid site aesthetics, this was done to enhance site usability, especially because StanLearn was created to accommodate all ages, including older people who may have deteriorated sight.

Overall, I feel the way the site was designed and the way the solutions for usability were deployed complimented each other nicely, and made for a positive user experience despite age and gender barriers that most sites have trouble passing.

11.2 Site Functions and Purpose Fulfilment

With regards to the site's purpose – it functions exactly as I intended it to. From the early design stages right up until finalising the layout and implementation, I consulted the feedback gained from the questionnaire completed at the start of the academic year, and stuck by the user suggestions, everything from the most suggested languages to the way the site tested users (the remarked "fill in the blank" approach), everything has kept true to my early designs. While the sites effectiveness at teaching a language cannot be tested and quantified now, it certainly follows examples and uses similar methods to other similar products currently available, and so it is a regularly safe assumption that StanLearn does what they do, albeit on a simpler level.

The user is presented with the opportunity to view translations, memorise the words in any way they prefer, and then take a quiz in one of four topics to test those memorisation skills. Considering all opportunities offered and all elements in the site, StanLearn should fulfil its current goal of introducing users to a new MFL at a basic level.

11.2.1 Strengths

While the strengths and weaknesses of the site are susceptible to my own biases, I will try to include all shortcomings and strengths of the site as objectively as possible.

- Provides a functioning platform for users to learn new words and then test their knowledge on said words.
- Provides a service designed from its inception to be indiscriminate of age, gender and computing ability.
- Provides the ability to learn the basics of four languages – even in its prototype stage.

11.2.2 Weaknesses

- Does little else besides ask users to memorise words, and then tests them on these words.
- Does not contain user-requested soundbites.
- Does not contain video aid.
- Site does not follow the effective flashcard approach as other products like Duolingo and Memrise do.
- Functionality is simple, and much more could be done to enhance user experience and progress – tracking (e.g. account creation, progress bar, levelling system).
- Only offers four basic European languages.

11.3 Significance and What Areas Could be Further Explored

While what StanLearn is offering on the surface may not differ much from what is offered by other MFL-teaching mediums, its effort to include more demographics and emphasise teaching people of all ages for free signifies a considerable step forward in trying to bring language learning ability to the masses, and help to meet its sub goal of raising the overall MFL literacy rate by any amount.

Given that StanLearn is very basic, there are thankfully a lot of further areas that could be explored. These include

- Soundbites from native speakers (or those fluent in the language).

- Video clips of people speaking said language.
- A progress tracking system.
- The use of a database to store user progress.
- A levelling system.
- Further accommodations for hearing / visually impaired users.
- More languages and translations.

I would like to think that StanLearn has achieved the main goal it was created to achieve, and has provided more people with the opportunity to develop an interest in language, as well as the opportunity to enhance cognitive abilities alongside a possible attainment of benefits to their health.

11.4 Evaluation Methods

The evaluation methods used were basic, but attempted to be as thorough as possible. They consisted of using test cases to use each part of the site, and test if they were functional and worked as intended. These can be seen for more detail in chapter 10, subsection 1 (10.1).

In the case of this project, it must be assumed that testing is evaluation, as there was no audience to roll this project out to, and it instead rested upon the developer to test if the site was functional and met all goals. The testing methods I used did result in testing all areas of the site, and tested if they were functional. Seeing as the site meets all the goals it sets out to do on an objective level, it is safe to infer that the site fulfils its purpose to attempt to teach users a new MFL, and therefore lead to the conclusion the sites purpose has been evaluated and considered a success.

The only other measurement that could be performed is to expose a diverse set of users to the project over a set period, and ask them if they can recite the words they have learned from the site, along with their translations. This is, however, completely out of this projects scope and time frame, and so could only be attempted past the project's deadline in external uncontrolled tests.

Chapter 12. Conclusions and Further Work

This chapter will aim to summarise my work, as well as explain a few issues beginning and throughout the projects development. I think StanLearn has turned out to be a very successful project, despite the given time constraints and a few too many development iterations – I feel I produced an adequate piece of work in the time given.

To start, I did have a few issues with the time constraints. While working to a deadline is task I find easy, it was particularly difficult to keep up when balancing my project alongside other work throughout the year. I feel that in the time frame given, I was able to build a website from the ground up that looks genuine and professional, provides a medium that offers a service that is navigable and usable, and integrate it with functionality that inefficient and effective.

In terms of the problem area that StanLearn tackles, I feel has been covered nicely in this report. I reinforce regularly the fact that StanLearn is an inclusive service that extends its offers to as many people as is willing to receive it. It tackles the problem of certain demographics not engaging themselves in language learning, and also aims to draw in these demographics at a younger age, so they can gain full access to the benefits that learning a new MFL brings, as is detailed in the literature review (chapter 3).

The results are currently hard to quantify, as the project has not yet been shown to any testers, nor has it been graced with the timeframe to be exposed for general use and gain

feedback on its effectiveness. Using test cases, I am currently able to infer that since the site is operational, and since all elements I aimed to include work as intended and are similar to services offered by other language learning mediums, that StanLearn is an effective site to help users become exposed and gain basic skills in French, German, Spanish or Swedish. In doing so, I hope StanLearn can be another medium which generates interest in MFL learning, and leads to a better connected and hopefully more tolerant global population.

Now that StanLearn's current stage has been brought to its conclusion, after writing this report I know now there are many ways in which it can be improved, to reach a wider audience and / or provide a better, more efficient and more interactable service for the users it currently has. For example, the inclusion of better visuals in the form of flashcards would be the next logical step for its development, along with audio and visual cues that would bring it up to par with the services offered by other websites and apps of its kind. Once this is done, the enhancement of the user experience with tools like account creation, progress tracking and a chat service would make StanLearn an up-and-coming potential competitor to the language learning giants like Memrise and Duolingo. Once StanLearn is at this stage, research can be conducted to deduce what its competitors are not doing in the language teaching sector that it could pursue, and could potentially become a successful site that becomes the new standard for learning for people of all ability levels – including computing ability levels and language knowledge ability level, with the addition of advanced language learning not being out of the question for a service of its size in the future.

Or, instead of simply becoming like other items on the market, StanLearn could be used as a research tool to expose people to learning a new MFL and quantify the effects later in life, to potentially contribute to scientific research where it may either reinforce or disprove established hypotheses that language learning has significant positive cognitive and mental health impact.

Chapter 13. Project Management

This final section documents all monthly meetings between me and my project supervisor that were attended, as well as information about what was discussed each month, in addition to what was to be discussed the next month and the production of deliverables.

13.1 November Meeting

6001PROJ Final Year Project

Monthly Supervision Meeting Record

Student: Nathan Stanley
..... Date: November

Main issues / Points of discussion / Progress made
First part of literature review. Review of existing language learning websites and apps. Discussion of questionnaire that was created and distributed, awaiting replies.
Actions for the next month

Begin situational / problem analysis.

Deliverables for next time

Situational analysis first draft.
Problem analysis first draft.

Other comments

None.

Supervisor signature: Denis Reilly

Student signature: Nathan Stanley

13.2 December Meeting**6001PROJ Final Year Project
Monthly Supervision Meeting Record**

Student: Nathan Stanley
..... Date: December

Main issues / Points of discussion / Progress made

Discussion of progress in problem analysis.
Discussion of progress in situational analysis.
Current state of literature review.
Format and structure of final year report.

Actions for the next month

Complete literature review.
Finalise and refine problem / situational analysis with suggestions discussed.

Deliverables for next time
Final literature review and problem / situational analysis. Final year report basic structure beginnings.
Other comments
None

Supervisor signature: Denis Reilly.....

Student signature: Nathan Stanley

13.3 January Meeting

6001PROJ Final Year Project

Monthly Supervision Meeting Record

Student: Nathan Stanley
..... Date: January

Main issues / Points of discussion / Progress made
Discussion of how website is going to work. Discussion of what software to use. Discussion of final year project HTML and CSS.
Actions for the next month
Begin construction of site Java.
Deliverables for next time
Storyboards for next month. HTML and CSS skeleton.

Other comments
None.

Supervisor signature: Denis Reilly

Student signature: Nathan Stanley

13.4 February Meeting

6001PROJ Final Year Project Monthly Supervision Meeting Record

Student: Nathan Stanley
..... Date: February

Main issues / Points of discussion / Progress made
Discussion of progress made in HTML and CSS of final year site. Discussion of progress in JavaScript the site will use. Problems with site div placement. Problems with navigation bar.
Actions for the next month
Produce prototype of working site.
Deliverables for next time
Prototype of working site. Site adequately styled.
Other comments

None

Supervisor signature: Denis Reilly.....

Student signature: Nathan Stanley

13.5 March Meeting

6001PROJ Final Year Project

Monthly Supervision Meeting Record

Student: Nathan Stanley
..... Date: March

Main issues / Points of discussion / Progress made

Discussion of final Java code used for site.
Discussion of final stylistic choices for site.
Comments and suggestions for optimising site layout.
Discussion of progress made in report.
Discussion of each element of the report, what is required, and how to fill them out.

Actions for the next month

Finish off all questions in site for each language.
Insert problem analysis and literature review into site.
Begin working more on report so it can be finished fully before deadline.

Deliverables for next time

None.

Other comments

None.

Supervisor signature: Denis Reilly

Student signature: Nathan Stanley

Gantt chart

References

1. <https://www.statista.com/chart/3579/europe-s-most-useful-foreign-languages/> (last accessed 5th April 2018.)
2. <http://termcoord.eu/2014/05/scandinavian-languages-mutually-understandable/> (last accessed 5th April 2018.)
- 3, 13. http://ec.europa.eu/eurostat/statistics-explained/index.php/Foreign_language_skills_statistics#Number_of_foreign_languages_known (accessed 4th January 2018)
4. <https://www.fluentu.com/blog/advantages-of-learning-a-foreign-language/> (last accessed 5th April 2018.)
- 5, 8. S.Hayakawa, et al.(2017) *Thinking More or Feeling Less? Explaining the Foreign-Language Effect on Moral Judgement*
- 6, 9. M.Perquin, et al.(2013) *Lifelong Exposure to Multilingualism: New Evidence to Support Cognitive Reserve Hypothesis.*
7. <http://making.duolingo.com/which-countries-study-which-languages-and-what-can-we-learn-from-it> (last accessed 30th March 2018)
7. <https://en.wikipedia.org/wiki/Duolingo> (last accessed November 2017)
10. <http://www.pewresearch.org/fact-tank/2013/09/05/what-is-the-future-of-spanish-in-the-united-states/> (accessed 4th January 2018)
11. <https://www.bls.gov/ooh/media-and-communication/interpreters-and-translators.htm#tab-6> (last accessed 14th April 2018)
12. <http://www.studentsabroad.com/handbook/why-learn-a-language.php?country=Israel> (accessed 3rd January 2018)
14. <https://slator.com/industry-news/german-translation-interpretation-industry-posts-strong-job-growth/> (accessed 3rd January 2018) (Statistics and sources come from Statista, but the source is in German. The article in question quotes the numbers and extrapolates the data in English).
15. <https://www.theguardian.com/money/2010/aug/28/learning-foreign-language-boost-career> (accessed 4th January 2018)
16. http://ec.europa.eu/commfrontoffice/publicopinion/archives/ebs/ebs_386_en.pdf (accessed 3rd January 2018)
- 17, 18, 19, 23. <https://www.ethnologue.com/statistics/size> (last accessed 10th April 2018)
- 17, 18, 19, 23. https://en.wikipedia.org/wiki/List_of_languages_by_total_number_of_speakers (last accessed 10th April)
20. <https://inews.co.uk/news/education/welsh-lessons-duolingo-wales/> (last accessed 6th April 2018)
21. <https://www.ons.gov.uk/peoplepopulationandcommunity/populationandmigration/populationestimates/bulletins/2011censuskeystatisticsforwales/2012-12-11> (last accessed 6th April 2018)
- 22, 26. <http://esol.britishcouncil.org/content/learners/skills/reading/british-worst-learning-languages> (accessed 4th January 2018)

23. <http://uk.pcmag.com/education-reference-products/8711/guide/the-best-language-learning-software-of-2018> (last accessed 25th March 2018)
24. <https://www.reuters.com/article/us-millennials-poll/teens-more-resilient-tech-savvy-than-older-millennial-study-idUSBRE95I1J420130619> (accessed 10th April 2018)

Bibliography

<https://ielanguages.com/swedish-colors.html> (used to fill in knowledge-gaps in website, accessed 25th March 2018.)

<http://www.frenchnumbers.org.uk> (used to fill in knowledge-gaps in website, accessed 24th March 2018.)

<http://www.bbc.co.uk/schools/primarylanguages/french/families/colours/> (used to fill in knowledge-gaps in website accessed 25th March 2018.)

Ylva Olausson(2015) *Swedish Tutor – Grammar and Vocabulary Workbook*. ISBN 978-1-473-60441-4

Appendices

6001PROJ Final Year Project Monthly Supervision Meeting Record

Student: Nathan Stanley
..... Date: December

Main issues / Points of discussion / Progress made
Discussion of progress in problem analysis. Discussion of progress in situational analysis. Current state of literature review. Format and structure of final year report.
Actions for the next month
Complete literature review. Finalise and refine problem / situational analysis with suggestions discussed.

Deliverables for next time
Final literature review and problem / situational analysis. Final year report basic structure beginnings.

Other comments
None

Supervisor signature: Denis Reilly

Student signature: Nathan Stanley

6001PROJ Final Year Project Monthly Supervision Meeting Record

Student: Nathan Stanley
..... Date: February

Main issues / Points of discussion / Progress made
Discussion of progress made in HTML and CSS of final year site. Discussion of progress in JavaScript the site will use. Problems with site div placement. Problems with navigation bar.
Actions for the next month
Produce prototype of working site.
Deliverables for next time
Prototype of working site. Site adequately styled.
Other comments
None

--

Supervisor signature: Denis Reilly

Student signature: Nathan Stanley

6001PROJ Final Year Project

Monthly Supervision Meeting Record

Student: Nathan Stanley
..... Date: January

Main issues / Points of discussion / Progress made
Discussion of how website is going to work. Discussion of what software to use. Discussion of final year project HTML and CSS.
Actions for the next month
Begin construction of site Java.
Deliverables for next time
Storyboards for next month. HTML and CSS skeleton.
Other comments
None.

Supervisor signature: Denis Reilly

Student signature: Nathan Stanley

6001PROJ Final Year Project

Monthly Supervision Meeting Record

Student: Nathan Stanley
..... Date: March

Main issues / Points of discussion / Progress made
Discussion of final Java code used for site. Discussion of final stylistic choices for site. Comments and suggestions for optimising site layout. Discussion of progress made in report. Discussion of each element of the report, what is required, and how to fill them out.
Actions for the next month
Finish off all questions in site for each language. Insert problem analysis and literature review into site. Begin working more on report so it can be finished fully before deadline.
Deliverables for next time
None.
Other comments
None.

Supervisor signature: Denis Reilly

Student signature: Nathan Stanley

6001PROJ Final Year Project

Monthly Supervision Meeting Record

Student: Nathan Stanley
..... Date: November

Main issues / Points of discussion / Progress made First part of literature review. Review of existing language learning websites and apps. Discussion of questionnaire that was created and distributed, awaiting replies.
Actions for the next month Begin situational / problem analysis.
Deliverables for next time Situational analysis first draft. Problem analysis first draft.
Other comments None.

Supervisor signature: Denis Reilly

Student signature: Nathan Stanley