Design and Development of a Web Extension to Help Facilitate the Learning of a Foreign Language

Connor Corbin, Deniz Cetinkaya[0000−0002−1047−0685], and Huseyin Dogan[0000−0002−9138−9319]

Bournemouth University, Department of Computing and Informatics
BH12 5BB, Poole, United Kingdom
http://hci.bournemouth.ac.uk/
{s4908654, dcetinkaya, hdogan}@bournemouth.ac.uk

Abstract. Learning a foreign language is a time consuming task that requires perseverance, commitment and hard work. The time required to learn a foreign language can range from hundreds to thousands of hours, depending on the language being learnt and the native language of the learner. Two major barriers to learning a language are lack of motivation and time. Spending increasing amounts of time online also has a detrimental effect on learning a language. The global digital language learning industry is growing and the market has more than doubled in size in the last decade due to the world becoming increasingly interconnected. There is a need to provide users with a web browser extension that is both interactive and continuous for learning a language. This paper presents the design and development of a web browser extension that will help facilitate the learning of foreign languages. The web extension overrides the default content when either a new tab or window is opened with interactive language learning material. The web extension is compatible with the desktop version of the Chrome web browser. Each time a new tab or window is opened, a range of language learning material are presented at random, which can be read, understood and completed in 30 seconds or less with immediate feedback. Overriding the web browser's functionality ensures the user is exposed to language learning material.

Keywords: Web Browser Extension · Online Language Learning · Accessible Design · Interactive Design.

1 Introduction

Learning a foreign language is a time consuming task that requires perseverance, commitment and hard work. The time required to learn a foreign language can range from hundreds to thousands of hours, depending on the language being learnt and the native language of the learner [1]. Two major barriers to learning a language are lack of motivation and time, with 34% and 28% of Europeans saying this discourages them, respectively [2].

The barriers identified above are exacerbated by internet users spending increasing amounts of time online; spending on average 6 hours and 43 minutes each day online [3]. However, the global digital language learning industry is growing [4]. Additionally, the global language services market has more than doubled in size in the last decade due to the world becoming increasingly interconnected, reaching 46.9 billion U.S. dollars in 2019 [5].

Many language learning applications exist, the most popular being Duolingo. Duolingo is the most downloaded educational application in the history of the Apple App Store, with approximately 300 million downloads and 30 million active users [6]. Language learning applications alone will not make an individual fluent in a language. However, they do allow an individual to learn new vocabulary, which is key to communicating in a foreign language. Communication is limited without grammar but without vocabulary nothing can be conveyed [7].

This paper presents the design and development of a web browser extension that will help facilitate the learning of foreign languages. The web extension overrides the default content when either a new tab or window is opened with interactive language learning material. The web extension is compatible with the desktop version of the Chrome web browser due to the mobile version not being able to support web extensions currently. Each time a new tab or window is opened, a range of language learning material are presented at random, which can be read, understood and completed in 30 seconds or less with immediate feedback. Our work is intended to supplement existing language learning solutions rather than replacing them. Overriding the web browser's functionality ensures the user is exposed to language learning material.

To the best of our knowledge, currently no web extension exists that provides interactive language learning material for users on Chrome Web Store. Interactive language learning material prompts users to engage with the content rather than passively absorb it. With the increasing amount of time individuals are spending online, as well as the increased digital language learning market, there exists an opportunity to provide users with a web extension that is both interactive and continuous. Thus, overriding the web browser's functionality ensures the user is exposed to interactive language learning material with immediate feedback. Moreover, many acknowledge the benefits of learning an additional language [2]. Therefore, a language learning web extension is appealing to many.

The web extension is built in a modular and extensible way so that with future development, further languages and different question types can be easily added. As a proof of concept, Spanish language materials are included, and user tests are held to evaluate the approach.

The remaining of the paper is structured as follows: Next section provides a brief background information and related work. Section 3 presents the analysis and design of the proposed approach. Section 4 explains the development details and relevant technologies. Section 5 presents the test results and outcome of the evaluation. Finally, last section concludes the paper and discusses the future work.

2 Background

This section presents the current state, outreach and market size of web extensions as well as the related work. Existing language learning web extensions and applications are studied to gain an understanding of their features, benefits and drawbacks. This provides information and guidance for eliciting the requirements and designing the web extension.

2.1 Chrome Web Extensions

Web extensions are small software programs that customise the browsing experience and are built with common web technologies such as HTML (Hypertext Markup Language), CSS (Cascading Style Sheets) and JavaScript [8]. Web extensions empower users to customise browsers' behaviour and functionality to their individual preferences or needs.

Chrome is based on a free and open-source software project by Google called Chromium. The Chromium project aims to provide users with a safer, faster, more stable and more secure way to experience the web [9]. Chrome is not the only web browser based on Chromium; Microsoft Edge and Opera are also Chromium based [10, 11]. However, Chrome is the most popular web browser across all device types with approximately 64% market share and 67% market share of desktop web browsers. Together, Chrome, Microsoft Edge and Opera have approximately 75% of the desktop web browser market share [12–14].

Web browsers based on Chromium support the same web extension application program interfaces (APIs) and manifest keys [15, 16]. Therefore, users can install web extensions directly from the Chrome Web Store with any browser that is based on Chromium. Given the market share of desktop web browsers based on Chromium and the potential outreach to users, the Chrome Web Store is the most lucrative market for publishing web extensions at present.

Chrome web extensions must serve and fulfil a single purpose that is easy to understand. A web extension that offers multiple and unrelated functionality would not serve a single purpose and therefore violate the Chrome extensions' policy. Violation of the Chrome extensions' policy will cause the web extension to be removed from the store [17]. There are approximately 188,620 web extensions on the Chrome Web Store. Of these, approximately 50% have less than 16 installs, 19,400 have 0 installs and 87% have less than 1,000 installs [18]. The statistics presented highlight the difficulty in producing a web extension that users like. Hence, it is important to correctly elicit the requirements to ensure the correct web extension is developed with features and functionality that the users will enjoy.

2.2 Language Learning Web Extensions

There are many language learning web extensions on the Chrome Web Store ranging in popularity and the number of active users. Three most relevant web extensions and techniques they have deployed to encourage users are discussed in this section.

Chrome web extensions are accessible on browsers which are based on Chromium, which controls the majority market share of desktop web browsers. This allows for the greatest exposure to potential users. Many existing language learning web extensions have hundreds of thousands of users. This was surprising due to background research presented in Section 2.1 showing that 87% of web extensions on the Chrome Web Store have less than 1,000 installs [18]. This highlights the demand and popularity for language learning web extensions.

Langly. This web extension supplements traditional language learning by encouraging users to learn a new word every time a new tab or window is opened by overriding the default content [19]. The opened new tab or window has a randomly chosen scenic background with a random word in a chosen language with the English translation. The web extension offers users the ability to pick one of three languages: German, French or Spanish. Furthermore, users can listen to how the word is pronounced by a native speaker. Langly aims to expand users' vocabulary in their chosen language which is important as limited knowledge of vocabulary impedes successful communication due to lexical errors [20, 21]. Although, the web extension is simple and aesthetically pleasing, the random scenic background has no relevance to the word displayed that the user is attempting to learn. It could be an opportunity to help users learn and retain newly learnt vocabulary by having relevant pictures relating to the displayed word on the screen. Pictures help maintain users' attention and facilitate learning foreign language vocabulary by making language learning interesting and entertaining [22, 23].

Mate Translate. This is an all-in-one translator with over a hundred languages to choose from. The web extension offers the ability to translate highlighted words, whole pages and Netflix subtitles [24]. The translated text displays the phonetic transcriptions with the ability to listen to the translated text. Translation of words is easy and straightforward; translation involves double clicking on a word, which opens an unobtrusive window without delay on the same page containing the translated text. However, while using the web extension, there was difficulty in selecting the desired language using the search selection dropdown menu. The dropdown menu did not recognise Spanish as a valid language option when using the search function. Consequently, this resulted in manually scrolling through all 103 languages to find Spanish, which was an inconvenience. One positive feature was the ability to switch between positive contrast polarity: light mode and negative contrast polarity: dark mode. This is advantageous for several reasons related to eye health [25] as well as personalised user experience [26].

Rememberry. It provides browser-based, interactive flashcards[27]. Using flashcards with spaced repetition is an effective learning strategy as opposed to massed study [28]. Spaced study allows for intervals between learning sessions, which allows the brain to form connections between ideas and concepts which helps the long-term retention of information. Massed study involves cramming one's learning into one mass study period and then moving onto another topic [29]. One negative experience with Rememberry was that there were no incentives. Nothing encourages or compels the user to make use of the flashcards feature, which would require them to click on the web extension logo in the toolbar. During periods of demotivation or taking breaks from learning, the web extension can be completely ignored and left unused. Additionally, the colour scheme of white, black and orange was not user friendly.

2.3 Language Learning Applications

The techniques utilised in language learning web extensions differ greatly to those in applications. Therefore, to gain a full and deeper understanding of the features and techniques used in a spectrum of digital language learning applications, existing language learning applications, not just web extensions, are also reviewed. Two most commonly used applications have been reviewed by outlining the advantages and disadvantages of the features and techniques deployed.

Rosetta Stone. The application provides a range of interactive language learning material and the ability to listen and learn offline. Rosetta Stone fully immerses the user from the beginning in one of the 24 language options they can chose from. Speech recognition software gives instant feedback, which helps fine-tune the accent and pronunciation of words [30]. Interactive learning is fun and engaging; it encourages users to think, explore and respond to new information, which aids their learning process and ability to retain information [31]. Lessons include multiple-choice questions, typing exercises, filling in the missing blanks relating to a picture and the pronunciation of words and phrases, which is then graded with speech recognition software. The learning material is both relevant and interesting. Furthermore, providing singular words in addition to phrases of various lengths is a practical approach to learning the target language.

An independent study measuring the effectiveness of Rosetta Stone shows a statistically significant improvement in language ability between the beginning and end of the study. Additionally, participants rated Rosetta Stone favourably with approximately 85% rating the software easy to use, enjoyable, helpful, would recommend to others and satisfied with their overall experience. Moreover, after 55 hours of studying Spanish, participants increased their oral proficiency by between 56% and 72% [32]. One negative experience of Rosetta Stone was the introduction of new words which were not immediately obvious in their meaning. Google translate was used to understand the word or phrase as there was no way to view English translations. Additionally, the full version of Rosetta Stone requires a costly subscription. Higher subscription costs can be off putting for users. Research suggests that the higher the subscription cost, the higher the attrition rate of the users [33].

Duolingo. This is a free language learning platform offering small bite-size interactive lessons that can typically be completed in 5 minutes or less. In the literature. research has been conducted to measure the effectiveness of Duolingo [34]. Participants completed one Spanish language test at the beginning of the study and one at the end. The results showed an overall improvement in participants' language abilities, which was statistically significant. Additionally, approximately 95% of participants rated Duolingo as easy to use, 92% believed that Duolingo helped them learn Spanish, 88% enjoyed learning Spanish with Duolingo and 79% were satisfied with Duolingo.

Each lesson provides learning material that can be read and studied before the interactive lessons. This contrasts with Rosetta Stone, which provides no learning material before lessons. Providing material prior to the lessons is an important and effective method for when new words and phrases are introduced. Interactive games include typing the Spanish translations of English expressions and vice versa, listening exercises, multiple-choice questions and speaking Spanish phrases, which are graded on accuracy.

Duolingo effectively uses gamification by offering experience points, a streak counter, leader boards and Lingots. Lingots are Duolingo's virtual currency that can be used to purchase additional features. The greater the progression through the levels, the more experience points and Lingots are gained. Gamification involves applying gaming mechanics to non-gaming environments to make difficult tasks more enjoyable. One drawback of Duolingo is that it groups words in semantic clusters, grouping animals, family and travel separately. This makes the acquisition of words grouped together more difficult due to the phenomenon called interference [35]. Interference theory dictates that forgetfulness is caused by memories interfering with and disrupting other memories [36].

3 Analysis and Design

Reviewing existing language learning solutions have highlighted the different approaches each have taken to facilitate language learning. Additionally, techniques and features that are advantageous and disadvantageous were identified. This helped to identify the requirements and features for the proposed web extension which users will enjoy and find helpful.

3.1 Requirements Elicitation

Requirements elicitation involves numerous activities and techniques that seek, uncover, acquire and elaborate requirements for a software system [37]. Requirement elicitation occurs during the initial phase of the project and is refined through an iterative approach throughout development. Iteratively refining requirements ensures the correct software system is developed [38]. This reduced the risk of the project failing due to inadequate requirements. Therefore, effective elicitation of requirements is important and vital in determining the success of the project [39, 40]. The techniques chosen for this research are questionnaires and introspection. Due to personal experience and our own interest in learning a foreign language, the introspection technique is also used to develop and identify requirements that would be required of a language learning web extension. Introspection is insufficient alone so in combination with other elicitation techniques.

Questionnaires allow for an efficient and quick method of collecting information from multiple stakeholders. The questions are centred around gaining an understanding of participants' current language learning activities and habits. Additionally, questions are asked about their current use of language learning applications and what they like and dislike. This feedback is essential for determining useful features and requirements for the web extension. Table 1 presents the questions that we used at this stage.

Table 1. Questions for the initial questionnaire.

No	Question
Q1	Are you interested in learning a foreign language?
Q2	Have you learnt any foreign language previously?
Q3	What applications/software do you currently use to learn a foreign language?
Q4	What do you like about the application/software you currently use?
Q5	What do you not like about the application/software you currently use?
Q6	How long do you currently spend each day learning a language on average?
Q7	What requirements would you want from a language learning application?
Q8	Any other comments?

All participants had an interest in learning languages and had used language learning software before such as Duolingo and Babbel. Some positive features are listed as simple design, easy to use interface, setting targets, quality of the content, and improving various skills. On the other hand, negative features are listed as lack of reminders or notifications for continuous usage, and not being able to access all content. The desired features are listed as simplicity, accuracy, listening correct pronunciation, easy to use, having nice colours, different question types, reminders for regular usage, and activities that can be completed quickly. Text translation and listening the question features are also stated to be beneficial. We had limited number of five participants so the results are not representative, yet they are used to gather initial set of requirements for the proof of concept implementation.

With the information gained through the critical analysis of existing language learning applications and web extensions, responses from the initial questionnaires and the effective use of the introspection technique, several requirements have been derived. Requirements are prioritised with the MoSCoW method to accomplish a minimum viable product and broken down further into specific tasks that can be added to a Kanban board. As an example, major requirements are listed below:

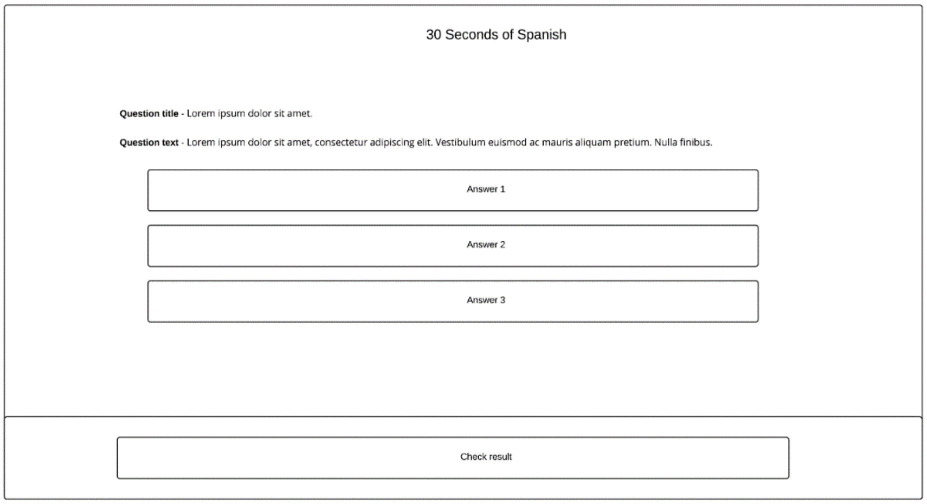
- The web extension must have a simplistic design and clear layout.

- The web extension must be easy-to-use.
- The web extension must display random interactive learning material each time a new window or tab is opened.
- The web extension must provide a single learning activity each time.
- The web extension must provide options to select the correct answer for multiple choice questions.
- The web extension must make use of images to aid language learning.
- The web extension must display clearly the result of the activity.
- The web extension must have high quality and accurate content.
- The web extension must make effective use of colours to be visually appealing

A minimum viable product is a version of a new product with enough features that allows for the maximum amount of feedback from users with the least amount of effort [41]. The main benefit of producing a minimum viable product is the feedback received from users interested in the product, without developing the full product. Therefore, feedback can be received from actual users quicker than if a full product is developed, which can be used to add, remove or adjust existing features depending on how the product resonates with the target market. This will lower the risk in producing a product that will not succeed whilst also reducing the effort, time and cost in producing the product.

3.2 User Interface Design

Wireframes were the initial design methodology used to create low-fidelity, two-dimensional designs of the web extension's user interfaces (UI). See Fig. 1 for an example.



30 Seconds of Spanish

Question title - Lorem ipsum dolor sit amet.

Question text - Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum euismod ac mauris aliquam pretium. Nulla finibus.

Fig. 1. Sample wireframe for the main page with multiple-choice questions.

Visual complexity and prototypicality are important in the users' aesthetic judgement process; users prefer minimal visual complexity and high prototypicality [42]. Users can experience a suboptimal initial impression when the UI contradicts their expectations, which can lead to lower product interaction [43]. Therefore, the web extension has a consistent design for all question types.

One advantage of wireframing UIs is to clarify and identify consistent designs for displaying categories of information to the user. This is crucial for the project due to its interactive nature. Users expect a question that is clear and easy to identify, in addition to feedback informing them if they have correctly or incorrectly answered the question. This information should remain consistent for all types of questions to increase usability, eliminate confusion and to evoke a positive emotional response. Additionally, wireframing helps allocate how much white space is given to each element. Increasing white space improves comprehension, clarifies the relationships between elements, attracts users' attention and leads users' eyes towards relevant content [44].

Designing the UI with wireframes enabled white space to be effectively allocated to elements. The produced wireframes determined the necessary containers required to achieve the desired visual result. The wireframe designs also identified the reusable React components that have been highlighted in different colours in Fig. 2 below.

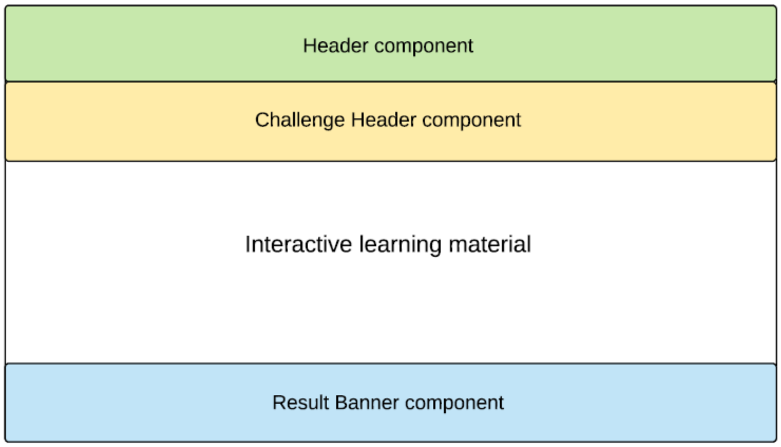


Fig. 2. Wireframe showcasing reusable React components.

3.3 Design of the Logo

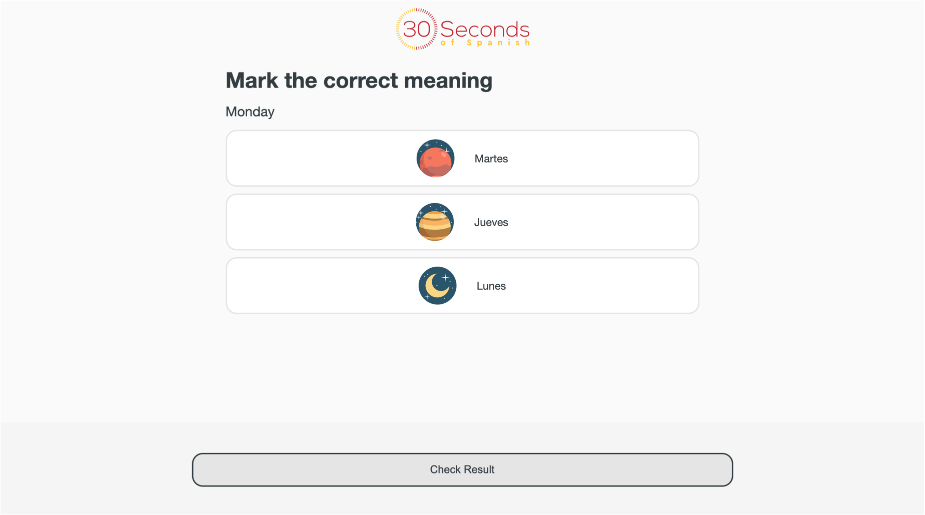
To make a good first impression to users, an aesthetically pleasing logo is required. A research study employed eye-tracking software and an infrared camera, tracked the participants' eye movements as they scanned different Web pages

[45]. The study found that the participants spent approximately six seconds looking at the organisation's logo before moving onto other sections. Additionally, users ascertain an opinion about the visual appeal of a web page within 50 milliseconds [46].

A descriptive logo was chosen due to having text and visual elements which communicate the service the web extension offers. The logo was based on the name of the web extension, 30 Seconds of Spanish. The logo has a watch dial design, which encloses the number 30 with lines spaced evenly in a ring which represents the minutes and seconds of a watch. The colour scheme reflects the colours of the Spanish flag. The creation of a visually appealing logo contributes to ensuring a warm welcome to the web extension. This is important as the positive emotional response attracts and encourages the user to return.

4 Development

The web extension is built in a modular and extensible way so that with future development, further languages can be easily added. As an exemplar study, Spanish language materials are included, and user tests are held to evaluate the approach. Fig. 3 shows a sample screenshot of the web extension with a sample multiple choice question. The web extension is built with JavaScript, React and Styled-Components. Patterns in the source code are statistically analysed, identified and reported to increase source code readability and maintainability. This section explains the implementation details.



Mark the correct meaning

Monday

Fig. 3. Sample screenshot of the implemented web extension.

4.1 Methodology

Agile principles are applied to perform each activity in the software development lifecycle, producing a web extension that helps users facilitate learning Spanish. Agile dictates that working software is the primary measure of progress. Two Agile principles can aid this: delivering working software frequently and maximising the amount of unnecessary work done [41]. In addition, Lean Software Development methodology principles are applied which is based on Lean manufacturing and aims to maximise the value to the consumer whilst reducing waste. Kanban was chosen to implement the Lean thinking principles [47]. Kanban places emphasis on just-in-time delivery; having the right work done at the right time whilst not overburdening the development team. Therefore, features that add value to the project should be implemented first.

The focus of Kanban is to accurately visualise work that needs to be completed and when such work needs to be completed by. This is done by prioritising tasks that are the most important, reducing the risk of their incompletion and allowing increased flexibility for other tasks. Kanban contains three main principles. The first principle, visualising the workflow, involves setting up a Kanban board containing columns, cards and tasks. Each column represents a specific step in the workflow and each card represents a task. The second principle, limiting work in progress, involves setting work-in-progress limits for how many tasks can be in each workflow state at any given time. Lastly, measuring the lead time, involves optimising the process to lower the average time one task takes to complete, whilst increasing the predictability of how long a task will take to complete.

4.2 Technology Stack

GitHub. GitHub is an open-source web-based Git repository hosting service, providing source code management, distributed revision control functionality of Git, access control, wikis, bug tracking and acts as a backup of source code. Throughout the development of the web extension, a straightforward branching model was followed. A develop branch was created from the default branch in Git: master available from https://github.com/ConnorCorbin/30-Seconds-Of-Spanish. The master branch contains production-ready code while the default branch contains pre-production code, which serves as an integration branch for features. For each new feature of the web extension, a feature branch was created from the develop branch.

Husky library is used to create Git hooks inside the project's repository [48]. Git hooks are scripts that automatically execute each time an event occurs. Common Git hooks include pre-commit and pre-push, which commonly run unit tests and linting scripts before the commit and push Git commands.

JavaScript. JavaScript is a programming language with first-class functions. This means functions can be passed as arguments to other functions, assigned as a value to a variable and returned by another function. JavaScript is essential for webpage manipulation and interacting with the user, therefore creating a dynamic and interactive experience.

ESLint JavaScript linting utility is used to identify and report on patterns in JavaScript code via static code analysis [49]. Static code analysis examines the code during development to identify and report on problematic patterns or code that does not cohere to certain style guidelines. Static code analysis of JavaScript code is important due to JavaScript being prone to developer error, which in turn is due to JavaScript being a dynamically typed language. ESLint provides the ability to configure a range of linting rules; each rule is standalone and can be turned on or off depending on the coding style.

React. React is a declarative, efficient and flexible open-source JavaScript library used for building UIs [50]. Complex UIs are created from small isolated pieces of code called components. Conceptually, components are similar to JavaScript functions. Components accept inputs called props and return React elements which describe what appears on the screen.

PropTypes library is used for type checking React props, exporting a range of validators to ensure the data received is of the correct type. Type checking is required because JavaScript is a dynamically typed programming language, meaning data types do not have to be specified while coding. Therefore, functions declared in JavaScript can accept different types of data. For example, a function that expects a string can receive a number. This can lead to runtime type errors due to an operation that is nonsensically performed, or an unexpected data type passed into the function. PropTypes minimises this risk by type checking the props for a React component. When an invalid value is passed in as a prop, a warning is shown in the JavaScript console.

Styled-components. Styled-components allow CSS to be written in JavaScript files to style React components [51]. There are several benefits of using styled-components. One includes the removal of class name bugs. This is because class names are automatically generated which removes the possibility of duplicated, overlapping or misspelt class names. Additionally, styled-components automatically injects vendor prefixes which ensures the latest CSS features work correctly and as expected on all web browsers. Lastly, styled-components automatically handles code-splitting; ensuring no redundant code is injected by tracking which React components are rendered and automatically injecting their styles only.

Styled-components exports a wrapper component called ThemeProvider that accepts a prop called theme. ThemeProvider enables all child React components to access the theme whether they are one or multiple levels deep. The value of the theme prop can be a variable or a function that returns an object. The returned object maps object properties to hexadecimal colour values, therefore, circumventing the necessity of repeating hard coded colour values in the styled components. Assigning property names that describe the function rather than the description of the colour allows easy interchangeability of a dark or light theme.

4.3 Manifest File

Chrome web extensions require a JavaScript Object Notation (JSON) formatted file which contains important information dictating how the web extension extends and modifies the web browsers capabilities [52]. The web extensions manifest file contains several properties. The name, version and manifest_version properties are all required whilst all others are optional. The chrome_url_overrides and new-Tab properties are essential; instructing Chrome to substitute an HTML file from the web extension for a page normally displayed when opening a new tab or window. The browser_action property is optional but advantageous; providing the default icon that appears in the main toolbar to the right of the address bar.

4.4 Question Components

The Multiple Choice Question (MCQ) component is a class component which utilises JavaScript classes; providing the same functionality of a functional component. Originally, the MCQ component was written as a functional component with React hooks. React hooks allow a functional component to use local state and other React features without writing a class component. However, the MCQ component was refactored into a class component due to external packages related to testing not fully supporting React hooks, which made testing more difficult and tedious.

The MCQ component uses the Label component. The Label component contains a possible answer to the multiple-choice question that can include an icon. Both text and image variants of the Label component are used. Additionally, the user can cycle through each Label component with the tab key and select their answer by pressing the enter key.

The Translate Text Question (TTQ) component is a class component due to requiring local state. However, unlike the MCQ component, the state object is defined outside the constructor due to the initial state values not being based on the props passed into the component. The TTQ component offers English and Spanish text translation questions that require the user to translate in the opposite language to the question. Additionally, the question title and text area prompt the user as to which language to type in. The user can submit their answer by pressing the enter key once they have entered at least one character into the text area. The TTQ component compares the users input with the correct answer, which is then passed into the component as a prop. However, strict string comparison is not suitable due to the possibility of users ignoring punctuation and misspelling words slightly. Therefore, a function based on the Sørensen-Dice coefficient was created, which measures the number of consecutive bigrams and returns a decimal number between zero and one [53].

4.5 Documentation

All components have an accompanying markdown documentation (MD) file containing various pieces of information directly relating to the component. The information includes the components features and consumed components. Additionally, a data model table is provided containing the components prop name, data type and default value, the component in which the prop is associated with and the description of the prop and function it serves. Furthermore, the repository contains a README MD file in the root directory. This file contains instructions on how to install the web extension locally and steps on how to create and release a new component, including the necessary files, which must be created and the naming conventions to follow.

5 Results

The section covers the results of testing and evaluation of the web extension.

5.1 Unit Testing

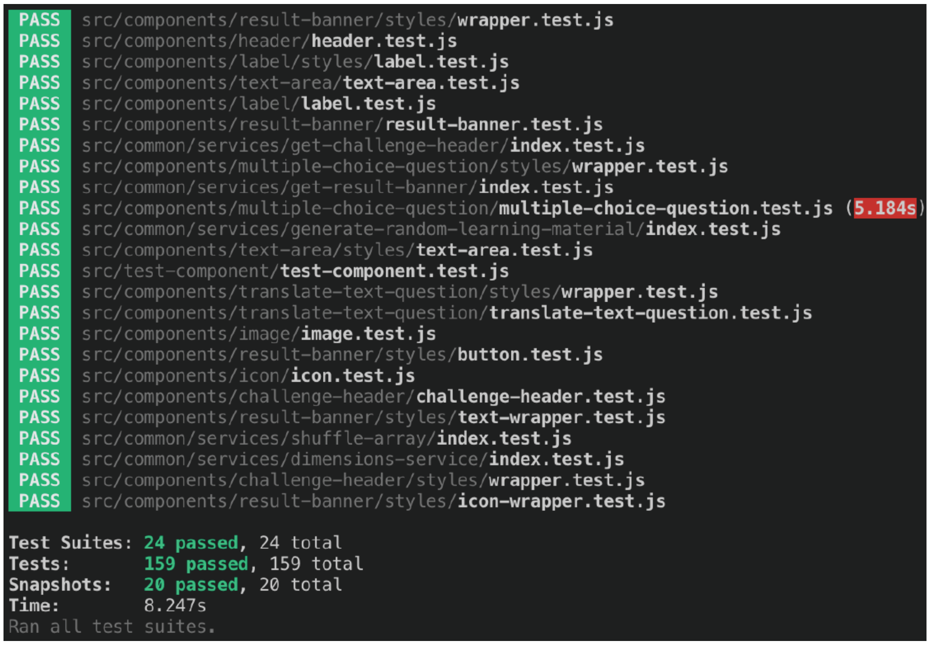
Enzyme and Jest were used for testing all modules inside the source folder, taking approximately 8.5 seconds to fully run the 159 unit tests created. Automated unit tests provide instant feedback, which is advantageous when refactoring; tests can be ran quickly to test that the existing functionality is as expected.

Enzyme is a JavaScript testing utility that simplifies testing React components by creating a simpler interface for manipulating, traversing and simulating components [54]. This is due to its intuitive and flexible API which mimics jQuery's API for manipulating and traversing elements in the document object model (DOM).

Jest is a JavaScript testing framework that is well documented and aims to work out of the box with little to no configuration, providing a fast and efficient task runner, mocking capabilities and assertion library [55]. Jest ensures tests have a unique global state that allows them to be ran in parallel. This allows a larger number of tests to be ran very quickly. Jest can also generate code coverage, with no additional setup. This collects coverage information on branches, functions, lines and statements. Code coverage thresholds can also be set for the collected information. Therefore, enforcing the specified minimum threshold for coverage results. Jest will return a failure if these thresholds are not met.

Jest was configured to provide a minimum threshold enforcement of 100% code coverage for branches, functions and lines. Additionally, 100% branch coverage implies 100% statement coverage [56]. Unit test creation would not stop once 100% code coverage was achieved. Previous experience, knowledge and intuition was used to derive unit tests.

Snapshot testing was used to test the DOM output for each React component. Snapshot testing is advantageous as it ensures the UI does not change unexpectedly. Snapshot testing captures the DOM output of the UI and compares it to a reference snapshot file stored alongside the tests. The produced snapshot is either expected or unexpected in which case the referenced snapshot must be updated, or the snapshot fails in which case an error has occurred. Fig. 4 below highlights the passing test suits and the number of successful tests.



```
 PASS  src/components/result-banner/styles/wrapper.test.js
 PASS  src/components/header/header.test.js
 PASS  src/components/label/styles/label.test.js
 PASS  src/components/text-area/text-area.test.js
 PASS  src/components/label/label.test.js
 PASS  src/components/result-banner/result-banner.test.js
 PASS  src/common/services/get-challenge-header/index.test.js
 PASS  src/components/multiple-choice-question/styles/wrapper.test.js
 PASS  src/common/services/get-result-banner/index.test.js
 PASS  src/components/multiple-choice-question/multiple-choice-question.test.js (5.184s)
 PASS  src/common/services/generate-random-learning-material/index.test.js
 PASS  src/components/text-area/styles/text-area.test.js
 PASS  src/test-component/test-component.test.js
 PASS  src/components/translate-text-question/styles/wrapper.test.js
 PASS  src/components/translate-text-question/translate-text-question.test.js
 PASS  src/components/image/image.test.js
 PASS  src/components/result-banner/styles/button.test.js
 PASS  src/components/icon/icon.test.js
 PASS  src/components/challenge-header/challenge-header.test.js
 PASS  src/components/result-banner/styles/text-wrapper.test.js
 PASS  src/common/services/shuffle-array/index.test.js
 PASS  src/common/services/dimensions-service/index.test.js
 PASS  src/components/challenge-header/styles/wrapper.test.js
 PASS  src/components/result-banner/styles/icon-wrapper.test.js

Test Suites: 24 passed, 24 total
Tests:       159 passed, 159 total
Snapshots:   20 passed, 20 total
Time:        8.247s
Ran all test suites.
```

Fig. 4. Unit test results.

5.2 Visual Screenshot Testing

BackstopJS was used to perform visual screenshot testing for the created React components, including variants of the same component [57]. BackstopJS automates visual regression testing of React components by comparing DOM screenshots over time. Tests are rendered with Headless Chrome, which allows the Chrome browser to be ran in a headless environment without the need for a full browser visible UI shell . This removes the memory overhead of running a full version of Chrome whilst providing the same environment used by users. Additionally, BackstopJS provides integrated Docker rendering to eliminate cross-platform rendering issues.

A total of 17 scenarios were created and tested at four breakpoints, therefore a total of 68 visual screenshot tests are ran each time a new feature is developed. For each breakpoint, BackstopJS will create and open a report in the web browser displaying the results of the scenarios. The scenario either passes or fails and therefore must be updated if the change is expected or an error has occurred. Automated visual screenshot testing is advantageous due to the rapid feedback received when performing regression testing.

The automated testing techniques implemented allow an objective assessment regarding the level of conformance of the web extension to the web extensions requirements. The techniques help finding the defects and errors made during the development and ensure the web extension does not experience any failures, leading to increased quality and confidence in the source code and web extension. Increased quality leads to fewer failures and delays in attaining product stability.

5.3 User Tests

Once a minimum viable product was produced, user tests were held to collect constructive feedback. All participants have interest in learning languages and have basic level of Spanish. Instructions were sent to the participants and help offered during the installation of the extension if needed. Users liked the simple and effective web extension layout, which highlights the importance of utilising the wireframe design methodology to position and apply white space to elements. Users enjoyed the two types of questions implemented. The feedback revealed that the text translation question is more challenging than the multiple-choice question, furthermore, a range of difficulty in the question types was found to be helpful. The web extensions logo was found to be relevant to the service the web extension provided and attracted users' attention.

The colour scheme and accessibility offered was well received, which aids the requirement of the web extension having high accessibility. No errors or failures were reported from the user feedback. While users enjoyed the multiple-choice questions, the pictures made answering the question too easy. Furthermore, users suggested additional features that would be beneficial in their learning, such as an option for further questions and an option to retake a question.

The proposed solution currently offers only Spanish language materials. However, the web extension has been produced in a modular and flexible fashion that will enable the addition of further languages and features to be offered with additional development. User evaluation study is limited to five participants and to the selected example. We will add more content and involve more participants to improve our evaluation in future validation studies.

6 Conclusion and Future Work

Chrome web extensions are accessible on browsers which are based on Chromium and there is a demand for language learning web extensions. The existing related work discussed in Section 2 miss several opportunities to enhance and help facilitate users' language learning. One such opportunity was that no incentives exist for users to use the web extension; only Langly addressed this issue with overriding the default content when opening a new tab or window. However, the content Langly provides is not interactive, offering only a single word translation with a background image that is irrelevant to the word.

This paper presents the design and development of a web extension to help facilitate the learning of a foreign language. The proposed web extension overrides the default content of a new tab or window with interactive language learning material. It provides random interactive language learning material each time a user opens a new tab or window in the desktop version of the Chrome web browser.

Agile principles are applied with the Lean Software Development methodology to perform each activity in the software development lifecycle, producing a web extension that helps users facilitate learning Spanish. Requirements for the web extension were derived through researching existing language learning web extensions and applications, questionnaires and introspection. The web extension is built in a modularised fashion with JavaScript, React and Styled-Components. Unit tests are written with Jest and Enzyme as well as visual screenshot testing is performed.

The web extension provides a solid foundation for future enhancements. As a future work, we will focus on expanding the number of languages that can be offered by the web extension as well as adding new learning materials. Additionally, implementing new question types and introducing difficulty levels are planned. Once these updates are implemented, we would like to improve the validation step with more participants and structured feedback.

Acknowledgments

We would like to thank the participants and interviewees for their valuable input and comments.

References

1. United States Department of State: Foreign Language Training. Available from https://www.state.gov/foreign-language-training/, last accessed 29 Jan 2020
2. European Commission: Europeans and their Languages. Report. Available from http://ec.europa.eu/commfrontoffice/publicopinion/archives/ebs/ebs_386_en.pdf (2012)
3. DataReportal Digital 2020: Global Digital Overview. Available from https://datareportal.com/reports/digital-2020-global-digital-overview, last accessed 17 Mar 2020
4. Statista: Digital language learning global market size by language 2025. Available from https://www.statista.com/statistics/948857/digital-language-learning-global-market-size-language/ (2019)
5. Statista: Global language services market size 2019. Available from https://www.statista.com/statistics/257656/size-of-the-global-language-services-market/ (2019)
6. Adams, S.: Game of Tongues - How Duolingo Built a $700 Million Business With Its Addictive Language-Learning App. Forbes article. Available from https://www.forbes.com/sites/susanadams/2019/07/16/game-of-tongues-how-duolingo-built-a-700-million-business-with-its-addictive-language-learning-app/ (2019)
7. Folse, K.S.: Vocabulary myths - Applying Second Language Research to Classroom Teaching. Ann Arbor: University of Michigan Press (2004)
8. Google Chrome: What are extensions? Available from https://developer.chrome.com/extensions, last accessed 6 Feb 2020
9. Chromium Projects. Available from https://www.chromium.org/, last accessed 19 Mar 2020

10. Opera version history - Opera Help. Available from https://help.opera.com/en/opera-version-history/, last accessed 18 Mar 2020
11. Microsoft - Download the new Microsoft Edge based on Chromium. Available from https://support.microsoft.com/engb/help/4501095/download-the-new-microsoft-edge-based-on-chromium, last accessed 18 Mar 2020
12. StatCounter Global Stats: Browser Market Share Worldwide. Available from https://gs.statcounter.com/browser-market-share, last accessed 18 Mar 2020
13. StatCounter Global Stats: Desktop Browser Market Share Worldwide. Available from https://gs.statcounter.com/browser-marketshare/desktop/worldwide, last accessed 19 Mar 2020
14. Statista: Desktop internet browser market share 2015-2020. Available from https://www.statista.com/statistics/544400/market-share-of-internet-browsers-desktop/ (2020)
15. Microsoft - Port Chrome Extension To Microsoft (Chromium) Edge. Available from https://docs.microsoft.com/enus/microsoft-edge/extensions-chromium/developer-guide/port-chrome-extension, last accessed 19 Mar 2020
16. Opera — Extension APIs Supported in Opera. Available from https://dev.opera.com/extensions/apis/, last accessed 19 Mar 2020
17. Google Chrome - Extensions Quality Guidelines FAQ. Available from https://developer.chrome.com/extensions/single_purpose, last accessed 19 Mar 2020
18. Extension Monitor: Breaking Down the Chrome Web Store. Available from https://extensionmonitor.com/blog/breaking-down-the-chrome-web-store-part-1, last accessed 7 Feb 2020
19. Chrome Web Store: Langly. Available from https://chrome.google.com/webstore/-detail/langly/lmhamlhlkpedkileakbifbdamgpnegma, last accessed 13 May 2020
20. Moghadam, S., Zainal, Z., Ghaderpour, M.: A Review on the Important Role of Vocabulary Knowledge in Reading Comprehension Performance. Procedia - Social and Behavioral Sciences, 66, pp. 555–563 (2012)
21. Alqahtani, M.: The importance of vocabulary in language learning and how to be taught. International Journal of Teaching and Education, 3 (3), pp. 21–34 (2015)
22. Carpenter, S., Olson, K.: Are pictures good for learning new vocabulary in a foreign language? Only if you think they are not. Journal of Experimental Psychology: Learning, Memory, and Cognition, 38 (1), pp. 92–101 (2012)
23. Lavalle, I., Briesmaster, M.: The Study of the Use of Picture Descriptions in Enhancing Communication Skills among the 8th Grade Students-Learners of English as a Foreign Language. i.e.: inquiry in education, 9 (1), article 4 (2017)
24. Chrome Web Store: Mate Translate – translator, dictionary. Available from https://chrome.google.com/webstore/detail/mate-translate-—translat/ihmgiclibbndffejedjimfjmfoabpcke, last accessed 13 May 2020
25. Aleman, A., Wang, M., Schaeffel, F.: Reading and Myopia: Contrast Polarity Matters. Scientific Reports, 8 (1), pp. 1-5 (2018)
26. Nielsen Norman Group: Dark Mode vs. Light Mode: Which Is Better? Available from https://www.nngroup.com/articles/dark-mode/, last accessed 8 Feb 2020
27. Chrome Web Store: Rememberry - Translate and Memorize.Available from https://chrome.google.com/webstore/detail/rememberry-translateand/dipiagiiohfljcicegpgffpbnjmgjcnf, last accessed 13 May 2020
28. Kornell, N.:Optimising learning using flashcards: Spacing is more effective than cramming. Applied Cognitive Psychology, 23 (9), pp. 1297-1317 (2009)
29. Kang, S.: Spaced Repetition Promotes Efficient and Effective Learning. Policy Insights from the Behavioral and Brain Sciences, 3 (1), pp. 12-19 (2016)

30. Rosetta Stone: How Rosetta Stone Works — Our Immersion Language Learning Methodology. Available from https://www.rosettastone.co.uk/how-it-works, last accessed 13 May 2020
31. Anderson, J.: The Benefit of Interactive Learning. Harvard Graduate School of Education. Available from https://www.gse.harvard.edu/news/14/11/benefit-interactive-learning (2014)
32. Vesselinov, R.: Measuring the Effectiveness of Rosetta Stone. Report. City University of New York. Available from http://resources.rosettastone.com/CDN/us/pdfs/Measuring_the_Effectiveness_RS-5.pdf (2009)
33. Danaher, P.: Optimal Pricing of New Subscription Services: Analysis of a Market Experiment. Marketing Science, 21 (2), pp. 119-138 (2002)
34. Vesselinov, R., Grego, J.: Duolingo Effectiveness Study. Report. Available from http://static.duolingo.com/s3/DuolingoReport_Final.pdf (2012)
35. Tavares, A.: Some Ideas to avoid L1 interference. World of Better Learning - Cambridge University Press. Available from https://www.cambridge.org/elt/blog/2017/10/06/some-ideas-to-avoid-l1-interference/ (2017)
36. Baddeley, A., Logie, R.: Working Memory: The Multiple-Component Model. In A. Miyake & P. Shah (Eds.), Models of Working Memory: Mechanisms of Active Maintenance and Executive Control, pp. 28-61. Cambridge University Press (1999) https://doi.org/:10.1017/CBO9781139174909.005
37. Yousuf, M., Asger, M.: Comparison of Various Requirements Elicitation Techniques. International Journal of Computer Applications, 116 (4), pp. 8–15 (2015)
38. Zowghi, D., Coulin, C.: Requirements Elicitation: A Survey of Techniques, Approaches, and Tools. In: Aurum A., Wohlin C. (eds) Engineering and Managing Software Requirements. Springer, Berlin, Heidelberg (2005)
39. Charette, R.: Why Software Fails. IEEE Spectrum. Available from https://spectrum.ieee.org/computing/software/why-software-fails (2005)
40. Stanley, R., Uden, L.: Why Projects Fail, from the Perspective of Service Science. Advances in Intelligent Systems and Computing, pp. 421–429. (2013)
41. Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., C. Martin, R., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D.: Manifesto for Agile Software Development. Available from https://agilemanifesto.org/ (2001)
42. Tuch, A., Presslaber, E., Stöcklin, M., Opwis, K., Bargas-Avila, J.: The role of visual complexity and prototypicality regarding first impression of websites: Working to-wards understanding aesthetic judgments. International Journal of Human-Computer Studies, 70 (11), pp. 794–811 (2012)
43. Raita, E., Oulasvirta, A.: Too good to be bad: Favorable product expectations boost subjective usability ratings. Interacting with Computers, 23 (4), pp. 363–371 (2011)
44. Bachvarova, V.: The power of empty space in UI design. Available from https://uxplanet.org/https-medium-com-viktorija-bachvarova-the-power-of-empty-space-in-uidesign-14f14f8b203 (2018)
45. Sheng, H., Lockwood, N., Dahal, S.: Eyes Don't Lie: Understanding Users' First Impressions on Websites Using Eye Tracking. Human Interface and the Management of Information. Information and Interaction Design, pp. 635–641 (2013)
46. Lindgaard, G., Fernandes, G., Dudek, C., Brown, J.: Attention web designers: You have 50 milliseconds to make a good first impression!. Behaviour & Information Technology, 25 (2), pp. 115–126 (2006)

47. Anderson, D.: Kanban - Successful Evolutionary Change for Your Technology Business. Sequim, WA: Blue Holes Press (2010)
48. Husky GitHub repository. Available from https://github.com/typicode/husky, last accessed 23 Mar 2020
49. ESLint: Pluggable JavaScript linter. Available from https://eslint.org/, last accessed 14 May 2020
50. React: Introduction to React Tutorial. Available from https://reactjs.org/tutorial/tutorial.html, last accessed 18 Feb 2020
51. Styled Components Basics. Available from https://styled-components.com/docs/basics, last accessed 23 Mar 2020
52. Google Chrome: Manifest File Format. Available from https://developer.chrome.com/extensions/manifest, last accessed 15 May 2020
53. Guindon, B., Zhang, Y.: Application of the Dice Coefficient to Accuracy Assessment of Object-Based Image Classification. Canadian Journal of Remote Sensing, 43 (1), pp. 48–61 (2016)
54. Enzyme - Introduction to Enzyme. Available from https://enzymejs.github.io/enzyme/, last accessed 23 Mar 2020
55. Jest Delightful JavaScript Testing. Available from https://jestjs.io/, last accessed 23 Mar 2020
56. Myers, G., Sandler, C., Badgett, T.: The art of software testing. 3rd ed. Hoboken, N.J.: John Wiley & Sons (2012)
57. BackstopJS GitHub repository. Available from https://github.com/garris/BackstopJS, last accessed 23 Mar 2020