

Final Report

My chosen project is a word-based game, one of the templates for Mobile Development. The project is based on creating a fun and addictive mobile app to learn words, in particular, a fun and engaging game where you can learn words in English/Spanish in order to have the necessary vocabulary to ace the CEFR tests. I chose The Common European Framework of Reference for Languages (CEFR) because this is an international standard for describing language ability, and because there is a defined list of words necessary for each level, which can be found on [this website](#).

Therefore, the app's aim is to teach the users the vocabulary needed to ace CEFR examinations. This game is aimed at people (adults and children alike), with one important constraint: they must already know how to read and write, at least in their own language. This app would not only help users who want to take language tests, but also those who want to expand their vocabulary in a particular language, even if it is their native language.

Since my objective is then to create an educational application which offers the users an entertaining and engaging way to learn new words, I performed a literature review on what are the best approaches to achieve this. These are the questions that I found necessary to elucidate with the review: What makes an app engaging and/or addictive? Are there any design approaches that should be followed when creating an application of this type? Are there studies done on the effectiveness of using mobile apps to increase the language learning potential of people?

While searching for research papers, books, journals, etc. in the Online library, which were related to the area I am interested in. From the results found, the most important were:

1. *Educational mobile game for learning English words* [\[11\]](#): In this paper, B K Ng, N M Suaib, A J Sihe, A Ali and Z A Shah (2020) set out to create a prototype mobile application named EWORD. This application objective was to teach primary school children English words according to the curriculum for English in Malaysia (SBECLC). They divided their word in four phases, the preliminary analysis, where they studied current options for learning and how a mobile application might fit into this educational environment. During this stage they found several papers that attested to their hypothesis that mobile games can be used as a tool for learning a new language thanks to their portability and convenience of use, along with the fact that using this approach can reduce the limitations of traditional teaching methods (limited time, limited materials, boring/passive learning environment, etc.). One of the results from this stage was a table which summarizes the requirements of Educational Mobile Games. The other phases were the design, development and evaluation of the proposed game. During the design

phase it was decided that the main character would be a magician called Kiki. The objective of the game was then to help her collect nine gems which would make her a full-fledged magician. In order to do this, the users should earn points while learning new words, by matching an image/text to a word. The game was developed using Unity3D and Android SDK. To evaluate the game they used two questionnaires (pre and post game activity) with the intention of measuring quality components of Usability Testing. The questionnaires were given to three target users (children 10-12 years old) and ten non-target users (adults 23-30 years old).

The evaluation of the app found that the game design and mechanic was easy and enjoyable to use and fulfilled the requirements for an educational mobile game that they had previously set. Nonetheless, they acknowledge that the testing was quite limited (due to restrictions set by COVID) and that further testing is needed to assess how this type of game could benefit children in the learning of a foreign language.

Personally, what I found quite useful of this work is the background research done and their summarizations, especially their definition of the Requirements of Educational Mobile Games and the way they compared different language apps to get a sense of what features successful language learning apps include and their shortcomings.

2. *A Review of Heuristics Evaluation Component for Mobile Educational Games* [\[12\]](#): In this paper, Nur Marissa Vee Senap and Roslina Ibrahim (2019) set out to identify the set of heuristics or principles needed to evaluate the usability of mobile educational games, since at the time there was not a clearly defined set in the literature. They analyzed 28 studies done on evaluation heuristics for three different technologies: games, educational games and mobile applications and created summarization tables for each of them along with a Venn Diagram that consolidates the suggested heuristics components for Mobile Educational Games, among which Usability and Playability principles were highlighted. They found (as was expected) that these technologies shared some common principles (e.g. Playability, Enjoyment, Usability, etc.) Based on their background research they were able to determine that:

- Mobile Educational games can improve student's performance.
- Although mobile phones are a popular way of playing games because of their ubiquitousness in our daily lives, they possess many distracting features that can hinder the learning process.
- To verify the success of an application, its usability must be tested, that means, that the users can use it to achieve a specific goal in an effective, efficient and satisfactory manner.
- Heuristic Evaluation is the most frequently used usability inspection method, because it is usually a less expensive alternative, and it can also be applied during any stage of the development process, unlike user testing.

Even when this paper does a thorough investigation on proper heuristics for mobile games, it would have been ideal that it also suggested implementations on how to properly test each of the principles found. The authors recognize that further research is needed to validate their findings but also point out the difficulties on establishing a determined set of principles for evaluating educational games since Heuristic Evaluation is a process usually applied to a software interface, but educational games present different aspects like pedagogical value, learning objectives, etc. which makes them have an unique context where common heuristics applied to mobile games can not be applied directly.

3. Mobile Game Design Essentials [\[1\]](#): In Chapter 11 of this book, the authors Scolastici and Nolte (2013) establish a set of best practices that should be followed when developing a mobile game, along with facts as to why it is important to take certain considerations when creating a mobile app, independently of its main objective or target audience. Here I will summarize some suggestions described in this chapter (the ones that directly apply to my project):

- Players must be rewarded according to the difficulty of the game.
- Mobile games are usually played casually, keep it short but entertaining.
- Casual gamers prefer "*polished games with well-tested mechanics, clear goals, and cute graphics*".
- You should not expect your users to invest much time in learning your game's mechanics.
- You should design games that are able to run smoothly on a variety of devices, not only high-end ones.
- Take into account different configurations (e.g. CPU speed, screen resolution and size, etc.) and operating systems where you want your game to be played.
- You should pick 2-3 mobile phones as a reference and find assets (e.g. pictures, icons, etc.) for each of them. Do not just resize them.
- Game characters should be appealing and meaningful.
- The game should not require complex controls that obstruct the playing area.
- Do not design games that highly depend on audio, since annoying sounds might push the user to leave the application or the environment where the user usually plays does not allow for sounds to be heard appropriately.
- Develop the game in a way that the memory requirements are minimum, this way you can target broader audiences.
- The game play should be short. About three minutes or less.
- Add pause and auto-save features. This should be a must.

- What makes a game fun can differ for different people. There is a theory that suggests four types of fun. One of them is “Hard fun”, where the main reason people play is overcoming obstacles, testing their skills and feeling accomplished.
- Give feedback on progress and success in the game.

Even when this book is ten years old, I found their suggestions to be applicable and meaningful in the current playing field. This chapter's dedication to give readers a well grounded list of do and don'ts for the development of mobile games is quite handy, since it summarizes in simple terms what we as developers need to know so that our projects do not fall into elementary pitfalls.

4. Addictive nature of Gamification [\[2\]](#): In their thesis, Bushra Qazi Abbasi and Samrah Awais set out to establish the relation between gamification and Smartphone Addiction along with highlighting game elements that contribute the most to that type of addiction. For this, they did an exhaustive research on current theories that try to explain the psychological reasons behind this phenomena. Based on their research, the authors were able to ascertain that gamification increases user engagement, which in turn can lead to addiction. Albeit, it can also be used for productive tasks like language learning. They also express that there exists Interaction Design Techniques that are being used indiscriminately to make users cling to an application.

They based their research on four psychological theories: Self-determination theory, Flow Theory, Hexad framework and Big Five model. These theories support the link between Gamification and addiction. One of these theories (Flow) was presented in 1990 by Csikszentmihalyi[\[10\]](#), who stated that flow is a condition where a person focuses their entire attention on the activity they are performing, forgetting about themselves and their environment, which can cause said activity to become impulsive. Later it was found that this same “flow” can be achieved while playing games. I highlight this theory in particular because it is mentioned in almost every literature that deals with gaming.

To reaffirm their arguments and their understanding on the background research done, the authors interviewed two behavioral psychologists. They also did two surveys which were distributed online and self-administered, aimed at a diversified group of people from around the world between 12-65 years old, who used their phone for at least 3 hours a day. The first survey was aimed at finding out the most common game elements, among a set of features that were found in the literature and in proven addictive applications. This survey concluded that the five most common features are coupons/rewards, emoticons/stickers, points, pop-up notifications and views/likes.

The second (and main) survey's objective was to take the results from the first survey and find out if those features were addictive in nature or not. For this they made statistical analysis on the data, from this they found that the most addictive features according to frequency distribution were scrolling, tapping/stories and pop-up notifications respectively. Chi-square analysis reported no link between these features and hourly daily usage. Nonetheless, "streaks" were found to be significantly related to usage.

The results of this project are highly applicable to the current gaming scenario, and even though their research was restricted to certain types of applications (i.e. social, communication and e-commerce), that they did not cover all the existent game elements, and that their survey population was limited to 269 persons (170 valid cases), the authors acknowledge their research shortcoming and it is fair to say that they accomplish their objective of laying the foundation for future endeavors to analyze the link between addiction and the design features used in mobile apps.

5. Gamification Model using Game Elements for E-Learning ^[3]: In the paper by F. S. Rahayu, L. E. Nugroho and R. Ferdiana, the authors undertook a thorough investigation on empirical studies in order to identify specific game elements that could influence the engagement/addictiveness of a game. This with the aim of proposing an e-learning model that could increase the engagement and motivation of users.

The main motivation for the study was the fact that traditional learning models are often viewed by students as tedious and repetitive, which is not helpful in the pursuit of knowledge, along with the fact that many prior studies had shown that gamification can have a positive influence in the learning experience. The proposed model was based on two theories, Self Determination and Gamified Learning Theory. The former states that three basic psychological needs must be covered: competence, relatedness and autonomy. These needs must be attained through the use of intrinsic (natural inclination to perform an activity) and/or extrinsic (e.g. rewards) motivations, being extrinsic motivators the most used in gaming experiences (though not the best to accomplish psychological well-being). The second theory explains how gamification can affect learning. The authors found that a game should have certain types of activities: empowering (increase status), farming (collecting resources) and raiding (overcoming obstacles). They also highlighted the influence that user types can have on gamification, different people can experience the game in a different way.

One feature of games that was discouraged for e-learning platforms is the use of *punishment*, since previous studies found that it can decrease student's performance. The game should include different types of rewards to fulfill the *competence* need and to *empower* the users. Rewards are also used as a *farming* mechanism, to induce users to create habits by repeating

certain patterns. *Relatedness* can be satisfied by social interaction (e.g. leaderboards, support groups, etc.). Autonomy can be satisfied by allowing users to choose their own learning path. The model should also include feedback to allow students to know how they are performing, which covers *competence* and *relatedness* objectives.

It must be noticed that the author's objective was not to prove that gamification can in fact increase the learning rate or that it has a direct impact on learning outcomes, but rather to summarize a set of rules to follow in order to create an engaging experience for users of e-learning platforms/apps, which they did. It might seem like the guidelines are quite broad, but since the study was not centered on a specific subject, the guidelines should encompass all possible applications for e-learning. Although the suggested model was not empirically tested, it was based on the research done on the subject over the years by many other authors and I must add that these guidelines are in line with my own findings in quite successful learning apps like Memrise, Duolingo, Elevate and Brilliant.

6. Engagement-Addiction dilemma ^[4]: In this research paper, Qianwen Yang and Xiang Gong (2021) explore the engagement-addiction dilemma in mobile games. The dilemma is about how mobile games exploit design features in order to increase user engagement and how users worry about becoming addicted to those games since they can be played at any time, anywhere. They based their research on the Stimulus-Organism-Response (SOR) framework and validated their findings by applying surveys to 410 users of the game "Arena of Valor" in China. The users differed in age, income, education, gender and experience.

The SOR framework corresponds to stimuli (external factors like design features, graphics, sounds, etc.), organism (how the user experiences the stimuli) and response (external user reaction to stimuli). Using this framework, many other researchers have proven the influence of design features in the user experience and behaviors. This research was focused on proving that certain design features are positively related with telepresence and social presence (which in turn can lead to addiction).

What I found interesting about this research was their summary of implications for the practice of mobile game interface design, since this was one of my objectives with this literature critique. Here I will list some of their conclusions:

- Use aesthetically attractive graphics, humanized characters, appealing colors.
- Controls must be of adequate size to allow easy data input.
- The objective of controls must be obvious.
- Use concise language and short labels.
- Follow a top to bottom approach, with a logical path.

- Must offer a way to get achievements or to compete with others.
- Allow social interaction with other users (in game chats, etc.)
- Use features of virtual worlds (like different characters, different gaming experiences)

The authors also provide a set of guidelines to providers, so that their games can act towards limiting the possibility of the users getting addicted. I do not believe this is something that the game developers actually want to address, but I think it is important that studies are being done on this subject and that there are actually easy ways for developers to avoid creating addictive content.

7. Gamification to learn code readability [\[5\]](#): In this paper, Qing M., Jacky K., Xiupei M., Yan X. and W.K. Chan (2018) set out to implement and test an online learning platform called GamiCRS which was aimed to teach CS students about code readability. The basic idea was to make the learning process more interesting by gamifying it, that is, adding intrinsic and extrinsic motivators. To evaluate the effect of these changes compared to a non-gamified counterpart, an experiment was conducted during two weeks with 161 (81 treatment group, 80 control group) undergraduate CS students in the City University of Hong Kong. The students did not know the main objective of the study being done and could use the platform at their leisure.

One interesting fact that was mentioned in this paper was about how previous research had shown that performance varies according to which type of motivators are used: intrinsic motivators offer a strong and long-lasting engagement and produce high quality behaviors in users, whereas extrinsic motivators are less effective but are exceptionally necessary when dealing with behaviors that are not naturally engaging/pleasurable to users, which is why extrinsic motivators are so ubiquitously used in games.

In their platform, the authors employed a classic mixture of game design elements: Points, Badges and Leaderboards (PBL) (extrinsic motivators). These features are used to fulfill the user's basic psychological needs. Users also have the possibility to upload code snippets and rate other people's code, this was done to activate elements of intrinsic motivations.

The authors also mentioned a crucial fact: while extrinsic motivators are necessary, they should not be overused because this can be detrimental for the learning process. For example, if a user receives too many rewards this can thwart their interest in the game, or if they do not score too well on the leaderboard they can feel frustrated and discouraged to keep playing.

After allowing students to use the learning platform for two weeks, the authors used a questionnaire based on the Technology Acceptance Model to test the User Acceptance. They tested Perceived Ease of Use, Perceived Usefulness, Attitude Towards Game Design Elements

and User Satisfaction. The platform mostly received positive feedbacks, some comments on it were:

- The platform was practical and useful.
- Points and leaderboards were seen as more effective than badges to motivate user participation.
- The game should use more attractive feature design.
- The game should also include non-deterministic rewarding mechanisms (i.e. challenges).

My interest in this paper stemmed from the fact that, as a CS student myself, I had used several gamified platforms offered by the University of London and was interested in knowing how to evaluate the usability and efficacy of a gamified learning environment, since my Final Project is based on this subject. The Technology Acceptance Model seemed easy to understand and put into practice, so I will probably make use of it to test my own application. A few maxims I learned from this research were:

- If the app is eye-pleasant but not useful then this is a bad design.
- If the app is useful and with clear objectives, but lacks in looks then this is a bad design.
- Rewards are important, but too much of something can be harmful.
- Aim to use intrinsic motivators rather than only trusting in extrinsic motivators.

PROJECT DESIGN

The central idea of the game is this: users will be given the definition of a word, then they must choose letters provided for them in the UI to form the word that matches the given definition (with the first 2-3 letters provided as a hint since many words might match the same definition), see [Figure 1](#) for a look at the UI design. The main challenge I need to overcome is creating a game experience for an educational game that is sufficiently entertaining for users, so that they don't feel like they are doing homework.

Since there can be more than one definition for a given word, and since most APIs for dictionaries are not open source, I decided to use *OpenAI GPT-3.5 Model* to create the most common definition for the list of words. It must be noted that I could only find the list of words for each CEFR level for the English language. I was not able to find an open source for the Spanish version of this test, not even on the Cervantes Institute [website](#). For this reason I had to use the same list of words for both courses, and I used *OpenAI GPT-3.5 Model* to translate each word-definition pair into Spanish.

My prototype included the use of pronunciations for each word. I found an open source for these files (English language only) at www.dictionaryapi.com, and was supposed to follow [their instructions](#) in order to download the required files. Nevertheless, my final app did not include this

functionality due to time constraints to implement it, even though I had already laid out the code for its UI appearance. The code is still included as a comment in the files.

I chose to name my app Koios after the mythological Greek God of the inquisitive mind. The theme of the game is ancient Greece, that is why the backgrounds used in the application are mostly Athens scenery (generated with the use of AI web applications by myself).

Since this is supposed to be a casual game, which might be played under conditions such as riding the bus, tram, subway, etc. Internet connection might be an issue. That is why all of my primary assets (vocabulary lists, images, sounds) are housed within my app. I will not be obtaining them from external APIs. I want to ensure that my app is always available, even if the external APIs are down. Another advantage of this is that I do not need to pay to query external APIs (even when some versions are free, there is a limit on the number of requests made by a single user). I will make use of expo-sqlite to create and populate a database when the application loads for the first time. The database schema is shown in [Figure 16](#).

My game will not offer the possibility to create a profile linked to email accounts or any other external sources. This is to ensure that there is no possibility of user information being leaked. All game data will be stored on the user's phone and will not be saved anywhere else. This is also to avoid dealing with the potential legal issues that may arise from handling children's data. See [Figure 2](#) for a detailed account of the assets used by the application.

The application will be developed for Android and iOS systems, using React Native. Some other technologies/libraries I will make use of are: JSX, JavaScript, Python and Pandas (to clean data when necessary), other React Native Libraries to implement certain functionalities of the game like: timers, graphs and animations,, Async Storage and expo-sqlite (to save data).

For each language, there are six levels defined by the CEFR (A1, A2, B1, B2, C1, C2). Each level presents its own list of words. The completion criteria I chose for the levels is that each word must be guessed correctly thrice, at different opportunities.

Here is a list of the most important features that I am planning to include in my application, along with the sources I found that support their use and efficacy in creating a fun, engaging and addictive educational game:

- Players are rewarded with points when they find a matching word. This is a feature (extrinsic motivator) that is used to fulfill the user's basic psychological needs for accomplishment, based on [\[1\]](#)[\[4\]](#)[\[5\]](#)[\[6\]](#)[\[7\]](#). See [Figure 1](#).

- Badges serve the same purpose as extrinsic motivators as points. Increase the user's motivation through the completion of goals [4][7][8]. See [Figure 11](#) and [Figure 14](#).
- Celebrate accomplishments, much like successful apps like Duolingo, Memrise and Elevate do. This is a type of feedback on progress that helps make the game more engaging according to [1][3][7]. This is a feature that was not included in my final version due to time constraints for their implementation, but they were supposed to look like [Figure 9](#) and [Figure 11](#).
- A timer is included to have short and timed game sessions, as suggested by [1] and [6]. The game is played for 60 seconds, to keep it casual and not boring [6]. See [Figure 1](#).
- The game makes use of appealing/cute graphics, scenery and colors, as suggested by [1], [5], [7], and [4]. If the app is useful and with clear objectives, but lacks in looks then this will be a bad design [5]. See [Figure 4](#) for an example.
- In keeping with the previous point, game characters should be humanized and appealing according to [1], [4] and [7]. Source [5] stresses this fact several times, *looks matters for games*. See [Figure 10](#).
- The game will make use of simple controls that do not obstruct the playing area [1], of adequate size and obvious functionality [4]. This is because confusing our users with lots of functionalities while gaming is not a good feature. This is why the main gaming screen only has four buttons: Pause, Submit, backspace and skip, with the latter two buttons represented with the most used graphic for each functionality. See [Figure 1](#).
- The game will have pause and autosave features, this in order to allow users to have freedom of coming back to the game at another appropriate time if needed, as suggested by [1].
- The game should not make use of annoying sounds, as suggested by [1]. The word pronunciations will be activated by the users if they choose to do it, and even a little beep sound that goes on when the time is running out can be disabled if considered not appropriate. See [Figure 8](#) and [Figure 13](#).
- The game will provide feedback on progress (e.g. scores, accuracy) as recommended by [1], [3] and [7]. Users' knowledge of their performance in a game can increase their motivation to continue playing. See [Figure 7](#) and [Figure 8](#).
- The game will provide notifications/reminders for playing, which can be deactivated by the users if they choose to. This is because this feature is linked with game addictiveness according to [2]. See [Figure 13](#)

- The game will provide streaks (e.g. longest streak, current streak, hours played). This is because [2] and [7] showed that providing streaks increases usage of the game. See [Figure 14](#).
- Users will be allowed to skip words they do not know. This is to keep in line with the recommendation from [3] and [9]: The user should not lose, since this can decrease the students' performance. This is why no points will be lost for skipping words (no-punishment).
- Another psychological need that must be satisfied is autonomy, according to [3]. This is why the game will allow users to choose their own path: They can choose to start at any of the six levels. If one level is too easy, they do not have to complete that in order to keep advancing in the game. See [Figure 6](#).
- The fact that there are no leaderboards, since this is not a social game in that sense, may prevent users from feeling frustrated/discouraged if they do not score too well on the leaderboard. So this might help to keep them playing according to [5].
- The game will provide a summary of the words studied during the session, along with its definitions. This is because as [6] states ``*mixing and varying educational elements and mechanics*'' is another best practice for creating a good educational game. See [Figure 8](#).
- Just like with the need for a Skip button, there is a need to include review sessions. This gives the opportunity to the users to revise the words they are having trouble with. As [9] states: "*The most successful games often provide multiple paths to success because when there is only one right way to do things all the way ... learners can quickly get frustrated*". Review sessions are then an alternative path to success. See [Figure 12](#).
- The fact that this is not a social game prevents it from satisfying a basic need for the users, that is, social interaction or relatedness, which according to [3] and [4] is an important feature to increase a game's addictiveness and user engagement. The rationale for this choice was discussed in [paragraph 5](#). This functionality was not part of my MVP and was not implemented.

Besides using the Jest library for unit testing, I conducted a survey (online questionnaire) on six users to get feedback on UX. I will use the questionnaire used by [5] in their research paper as a foundation for my own questionnaire, since it was based on the Technology Acceptance Model to test the User Acceptance. This way I can test for Perceived Ease of Use, Perceived Usefulness, Attitude Towards Game Design Elements and User Satisfaction.

Implementation

When the game loads the first time, a database is created with expo-sqlite using the `createAndPopulateDatabase` function (see `database.js` file) and a key named ‘`databaseInitialized`’ is saved with `AsyncStorage` so that the code can check if the database already exists or not during later re-runs. The database is one of the most important parts of the game because it stores all the information about the scores earned for each game, the list of words for each language, the words that have and have not been studied, the amount of times they have been correctly submitted and other information that is essential for the app. In the `database.js` file I coded functions to create and populate each table, and also to query them, which is an indispensable feature for this app.

After the user selects a game (English or Spanish for CEFR) and a Level (A1 through C2), the Game Screen starts loading. At this point the `English_Words` (or `Spanish_Words`) table is queried to get a list of 60 words at that level that are yet considered unknown, which are those which have a “Revision” value of less than 3. If there are not enough unknown words, that list is padded with known words. This is to make sure that the game never runs out of words to show during the 60 seconds of play. I must highlight the fact that these queries to the database are asynchronous and I had to make sure to call them in a specific order, and wait for their execution in order to call further functions that depended on the values that were results of previous functions. If this order was not strictly followed, the state of the database at the moment of the queries could not be certainly known and the results could be compromised.

The primary functionality of the app, the actual game, works as expected:

- A definition is shown
- Users can select letters to form a word.
- Users can only select the amount of letters necessary to create the matching word.
- Users can skip words they do not know, and in that case the word will change color to red.
- If the user inputs the right word then the color of the matched word changes to green (or red in case there is a mismatch).
- The timer changes color appropriately when the time changes. The seconds remaining in the game have an animation to make them appear/disappear.
- The game can be paused/unpaused.
- Points are awarded for each matched word.
- Animations are used to update the timer (which I implemented on top of the library used for the timer).
- The score increases are animated.
- Letters change opacity once they are selected and can not be selected again.

- The button to deselect a letter makes that letter available again (changes opacity) in the order they were selected in the first place.
- The Final score is animated in the FinishedGameLayoverView component. I implemented the animation by creating and updating a random array of numbers once the game is finished.
- The game correctly creates the database and queries it.
- The game creates the database if it does not exist already.
- A sound is played when there are three seconds left in the timer.
- Highest scores are updated when appropriate once the game is finished (see FinishedGameLayoverView.js lines 68-73).

The Game Screen sends the information about all the words seen during the game session to the Stats Screen (See [Figure 8](#)) through the use of React navigation and prop values. The information sent includes: the word, its definition, the status (either matched, skipped or unmatched), the score achieved, the name of the table used to get the list of words and the level being played. Once the game is over, the information about that game session is stored in the database: the score is saved in the Scores table, all the words that were correctly matched have their *Revision* column updated in the appropriate table (either English_Words or Spanish_Words). This information is used in the Stats Screen to show the latest ten scores and the highest score achieved in a particular game, in order to show this information in a Line Chart. A Pie Chart is used to show the accuracy obtained during the game session.

An important feature in the Stats Screen is that a user is able to mark a word as already known, no matter if it was matched, unmatched or skipped during the gameplay. A switch next to each word definition allows the user to mark that word as known so that it is not included in further game sessions (but all seen words are included in the revision sessions though). I consider this functionality quite useful because it allows the user to quickly mark words they already know so that they can progress quicker through the levels and also for testing.

The Streaks Screen is able to show the longest streak achieved for the games, that is, the longest amount of continuous days that the user has played. Since each score is stored in the Scores table along with a DateTime value, I implemented calculateCurrentStreak and calculateLongestStreak functions in the database.js file. To calculate both the current and longest streak it was necessary to first obtain the list of unique different dates that the user played (no matter which game or level), and must be noted that it should be different days since there could be several entries for the same day, this is why the *getSortedUniqueDates* function is used.

It was also necessary for me to implement a function to check if two dates were consecutive, without taking into account the time. I had to change the date values stored in the database to

their UTC values to account for certain errors in this calculation. Both entries are set at 12:00 am so that I could compare their difference in milliseconds to the total difference in milliseconds from one day to the next.

I had to make updates to the UI because of time constraints to implement all the functionalities I previously selected. I was not able to include the possibility to set up reminders (See [Figure 13](#)).

Evaluation

I tested my app in iOS (iPhone 14 Pro Max) and Android (Pixel_6_Pro_API_33_2) simulators on my computer, also in two Android devices with Expo installed (Pixel 5 pro and Samsung Galaxy Tab S6). While testing in these devices I noticed:

- The app runs faster in real devices than in the simulators (which was expected).
- The animation for the increase of the score during the game does not work properly on the Android simulator but it does perfectly in all the others.
- The audio played when the timer reaches 3 seconds sounds distorted in the Android simulator, but it plays well on all the other devices.

Due to time constraints and unforeseen circumstances (started a new job), I was left with insufficient time to implement the unit testing with the JEST library. Because of this reason I limited the amount of words included in the course for Spanish Words, that version of the game is shorter. Since the game requires to “know” all the words in the level in order to grant the badge for that particular level, and since the actual list is quite long (> 1000), it would take a longer time for me and you as testers to check if in fact a badge is awarded upon completion of a level.

So I limited the amount of words per level to a bit more than 60 words, this way you can play the game and skip words over and over (to loop through more of them in the same session) and mark them as known once you reach the Stats Screen. This was an easy way to test that the information shown in the Course Screen ([Figure 6](#)) was actually updating (the progress bars, the completeness percentage and the amount of studied words shown at each level). In this way it was easy to check that a badge is awarded for each level once the completion percentage reaches 100% (the progress bar is replaced by a badge in the Course Screen). See lines 79-93 in the GameLevelsComponent.js file for detailed information.

Besides my own testing, I had five other people (friends and family members) to test my application. I shared a survey with them once they have used the game at least 3 times. The questions in the survey had a punctuation between 1 and 5 (strongly disagree to strongly agree

scale) and a section for comments on the reason for their answer. These were the questions and answers:

- 1) I consider that the game is enjoyable and fun to play: 100% of users responded with a score of 5. I recognize that this might be biased because of the fact that these are my relatives and close friends, but I did ask them to give me their most honest opinions.
- 2) I consider that the animations included in the game are appropriate: three out of five people responded with a score of 4 and two people responded with 5. When asked why they gave that punctuation, these three people responded with a similar answer, that more animations are needed to celebrate accomplishments. The two people who responded with 5 were the oldest ones in the group (> 50 years old), which might be the reason why they did not believe that more animations were needed since they are not so used to playing mobile games.
- 3) I consider that the game is easy to understand since the first time playing it: three out of five people responded with 5 and the other two responded with a score of 3. When asked the reason for their responses the group of score 5 responded that they had previous experience with similar games and that helped them quickly navigate my app. Of the other two people, one person thought that the game did not allow enough time to guess a word and one person thought that the first time playing it was difficult because he did not have previous experience.
- 4) I consider that earning badges once I completed a level is an enjoyable experience: 4 of 5 people responded 3 to this question because none of them actually finished a level, and 1 person responded 4. So I can not ascertain that earning badges is actually an enjoyable experience for the users.
- 5) I consider that the imagery used in the game is appealing: 100% of users responded with a score of 5.
- 6) I consider that the Instructions for the game are clear: four out of five people responded with a score of 5 and one with a score of 3. When asked why, the person who gave a score of 3 said it was because she did not read the Instructions before playing the game the first time, which I do not consider as a reason to score a 3 in this question but everyone is allowed their own reasoning.
- 7) I consider that this game would be useful to help me learn new words: three out of five responded with a score of 3, when asked why their opinions varied: "I do not like to play games on my phone quite that much", "I prefer other ways of learning" and "it is so hard for

me to come up with a word in such a short time". One responded with a score of 5, which was expected because I know she loves these kinds of games, so her answer might be biased. The fifth one responded with a score of 4 and did not give a comment.

Overall, I believe that the results obtained in the survey demonstrate that my application meets the criteria I set for it during the design phase, even though many adjustments had to be made.

After the survey, I asked my users to provide ideas to improve the application, some of them I already knew (some functionality that was included in my design but could not implement) and some of them about the quality of the definitions:

- Sometimes the given definitions are too long.
- The definitions for words in Spanish should improve. This was expected since I just translated the English definitions into Spanish using ChatGPT (because I did not find freely available ones).
- Once you mark a word as known in the Stats Screen and you click the Home or Play button, there is no way of setting that word back to unknown. I did not realize that when testing my app, this is a bug that I miss.
- A sort of animation should be shown when the user gets a new highest score and when a badge is earned. I did know this one, but could not implement it because of time constraints, but it was something that I did account for during my design (See [Figure 9](#) and [Figure 11](#)).
- The game should have the option to change its main language. This was something I accounted for during my design, but I did not have enough time to implement since I did not do it that way since the start.
- Sounds for the words are missing. Four out of five of my users asked for this in particular because they are not native English speakers and they considered that being able to listen to the pronunciation of a word would make the learning process better.

Conclusion

I believe that my project meets the majority of the initial expectations, even though I had to make certain adjustments in order to expedite the completion of the final application. Developing a mobile application with React Native was quite a fun an educational experience for me, even though thanks to my previous knowledge of React Native on how to properly use its components

to create an appealing User Interface, I acknowledge that working with database queries and animations was a challenge due to their asynchronous nature.

There are some improvements that the current application requires, nevertheless most of these improvements are about features that I had already considered to include, so I know that it was not because I lacked the knowledge necessary to create a better application, I just needed more time. This is something I did not take into consideration when planning for the project since at the time I was a full time student, and I started a full time job in the middle of the semester, which hindered my scheduling quite a bit.

Nevertheless, that was something I should have been more prepared for. Thanks to this project I am more aware of the fact that it is essential for us as professionals to remain prepared for all kinds of unexpected challenges that may arise during a project's development cycle, and that allocating extra time in my planning will allow me to proactively address these unforeseen issues, so that I can have a smoother and less rushed project execution.

The link to my code repository in Google Drive is [this](#).

REFERENCES

- [1] Scolastici, Claudio, and David Nolte.(2013). *Mobile Game Design Essentials*, Packt Publishing. Chapter 11.
- [2] Abbasi BQ, Awais S. (2022). *Playing mind gamification : Theoretical evidence of addictive nature of gamification and identification of addictive game elements used in mobile application design*. Dissertation (MSc). JÖNKÖPING University, School of Engineering.
- [3] F. S. Rahayu, L. E. Nugroho and R. Ferdiana, "E-Learning Gamification Model using Video Game Elements that Influence Engagement or Addiction," *2022 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, Surabaya, Indonesia, 2022, pp. 220-226, doi: 10.1109/ISITIA56226.2022.9855202. Publisher IEEE.
- [4] YANG, Q. and GONG, X. (2021) . The engagement–addiction dilemma: an empirical evaluation of mobile user interface and mobile game affordance. *Internet Research*, **31**(5), pp. 1745-1768.
- [5] Q. Mi, J. Keung, X. Mei, Y. Xiao and W. K. Chan. (2018) "A Gamification Technique for Motivating Students to Learn Code Readability in Software Engineering," *2018 International Symposium on Educational Technology (ISET)*, Osaka, Japan, pp. 250-254, doi: 10.1109/ISET.2018.00062.

- [6] Svitlana Varaksina, Ivan Dyshuk. "Educational Game Development: How to Make Learning Games for Kids". MINDSTUDIOS. <https://themindstudios.com/blog/educational-game-development/>. Accessed 09 May 2023.
- [7] video-game-addiction.org. "What Makes a Video Game Addictive?". video-game-addiction.org. <https://www.video-game-addiction.org/what-makes-games-addictive.html>. Accessed 09 May 2023.
- [8] Hamari, Juho. (2017). "Do badges increase user activity? A field experiment on the effects of gamification". Computers in Human Behaviour., 71 (2017), pp. 469-478, 10.1016/j.chb.2015.03.036.
- [9] Shauna Carson, (2021). "Top 6 Features of Successful Game-Based Learning". SkillBuilder. <https://skillbuilder.ca/blog/top-features-game-based-learning>. Accessed 08 May 2023.
- [10] Csikszentmihalyi, M., (1990). Flow: The psychology of optimal experience (Vol. 1990). New York: Harper & Row.
- [11] NG, B.K., SUAIB, N.M., SIHES, A.J., ALI, A. and SHAH, Z.A. (2020). *Educational mobile game for learning English words*. IOP Conference Series.Materials Science and Engineering, 979(1).
- [12] Nur Marissa Vee Senap, Roslina Ibrahim. (2019). *A Review of Heuristics Evaluation Component for Mobile Educational Games*, Procedia Computer Science, Volume 161, pp. 1028-1035, ISSN 1877-0509.

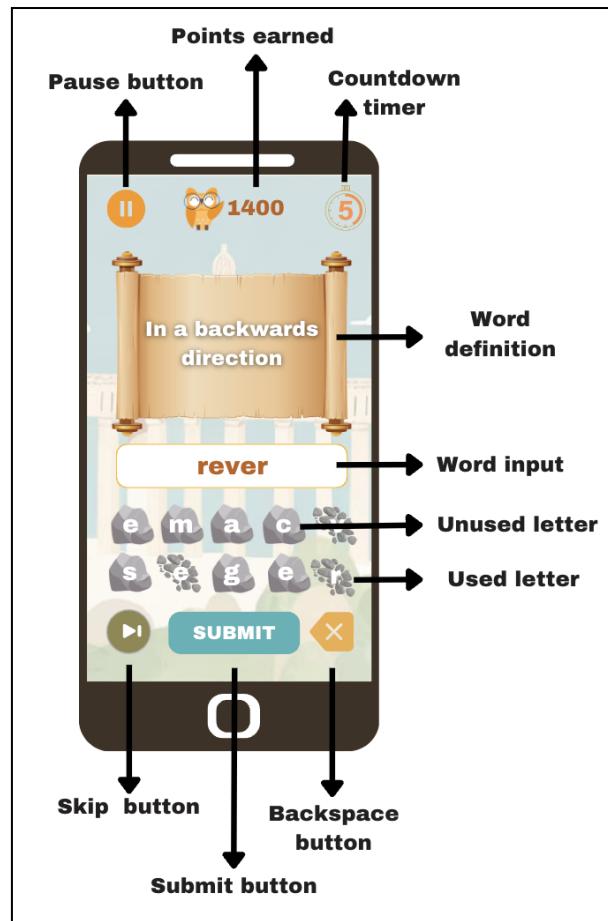


Figure 1. Game Screen

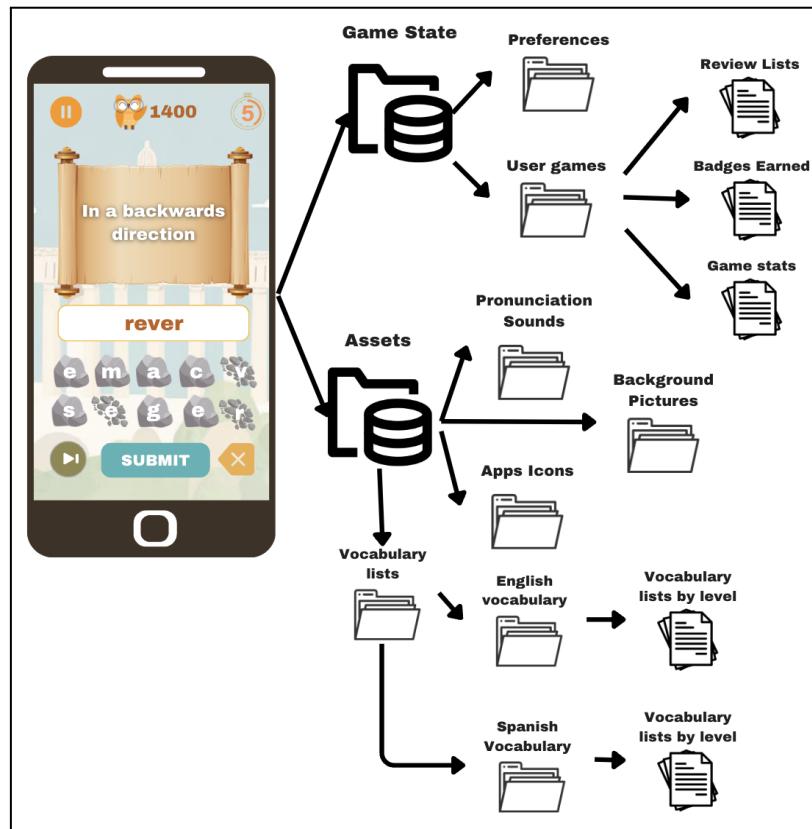


Figure 2. Data sources for application

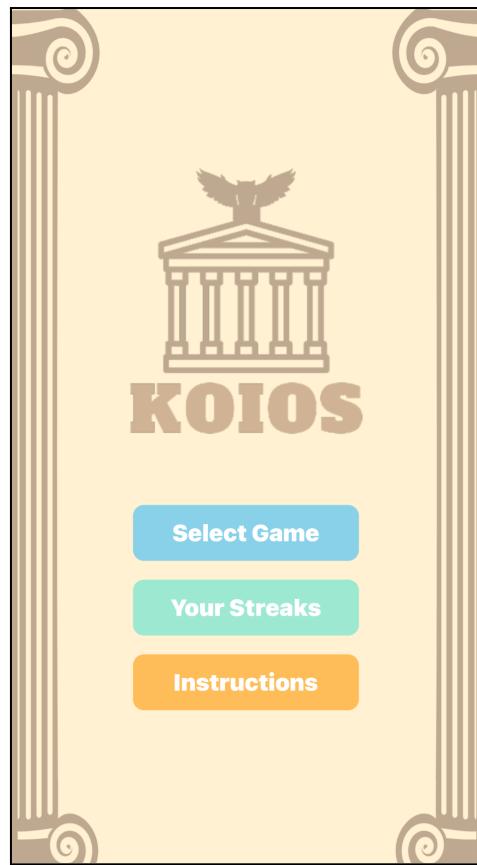


Figure 3. App Home Screen



Figure 4. Previous design for Courses Screen

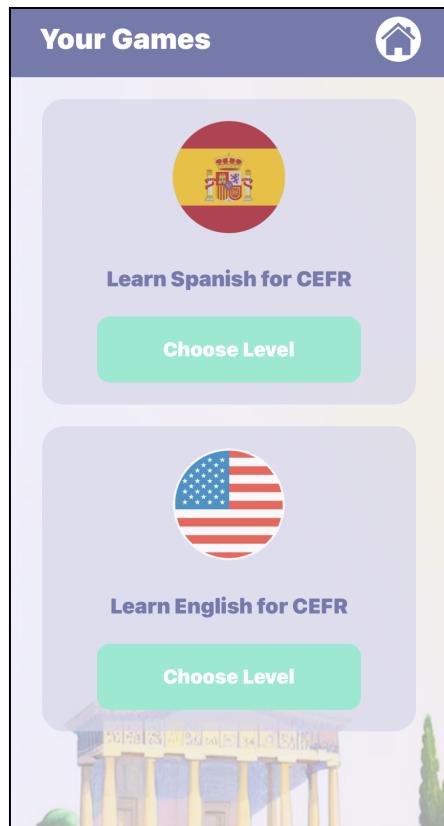


Figure 5. App showing games started by the user (can be one or two games)

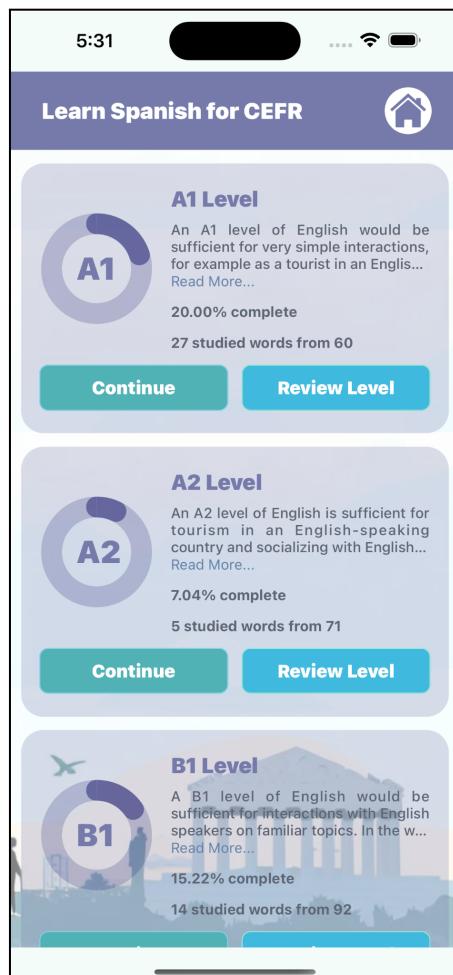


Figure 6. Game Levels showing completeness percentage.

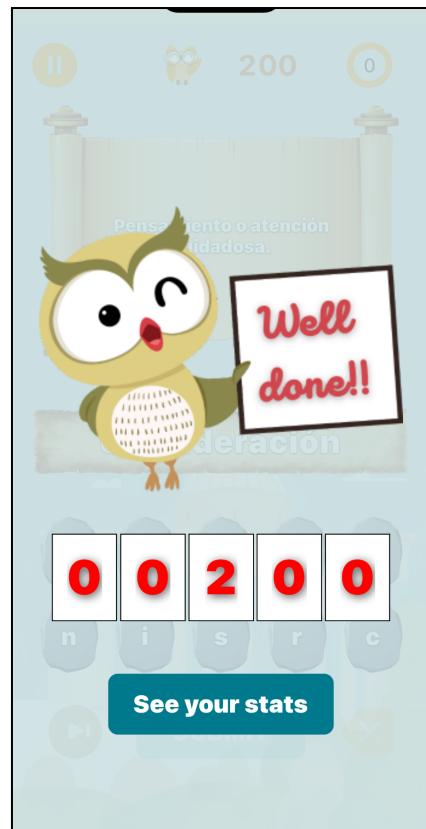


Figure 7. Screen shown when time runs out in game

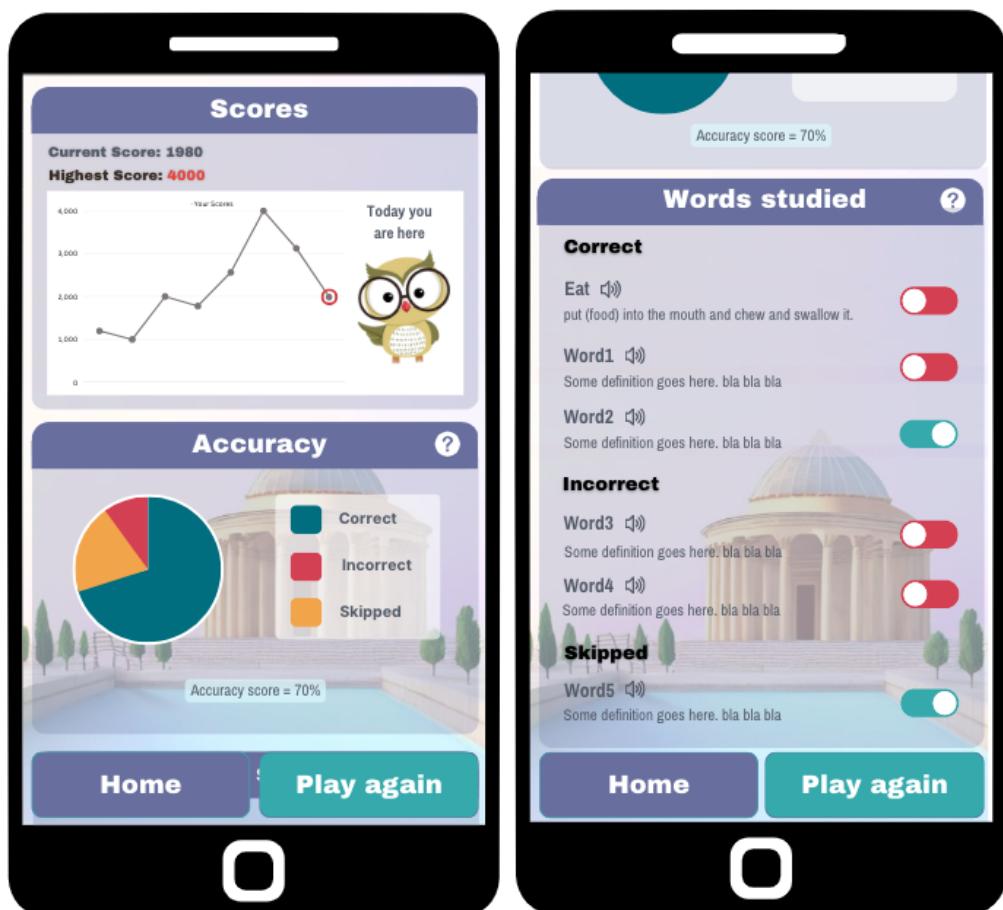


Figure 8. Screen showing the stats from the game session



Figure 9. Screen to celebrate new high score accomplishment

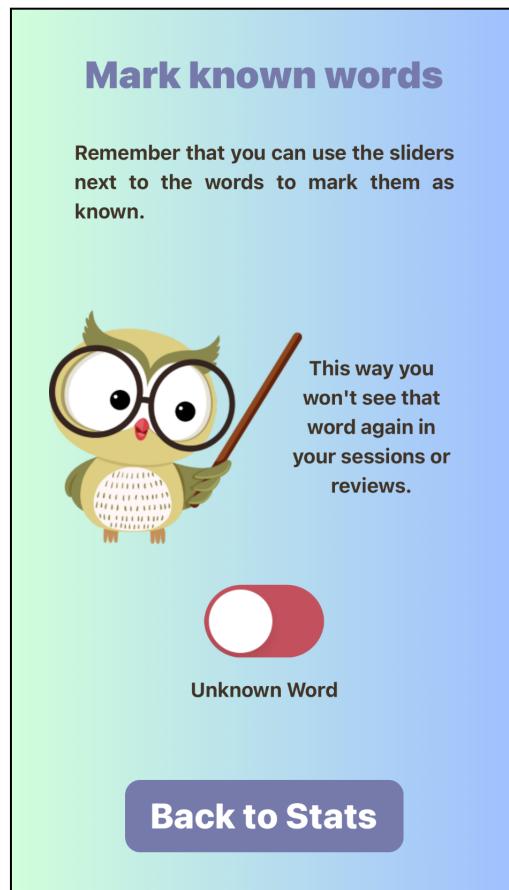


Figure 10. Screens showing information about the game to users.



Figure 11. Screen to celebrate the earning of a badge.

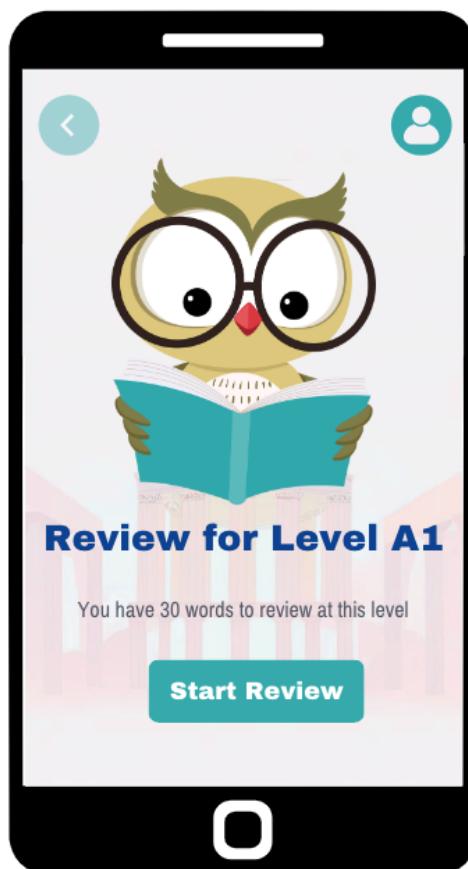


Figure 12. Screen shown when user wants to make a review session (not implemented)

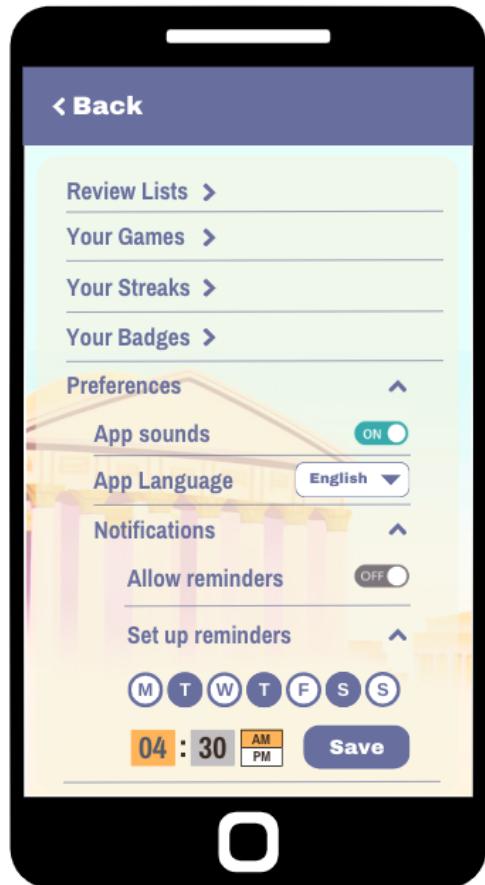


Figure 13. Preferences (not implemented).

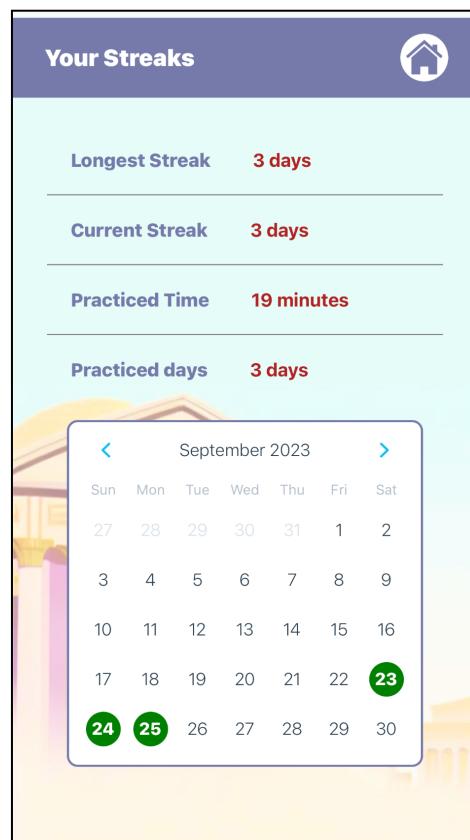


Figure 14. Screen showing user's streaks



Figure 15. Screen showing badges earned/unearned

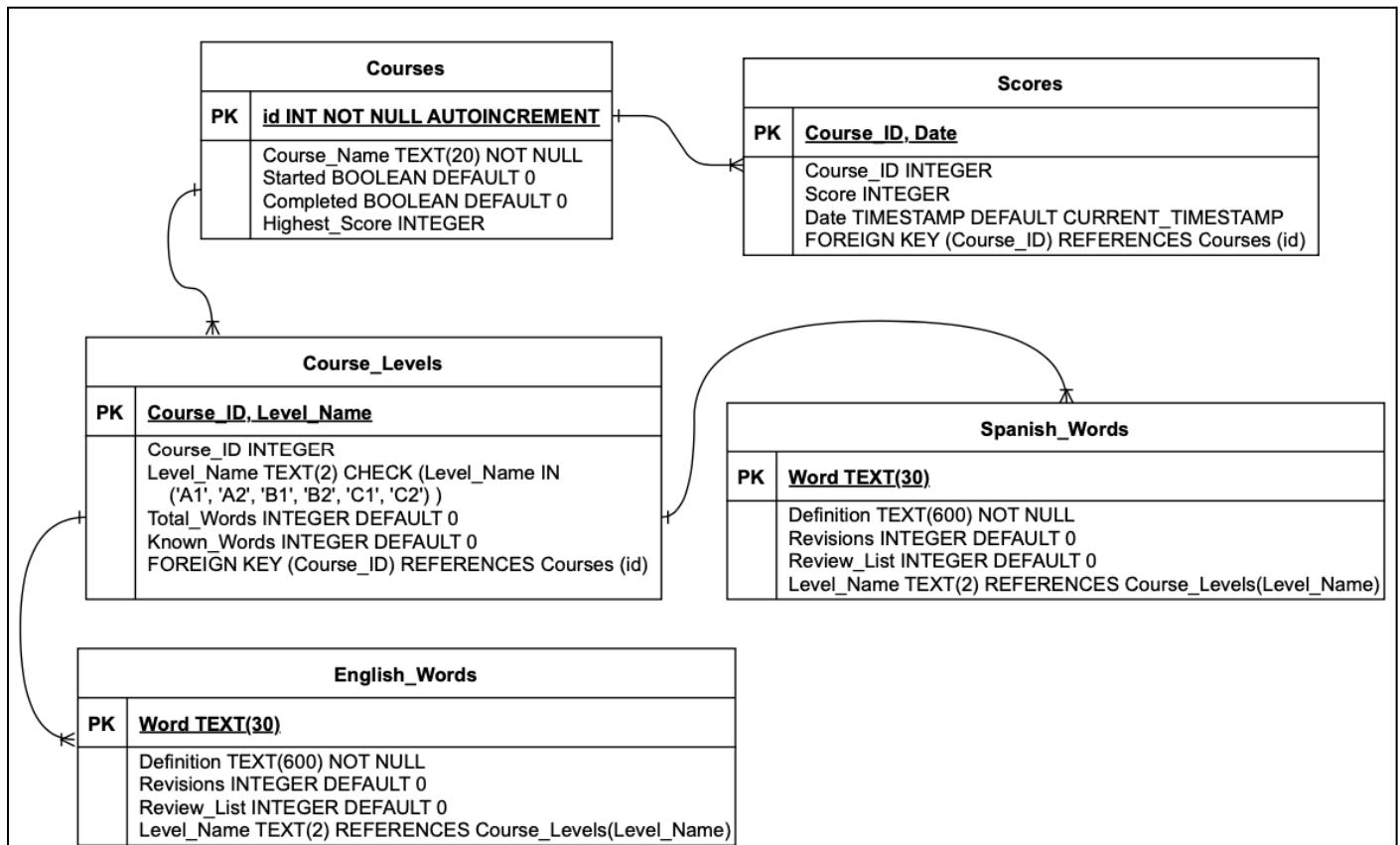


Figure 16. Database design