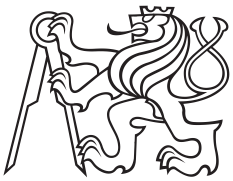
Bachelor Project



Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Computer Graphics and Interaction

Gamification for English Vocabulary Learning

Petr Nejedlý

Supervisor: Ing. Ivo Malý, Ph.D.
Field of study: Software Engineering and Technology
Subfield: Business Informatics
May 2025

BACHELOR‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Nejedlý Petr** Personal ID number: **516155**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Computer Science**

Study program: **Software Engineering and Technology**

Specialisation: **Business informatics**

II. Bachelor’s thesis details

Bachelor’s thesis title in English:

Gamification for English vocabulary learning

Bachelor’s thesis title in Czech:

Gamifikace ve výuce anglických slovíček

Name and workplace of bachelor’s thesis supervisor:

Ing. Ivo Malý, Ph.D. Department of Computer Graphics and Interaction FEE

Name and workplace of second bachelor’s thesis supervisor or consultant:

Date of bachelor’s thesis assignment: **29.01.2025** Deadline for bachelor thesis submission: **23.05.2025**

Assignment valid until: **20.09.2026**

Head of department’s signature

prof. Mgr. Petr Páta, Ph.D.
Vice-dean´s signature on behalf of the Dean

III. Assignment receipt

The student acknowledges that the bachelor’s thesis is an individual work.
The student must produce his thesis without the assistance of others, with the exception of provided consultations.
Within the bachelor’s thesis, the author must state the names of consultants and include a list of references.

Date of assignment receipt

Student’s signature

BACHELOR‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Nejedlý Petr** Personal ID number: **516155**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Computer Science**

Study program: **Software Engineering and Technology**

Specialisation: **Business informatics**

II. Bachelor’s thesis details

Bachelor’s thesis title in English:

Gamification for English vocabulary learning

Bachelor’s thesis title in Czech:

Gamifikace ve výuce anglických slovíček

Guidelines:

Review the English Mind app. Describe the learning concepts on which it is based. Suggest other learning concepts that this app can be extended with based on analysis of existing applications and literature. Design specific extensions in the form of prototypes and describe their use in the form of appropriate scenarios or use cases. Evaluate prototypes with users.
Analyze results of user tests. Select at least two suitable prototypes for implementation into English Mind application. Implement the prototypes and evaluate the application in a long term user study with at least 5 people.

Bibliography / sources:

Reza Hadi Mogavi, Bingcan Guo, Yuanhao Zhang, Ehsan-Ul Haq, Pan Hui, and Xiaojuan Ma. 2022. When Gamification Spoils Your Learning: A Qualitative Case Study of Gamification Misuse in a Language-Learning App. In Proceedings of the Ninth ACM Conference on Learning @ Scale (L@S '22). Association for Computing Machinery, New York, NY, USA, 175–188. https://doi.org/10.1145/3491140.3528274
Shortt, Mitchell & Tilak, Shantanu & Kuznetcova, Irina & Martens, Bethany & Akinkuolie, Babatunde. (2021). Gamification in mobile-assisted language learning: A systematic review of Duolingo literature from public release of 2012 to early 2020. Computer Assisted Language Learning. 36. 10.1080/09588221.2021.1933540.
Dehganzadeh, H., Dehganzadeh, H. Investigating effects of digital gamification-based language learning: a systematic review. Journal of English Language Teaching and Learning, 2020; 12(25): 53-93. doi: 10.22034/elt.2020.10676
Terence Govender and Joan Arnedo-Moreno. 2021. A Survey on Gamification Elements in Mobile Language-Learning Applications. In Eighth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'20). Association for Computing Machinery, New York, NY, USA, 669–676. https://doi.org/10.1145/3434780.3436597

FAKULTA ELEKTROTECHNICKÁ
FACULTY OF ELECTRICAL ENGINEERING
Technická 2
166 27 Praha 6

DECLARATION

I, the undersigned

Student's surname, given name(s): Nejedlý Petr
Personal number: 516155
Programme name: Software Engineering and Technology

declare that I have elaborated the bachelor's thesis entitled

Gamification for English vocabulary learning

independently, and have cited all information sources used in accordance with the Methodological Instruction on the Observance of Ethical Principles in the Preparation of University Theses and with the Framework Rules for the Use of Artificial Intelligence at CTU for Academic and Pedagogical Purposes in Bachelor's and Continuing Master's Programmes.

I declare that I used artificial intelligence tools during the preparation and writing of this thesis. I verified the generated content. I hereby confirm that I am aware of the fact that I am fully responsible for the contents of the thesis.

In Prague on 20.05.2025

Petr Nejedlý
.....
student's signature

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Ing. Ivo Malý, Ph.D., for his guidance and constructive feedback throughout the development of this thesis.

I am also grateful to all the participants who took part in the user testing sessions, providing valuable insights that helped improve the application.

Special thanks go to my family and friends for their continuous support and understanding during the course of this project.

Declaration

I hereby declare that this thesis is my own work and that I have cited all sources of information in accordance with the guidelines of the Faculty of Engineering, CTU in Prague.

In Odense, Denmark, May 20, 2025

Abstract

This thesis explores the integration of gamification elements into English Mind, a language learning mobile application, to enhance user engagement and learning outcomes. Through analysis of existing methodologies and competitor applications, four key gamification features were identified and implemented: diversified flashcard types, vocabulary progress tracking, post-practice analytics, and a streak system.

The implementation involved restructuring the system architecture using a monorepo approach with Melos, enabling parallel development of the main application and a demonstration prototype. User testing showed strong satisfaction, confirming the effectiveness of the implemented features.

The results demonstrate that carefully designed gamification elements can significantly improve a language learning application's competitive position while maintaining its core learning methodologies.

Keywords: gamification, language learning, vocabulary acquisition, mobile application, Flutter

Supervisor: Ing. Ivo Malý, Ph.D.

Abstrakt

Tato práce se zabývá integrací gamifikačních prvků do mobilní aplikace English Mind zaměřenou na výuku anglických slovíček s cílem zvýšit zapojení uživatelů a zlepšit výsledky výuky. Prostřednictvím analýzy stávajících metodik a konkurenčních aplikací byly identifikovány a implementovány čtyři klíčové prvky gamifikace: rozmanité typy kartiček, sledování pokroku jednotlivých slovíček, analýza po skončení procvičování a streak systém.

Implementace zahrnovala restrukturalizaci architektury aplikace s využitím monorepo přístupu pomocí systému Melos, což umožnilo paralelní vývoj hlavní aplikace a demonstračního prototypu. Uživatelské testování ukázalo vysokou míru spokojenosti a potvrdilo účinnost implementovaných funkcí.

Výsledky ukazují, že pečlivě navržené gamifikační prvky mohou výrazně zlepšit konkurenční pozici aplikace pro výuku jazyků s ohledem na zachování jejích základních výukových metodik.

Klíčová slova: gamifikace, výuka jazyků, osvojování slovíček, mobilní aplikace, Flutter

Překlad názvu: Gamifikace ve výuce anglických slovíček

Contents

Figures

Tables

Chapter 1

Introduction

In today's fast-paced digital world, language learning applications have become an essential tool for individuals seeking to enhance their vocabulary and language skills. However, maintaining user engagement and motivation over time remains a significant challenge [2]. This project addresses this issue by exploring the integration of gamification elements into English Mind [3], a language learning mobile application. Aiming to transform the learning experience into a more engaging and rewarding journey.

English Mind, the subject of this study, is built on a well-designed learning approach that effectively supports language acquisition. However, when compared to leading competitors [4, 5, 6], it falls short in terms of gamification elements. This lack of engaging features may negatively impact user motivation and retention, as users are increasingly drawn to applications that offer interactive and rewarding experiences [7, 2].

This project aims to enhance user engagement and motivation by carefully selecting and implementing gamification features that align with English Mind's core pedagogical principles. As the application's developer, this research benefits from an in-depth understanding of its architecture and pedagogical foundations, ensuring that any added gamification elements complement rather than compromise the existing learning approach.

This thesis is organized into eight chapters. After introducing the motivation and success criteria (Chapter 1.1), the current state and learning approach of English Mind are analyzed (Chapter 2). Gamification principles in leading applications are then examined (Chapter 3), followed by the selection of features for implementation that align with English Mind's core pedagogical principles (Chapter 4). The selected features are validated through user testing of an interactive prototype (Chapter 5), implemented (Chapter 6), and thoroughly tested by end users (Chapter 7). Finally, the findings are summarized, success criteria are evaluated, and future improvements are suggested (Chapter 8).

1.1 Success Criteria

The main goal of this work is to enhance English Mind by adding gamification features that align with its core pedagogical principles while maintaining the effectiveness of its learning approach. Based on this goal, the evaluation of this work's success is structured around three principal dimensions: competitive positioning, user feedback, and implementation success.

1. **Enhancement of Competitive Positioning**
   a. Analyze the English Mind vocabulary learning approach to understand the key principles of the application.
   b. Conduct analysis of gamification features used by English Mind and leading competitors.
   c. Identify opportunities for enhancing the gamification of English Mind.
2. **Positive User Feedback on Gamification Features**
   a. Design gamification features and elements that are on par with or surpass those of leading competitors and align with English Mind's core pedagogical principles.
   b. Conduct user testing of an interactive prototype showcasing the new gamification features, aiming for positive feedback.
3. **Successful Implementation of Gamification Features**
   a. Implement the newly designed gamification features while preserving the application's existing learning approach.
   b. Conduct a long-term testing with end users to ensure functionality and user satisfaction.

Chapter 2

Mobile Application English Mind

English Mind is a mobile application available on Android [8] and iOS [9] platforms that focuses on learning English vocabulary. To achieve high efficiency in learning new English vocabulary, the application combines three main teaching methods on which it is based [3]:

- Frequency list of English vocabulary
- Active recall utilizing flashcards
- Spaced repetition system

These methods create a structured and efficient approach to mastering vocabulary with minimal effort, setting the app apart from its competitors in the field of vocabulary learning apps.

2.1 Frequency List

A frequency list ranks words according to their occurrence in common texts and speech. This approach is particularly valuable in vocabulary teaching for foreign language learners, as learning high-frequency words first enables students to comprehend more of their target language earlier in their studies.

Research by Nation [10] demonstrates the effectiveness of this approach. Mastering the 1,000 most frequent word families enables understanding of 78-81% of written text, while 8,000-9,000 word families are required for 98% comprehension. This data underscores the importance of prioritizing high-frequency vocabulary in language acquisition.

The frequency list implementation in English Mind allows users to browse words and assign them one of three states: UNSEEN (default), KNOWN, and LEARNING. Users can modify these states as shown in Figure 2.1. Words marked as "LEARNING" are practiced using flashcards and scheduled for review through spaced repetition.

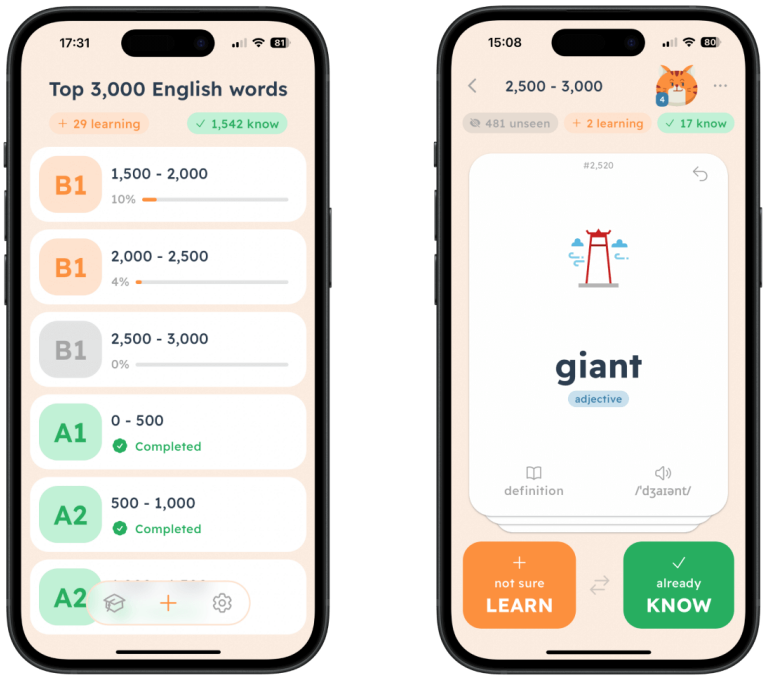


Figure 2.1: English Mind - Frequency List

2.2 Active Recall and Flashcards

Active recall is a learning method where students attempt to retrieve information without reference to the source material, contrasting with passive review where information is simply reread. Research from Washington University [1] demonstrates active recall's superiority for long-term retention: in a study of 120 students, active recall consistently outperformed passive review across various time intervals (5 minutes, 2 days, and 1 week), as shown in Figure 2.2.

Flashcards represent a practical implementation of active recall, where information is split between two sides of a card, forcing active retrieval before verification.

English Mind implements flashcard-based active recall by presenting vocabulary words on the front and comprehensive word information (definition, usage examples, pronunciation, and native language translation) on the reverse, as shown in Figure 2.3. Users must attempt to recall the word's meaning before revealing the reverse side.

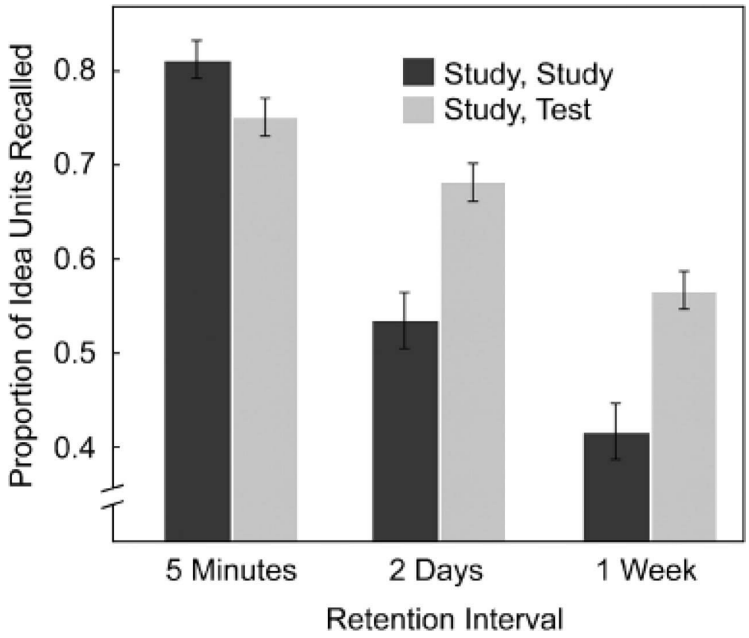


Figure 2.2: Comparison of retention rates between active recall and passive review methods across different time intervals [1]

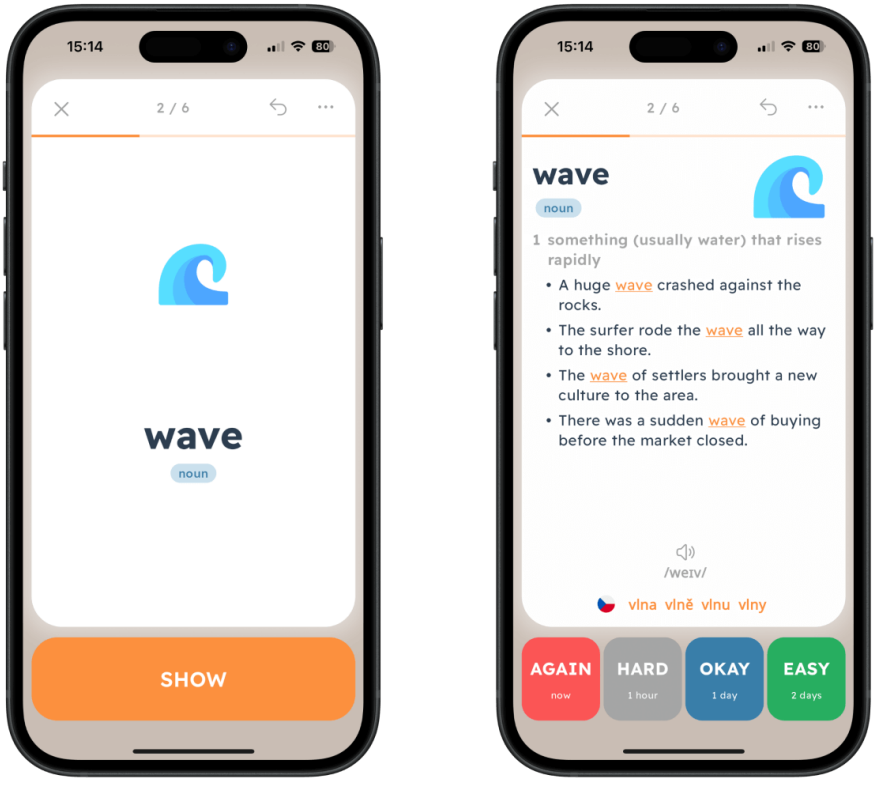


Figure 2.3: English Mind - Active Recall Utilizing Flashcards

2.3 Spaced Repetition System (SRS)

Spaced Repetition System (SRS) optimizes learning by adjusting intervals between review sessions based on recall performance. This method builds on Ebbinghaus's forgetting curve theory [11], which demonstrates that information retention improves when review occurs just before predicted forgetting. Research shows that SRS increases learning efficiency by optimizing review timing and reducing unnecessary repetition [12].

The application implements SRS through a four-button feedback system (AGAIN, HARD, OKAY, EASY) that appears after each flashcard review, as shown in Figure 2.4. Each button adjusts the next review interval: AGAIN and HARD decrease it, while OKAY and EASY increase it. Words consistently recalled over several months automatically transition from "LEARNING" to "KNOWN" status.

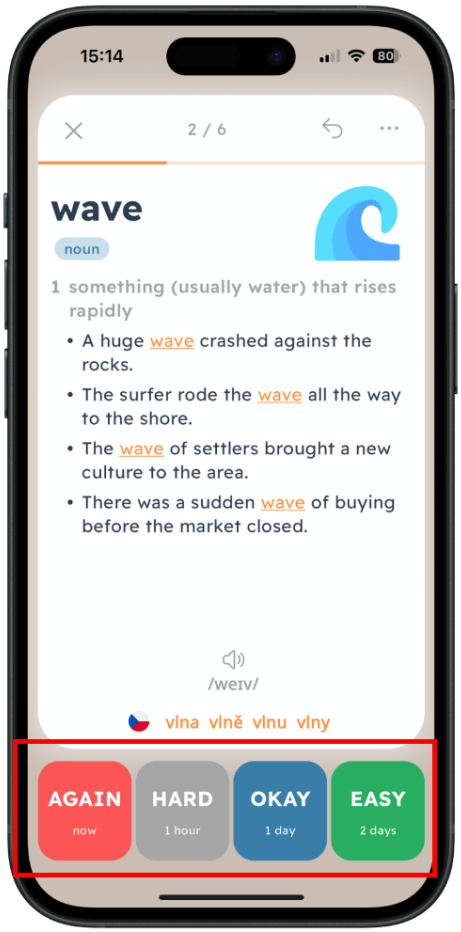


Figure 2.4: English Mind - SRS

2.4 Application Workflow

The application's learning process consists of two primary phases: vocabulary selection and practice. Figure 2.5 illustrates the hierarchical breakdown of these tasks.

Users first browse the frequency-ordered vocabulary list, marking words as either "KNOWN" (already mastered) or "LEARNING" (to be studied). This initial classification ensures that learning efforts focus on appropriate vocabulary. Subsequently, words marked as "LEARNING" enter the practice phase, where they are studied through flashcards and systematically reviewed using spaced repetition.

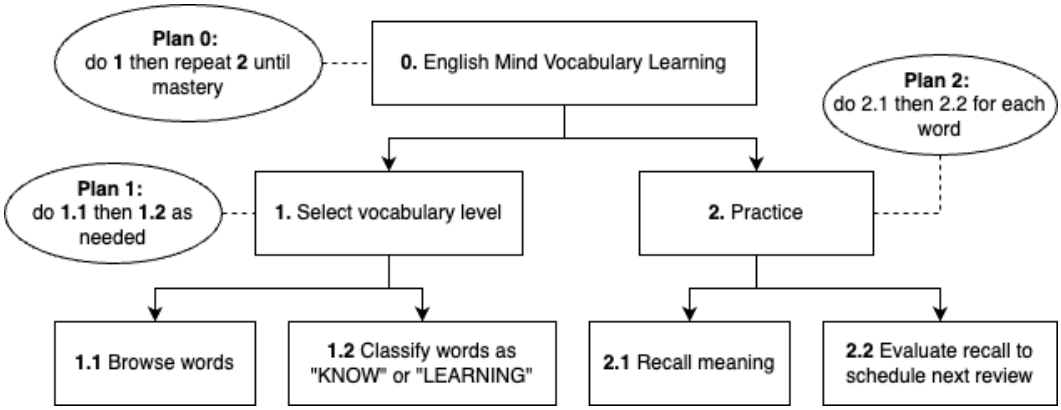


Figure 2.5: English Mind - Hierarchical Task Analysis of the core learning workflow

Chapter 3

Gamification Analysis of Language Learning Applications

Gamification is defined as ”the use of game design elements in non-game contexts” [7]. In educational settings, it enhances learning activities through game mechanics to increase student engagement and motivation. The field encompasses numerous concepts, including streaks, achievements, leaderboards, progress tracking, social elements, virtual currencies, and much more [2].

Traditional vocabulary learning methods often rely on mechanical repetition or passive reading, which can lead to decreased engagement and compromised learning outcomes. This challenge is particularly evident in language acquisition, where consistent practice is crucial for long-term retention. Gamification addresses these limitations by incorporating structured incentives for regular practice while maintaining learner engagement through various game mechanics.

While gamification offers numerous potential elements for implementation, this analysis focuses specifically on features that appear in language learning applications. This chapter first examines the current gamification features in English Mind, identifying areas for potential enhancement. It then analyzes three prominent language learning applications — WordUp [5], DuoCards [6], and Duolingo [4] — selected based on either their similarity to English Mind’s learning approach or their innovative implementation of gamification elements. The chapter concludes with a comparative analysis of these applications and identifies key opportunities for improvement.

3.1 English Mind

English Mind currently implements minimal gamification features, consisting primarily of two basic progress visualization elements:

- **Vocabulary Progress Overview**

  The application provides quantitative metrics of vocabulary acquisition progress, displaying the number of known words, words in active learning, and remaining words to be learned (see Figure 3.1). This statistical overview enables users to track their overall learning trajectory.

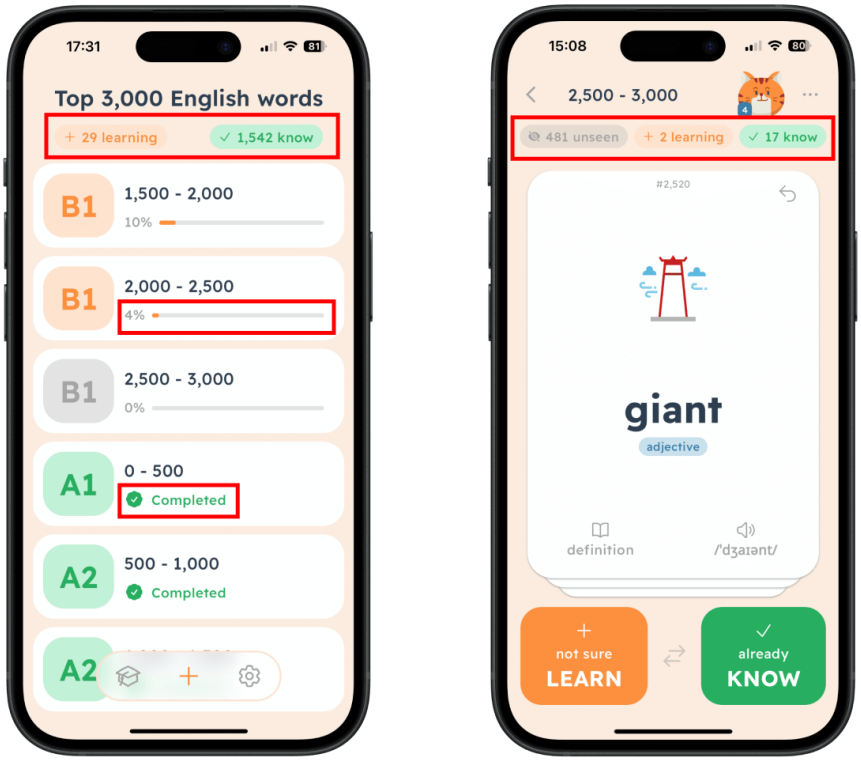


Figure 3.1: English Mind - Vocabulary Progress Overview

- **Session Progress Indicator**

  A linear progress indicator during flashcard practice sessions provides visual feedback on session completion status. This basic implementation offers immediate context for session progress.

This limited implementation of gamification elements suggests significant potential for enhancement through more sophisticated engagement mechanisms, which will be examined through comparative analysis in subsequent sections.

3.2 WordUp

WordUp represents a particularly relevant case study as it shares the core learning methodologies with English Mind, combining frequency lists with spaced repetition and flashcard-based practice. With over 5 million downloads on Google Play [13], its implementation of gamification elements provides comparative insights for potential enhancements to English Mind. The application implements three key gamification features:

- **Diversified Flashcard Types**

  WordUp employs four variants of flashcards, each targeting different aspects of vocabulary acquisition (see Figure 3.2). This variation in exercise types helps maintain engagement while addressing multiple aspects of language acquisition — meaning comprehension, recall, spelling accuracy, and listening skills.

  - Word-to-Definition matching
  - Definition-to-Word recall
  - Definition-to-Spelling association
  - Audio-to-Spelling transcription

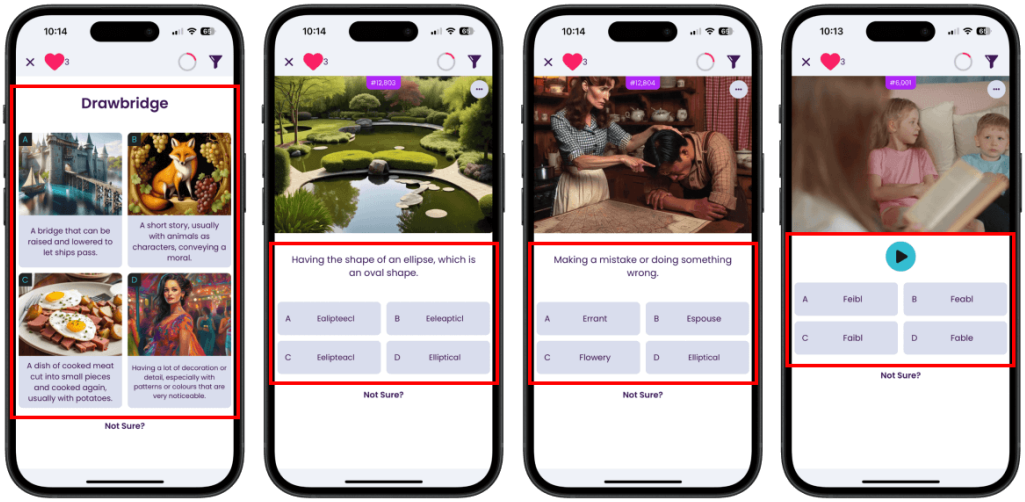


Figure 3.2: WordUp - Flashcard Type Variations

- **Individual Word Progress Tracking**

  The application implements a granular progress tracking system that visualizes individual word mastery through a numerical indicator (see Figure 3.3). This mechanism provides immediate feedback on learning progress at the word level.

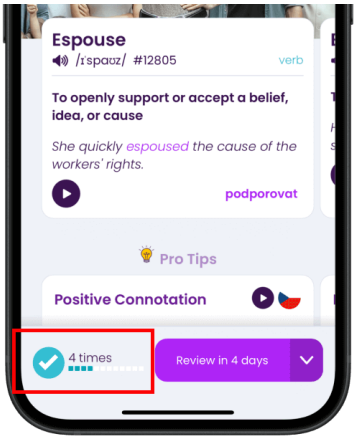


Figure 3.3: WordUp - Individual Word Progress Indicator

- **Time-Based Goals and Social Competition**

  The application incorporates daily practice goals measured in minutes, complemented by a leaderboard system (see Figure 3.4). This combination of personal goals and social comparison creates multiple motivation vectors for sustained engagement.

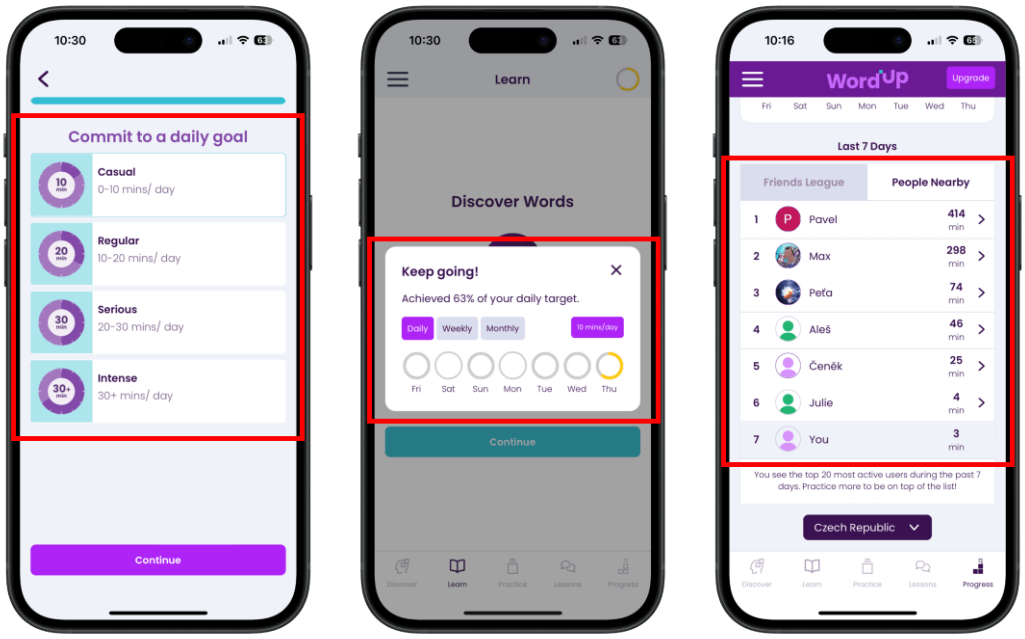


Figure 3.4: WordUp - Daily Goal and Leaderboard

3.3 DuoCards

DuoCards, with over a million downloads on Google Play [14], implements an SRS-based flashcard system for vocabulary acquisition. The application's gamification strategy centers on two primary mechanisms: diversified practice formats and an incentive-based progression system.

- **Diversified Practice Formats**

  The application implements four distinct flashcard variants that systematically address different aspects of vocabulary acquisition (see Figure 3.5). This methodological variation facilitates comprehensive vocabulary acquisition while maintaining engagement through task diversity.

  - Bidirectional translation exercises
  - Auditory comprehension and pronunciation exercises
  - Multi-item matching exercises for vocabulary reinforcement

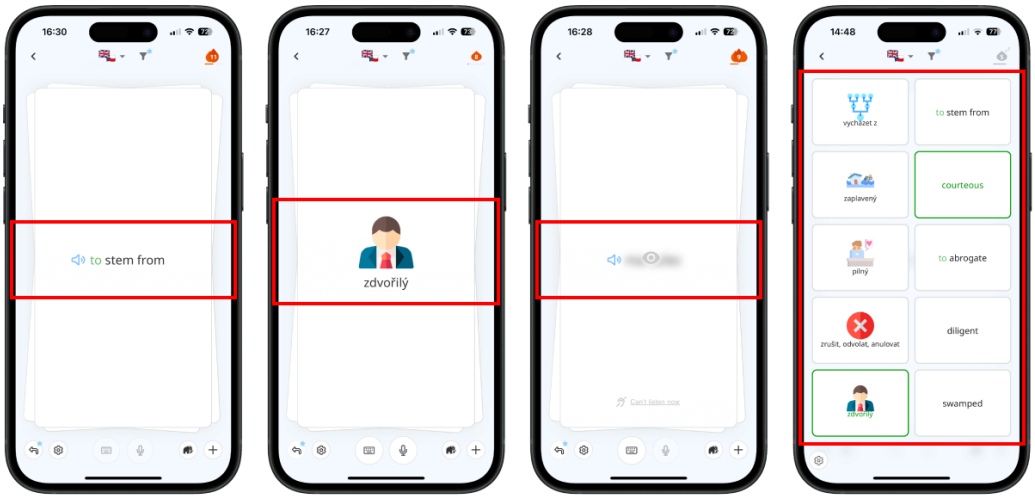


Figure 3.5: DuoCards - Variation in Practice Formats

- **Virtual Currency and Achievement System**

  The application implements an experience-based reward system (XP) integrated with a customizable mascot interface (see Figure 3.6). This gamification mechanism creates a tangible reward structure where practice completion and vocabulary acquisition yield currency for environmental customization, thereby reinforcing consistent engagement patterns.

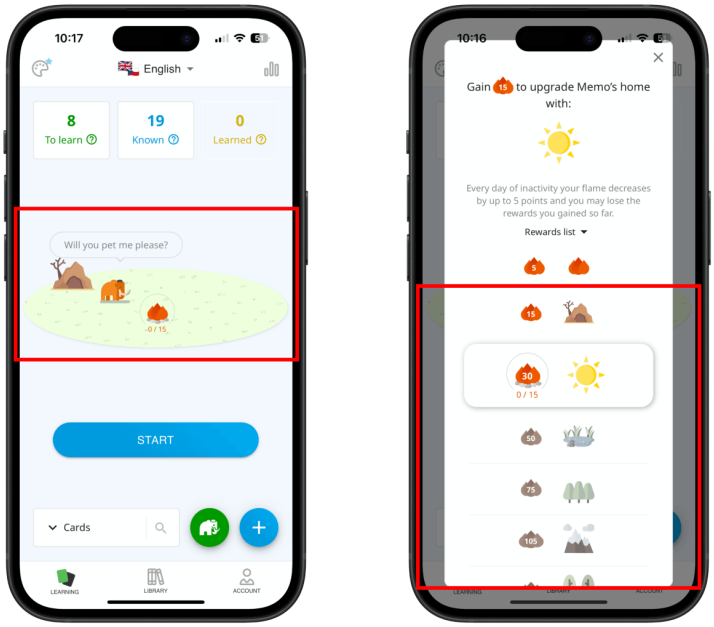


Figure 3.6: DuoCards - Achievement-Based Customization

3.4 Duolingo

Duolingo, with over 100 million monthly active users [15], has established significant precedent in digital language education gamification. While extensive research exists on Duolingo's comprehensive feature set, this analysis focuses on three key elements particularly relevant to English Mind's vocabulary learning context.

- **Exercise Diversification**

  The application implements multiple exercise variants to reinforce vocabulary acquisition. This methodological variation addresses multiple aspects of language acquisition while maintaining engagement through task diversity (see Figure 3.7).

  - Word-translation pair matching exercises
  - Speech recognition-based pronunciation assessment
  - Audio-to-text transcription for listening comprehension

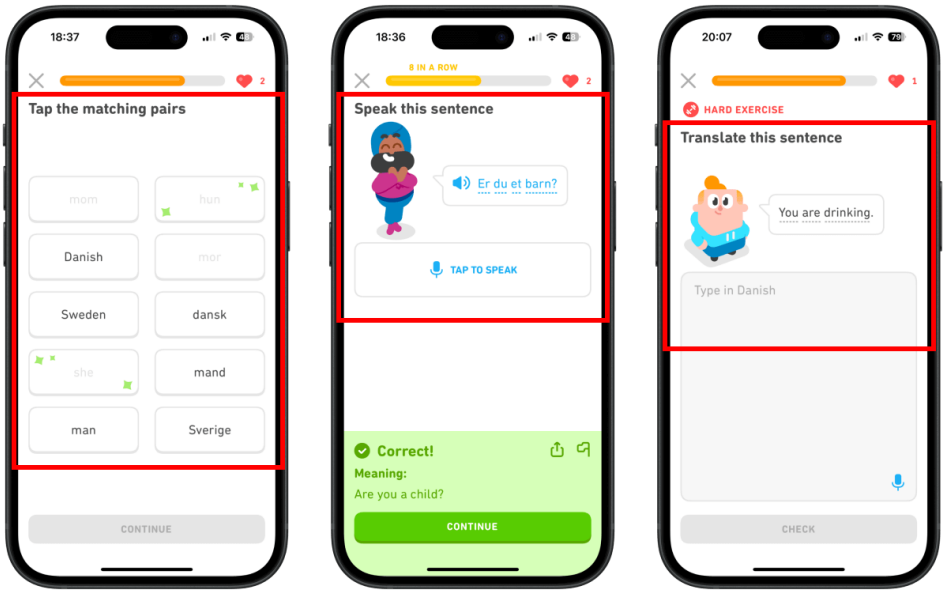


Figure 3.7: Duolingo - Exercise Types

- **Post-Session Analytics**

  The application provides immediate post-session feedback through quantitative metrics. This feedback mechanism facilitates progress tracking while reinforcing achievement through immediate performance visualization (see Figure 3.8).

  - Performance accuracy
  - Experience points acquired
  - Time investment
  - Streak metrics

- **Streak System**

  The application implements a streak-based tracking mechanism that incentivizes daily engagement. This system includes flexibility features (”Streak Freezes”) to accommodate occasional missed sessions while maintaining motivation. The effectiveness is demonstrated by retention metrics, with 20% of daily active users maintaining streaks exceeding one year [15]. The system is supported by a structured notification framework incorporating:

  - Streak maintenance alerts
  - Customizable practice reminders
  - Achievement milestone recognition

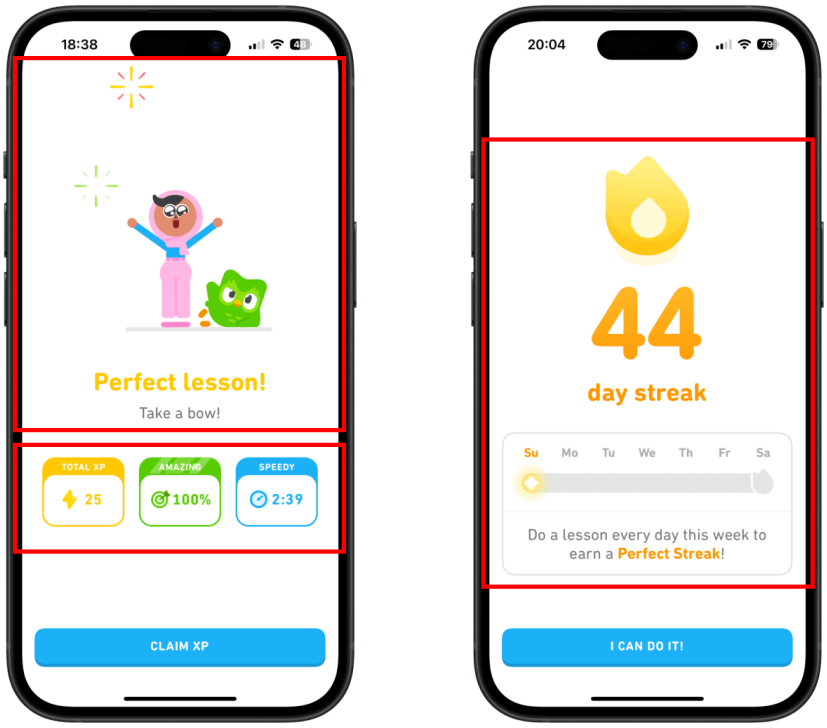


Figure 3.8: Duolingo - Post-Session Analytics

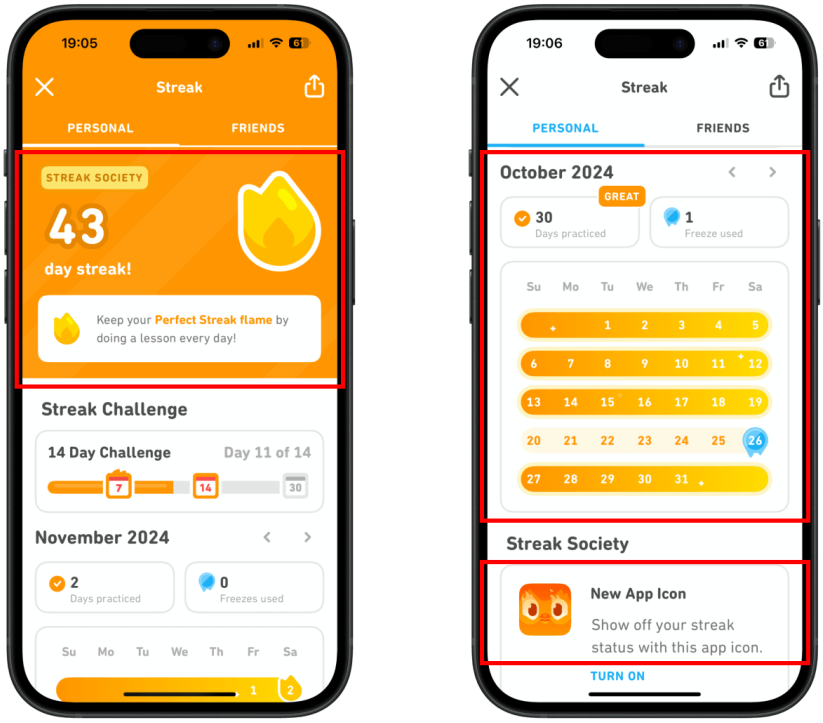


Figure 3.9: Duolingo - Streak System

3.5 Comparison and Findings

The comparative analysis of gamification features across the examined applications reveals distinct patterns in implementation approaches and feature adoption (Table 3.1). The analysis yields three primary findings:

1. Multiple exercise formats emerge as an industry standard, with WordUp, DuoCards, and Duolingo each implementing at least three distinct practice types. This contrasts with English Mind's single-format approach. The prevalence of varied exercise types suggests their effectiveness in both engagement and comprehensive vocabulary acquisition.

2. Progress visualization implementations vary in granularity. While all applications provide aggregate progress metrics, WordUp's individual word progress tracking and Duolingo's post-session analytics demonstrate more sophisticated approaches to progress feedback.

3. Engagement retention mechanisms vary across applications, with mainly Duolingo's streak system showing notable effectiveness.

Table 3.1: Comparative Analysis of Gamification Elements Usage

Gamification	English Mind	WordUp	DuoCards	Duolingo
Flashcard Types				
Exercise variety	—	✓	✓	✓
Recall meaning	✓	✓	✓	✓
Speech recognition	—	—	—	✓
Spelling assessment	—	✓	✓	✓
Matching definitions	—	✓	—	—
Matching translations	—	—	✓	✓
Progress Tracking				
Aggregate progress metrics	✓	✓	✓	✓
Practice session progress	✓	—	—	✓
Word-level analytics	—	✓	—	—
Post-practice analytics	—	—	—	✓
Engagement Features				
Streak system	—	—	✓	✓
Time-based goals	—	✓	—	—
Social comparison	—	✓	—	✓

Chapter 4

Proposed Solution Design

Through systematic analysis of existing language learning applications and their gamification implementations, this chapter presents a carefully curated selection of gamification features and a proposed solution design for English Mind.

The selection methodology prioritized features that have demonstrated effectiveness in user engagement or are commonly implemented in existing applications, while maintaining strong alignment with English Mind's core pedagogical principles. Identified features strategically enhance two critical aspects of the learning experience:

1. **Enhanced Practice Experience**
   - Diversified flashcard types
     - Word spelling
     - Word pronunciation
     - Word-Definition matching
     - Word-Translation matching
   - Post-practice review providing immediate feedback and reinforcement
   - Individual word progress tracking breaking the abstract concept of SRS into tangible stages
2. **Consistent Practice Motivation**
   - A streak system requiring daily engagement with new vocabulary

4.1 Diversified Flashcard Types

The current English Mind's flashcard system employs a single format where users view an English word and recall its meaning. While effective for basic vocabulary acquisition, comparative analysis of language learning applications reveals that the monotony of a single flashcard type can reduce practice effectiveness. The variety in practice formats aims to maintain user interest throughout longer practice sessions, potentially increasing both practice duration and frequency.

This section proposes an expanded word revision system with five distinct flashcard types (see Figure 4.1), each targeting specific learning objectives.

1. **Recall Meaning**

   The existing English Mind flashcard format, where users see an English word and recall its meaning, will be maintained as the foundation of the practice system. This format effectively tests basic word recognition and meaning recall (see Section 2.2).

2. **Match Words to Translations**

   Following successful implementations in applications like Duolingo and WordUp, users match English words with their corresponding translations, presented in sets of five pairs. This format reinforces the connection between words in the target and native languages.

3. **Match Words to Definitions**

   Similar to the translation matching, this format requires users to match English words with their definitions, reinforcing deeper understanding of word meanings. This format focuses on comprehension rather than translation.

4. **Spell Word**

   This exercise type presents users with a word's definition and a contextual sentence containing a blank space. Users must correctly spell the target word to complete the sentence. An audio pronunciation button is available as a hint if users cannot determine the required word. The system evaluates spelling accuracy and provides immediate feedback.

5. **Pronounce Word**

   Users are shown an English word and prompted to pronounce it correctly. The system evaluates pronunciation accuracy using speech recognition technology, providing binary feedback (correct/incorrect). Understanding that not all practice environments are suitable for speaking exercises, users can opt to skip these cards when needed.

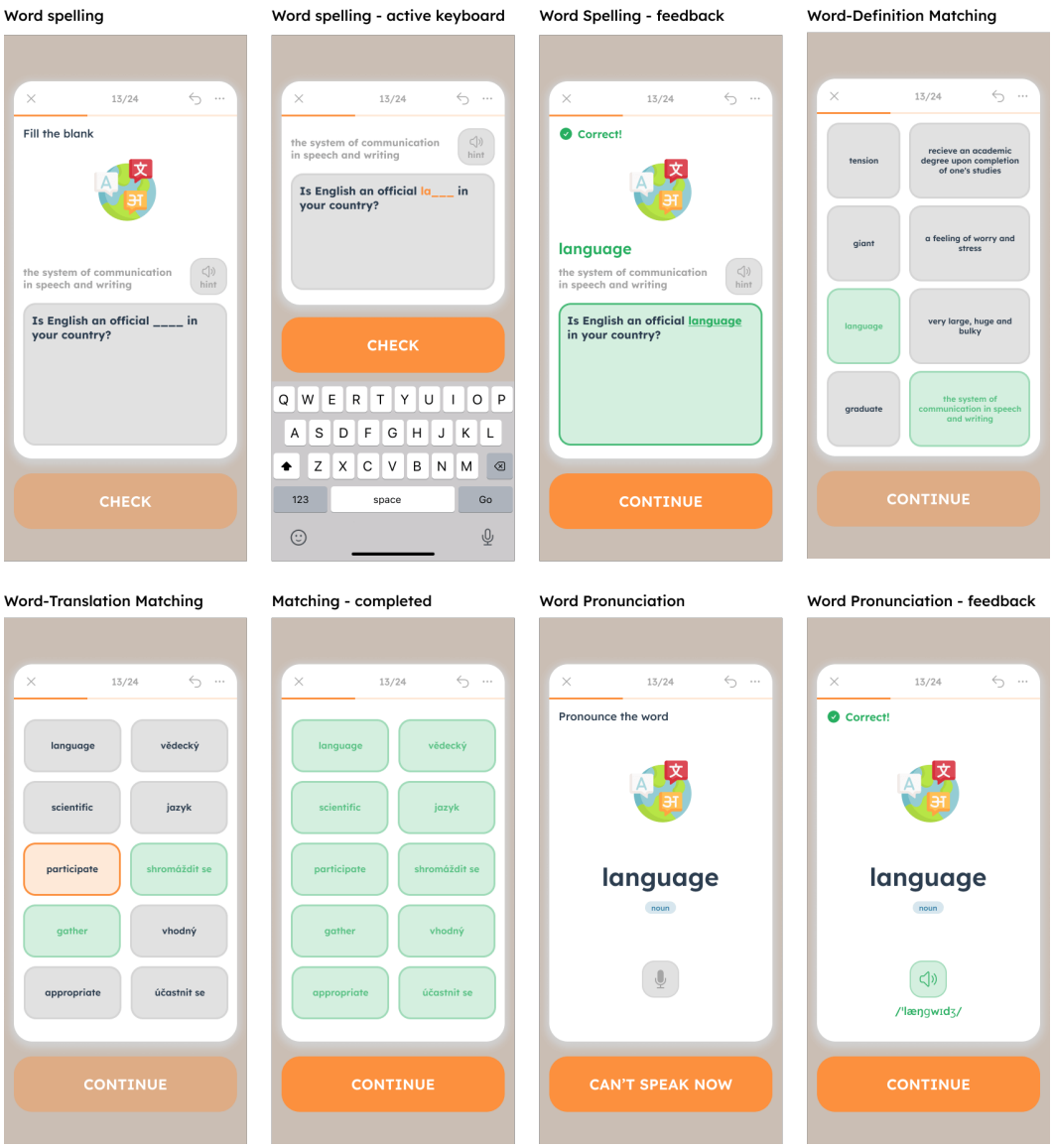


Figure 4.1: Prototype - Flashcard Types

The integration of new flashcard types requires careful consideration of the existing spaced repetition system. Only recall meaning flashcards will influence the SRS scheduling of words, while additional types (match, spell, and pronounce) serve as supplementary practice. This approach preserves the proven effectiveness of the current SRS algorithm while enhancing learning through varied practice formats.

The distribution of flashcard types in a practice session follows a structured approach. Recall meaning flashcards appear first and supplementary types follow in randomized order. This prioritization ensures that the core SRS algorithm remains the primary driver of vocabulary acquisition and prevents other flashcard types from inadvertently reminding users of word meanings before they attempt recall.

4.2 Vocabulary Progress Tracking

Inspired by WordUp's individual word progress tracking (see Section 3.2), we propose implementing a visual progress indicator system that integrates seamlessly with English Mind's existing spaced repetition system. This feature provides users with clear, immediate feedback on their progress with each vocabulary item, helping them understand where they stand in the learning journey for specific words. The visual indicator system maps the existing spaced repetition intervals into five distinct stages (see Table 4.1).

Table 4.1: Vocabulary Progress Tracking Stages

Stage	Name	SRS Interval
1	Starting	0 minutes - 3 days
2	Recognizing	3 days - 1 week
3	Reinforcing	1 week - 3 weeks
4	Strengthening	3 weeks - 3 months
5	Mastering	3 months - 8 months

The progress indicator appears only on the recall meaning flashcards (see Figure 4.2). It consists of five sequential boxes that fill progressively as the word advances through the stages, making the abstract concept of spaced repetition more concrete and understandable. The visual representation of progress creates a game-like element that enhances motivation and engagement.

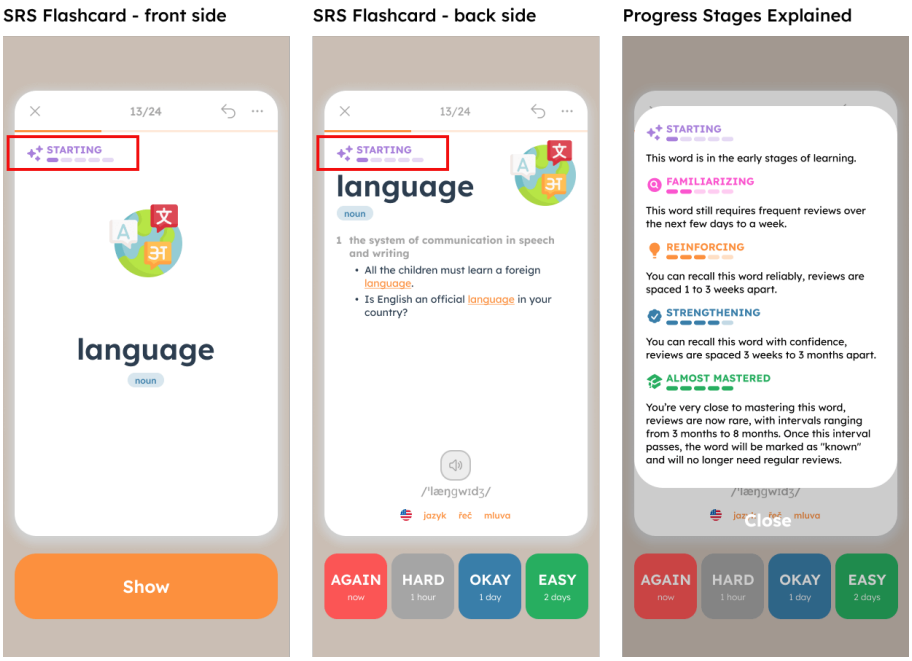


Figure 4.2: Prototype - Vocabulary Progress Tracker

4.3 Post-Practice Review

The post-practice review screen provides immediate feedback after completing a practice session (see Figure 4.3). It displays key metrics including the number of words practiced and time spent, accompanied by a brief motivational message. The implementation includes a lightweight confetti animation to provide positive reinforcement for session completion. This feature aims to increase user engagement and potentially increase practice session completion rates.

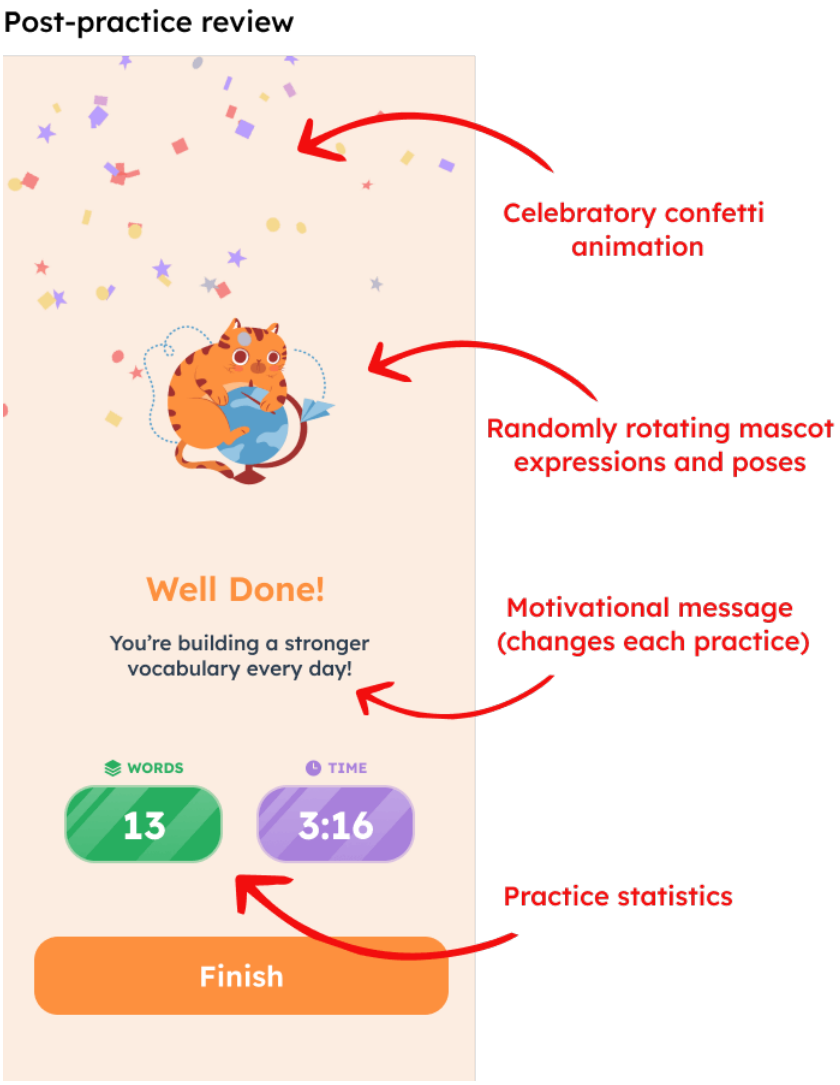


Figure 4.3: Prototype - Post-Practice Review

4.4 Streak System

The streak system draws inspiration from successful implementations in language learning applications like Duolingo, where 20% of daily active users maintain streaks exceeding one year [15]. The system implements a daily engagement mechanism that requires users to learn at least one new vocabulary item per day to maintain their active streak. This design choice balances the need for consistent progress with manageable daily commitments. It provides users with a clear, quantifiable metric for tracking their consistency. This approach facilitates the development and maintenance of regular practice habits.

The system deliberately omits advanced features such as streak freezes or recovery options, prioritizing the establishment and validation of core streak mechanics. This simplified approach allows for future feature expansion based on empirical user engagement data and feedback.

The system's visual implementation comprises two distinct components. The primary interface element is the streak counter (badge), prominently displayed at the top of the main screen (see Figure 4.4). This counter features a numerical representation of consecutive practice days, employing distinct visual states to indicate streak status: a golden appearance for active streaks and a faded state when the streak is at risk.

The second component focuses on achievement recognition. Upon successful completion of daily practice, a celebration screen appears displaying the updated streak count and a weekly progress visualization. The weekly view presents the current week's practice activity (Monday through Sunday), with perfect weeks highlighted in a special golden frame to recognize consistent engagement.

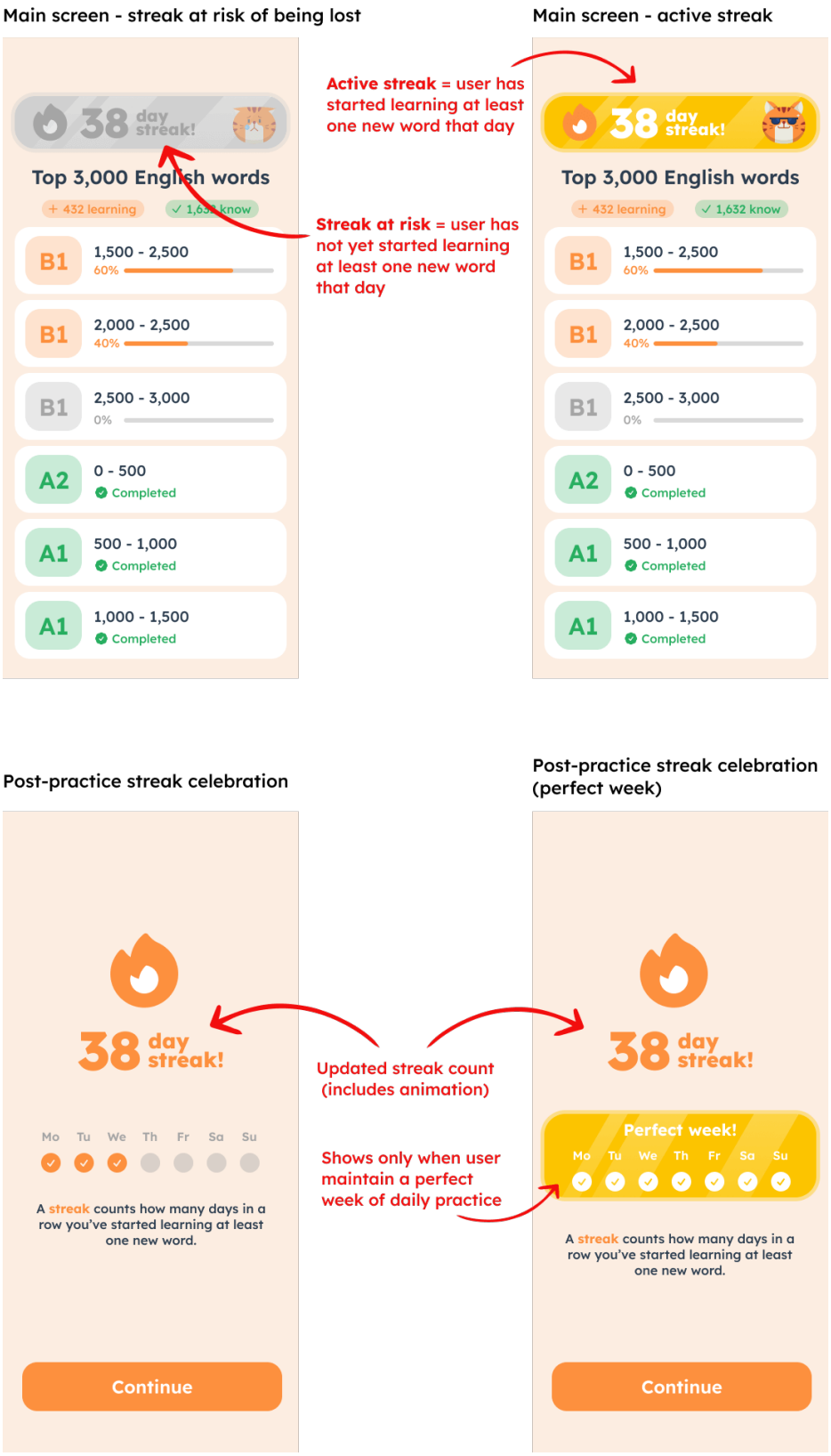


Figure 4.4: Prototype - Streak

Chapter 5

Prototype User Testing

This chapter describes the methodology and results of user testing conducted on the previously proposed and designed gamification features for English Mind. The designed features were implemented into an interactive prototype to enable hands-on user testing. The testing focused on evaluating the clarity, intuitiveness, and potential engagement effectiveness of the new gamification elements.

Two distinct user groups were selected for testing to provide diverse perspectives. Group A consisted of participants with prior experience using language learning applications, while Group B included participants with no experience in this area.

- **Group A – Experienced Users**
  - 3 participants
  - Regular users of apps like Duolingo, WordUp, or Duocards
  - Familiar with gamification concepts through regular usage of language learning applications (even without knowing the formal terminology)
- **Group B – Novice Users**
  - 3 participants
  - Interest in learning English vocabulary
  - No experience with language learning applications

The testing was conducted in a controlled environment where participants interacted with high-fidelity interactive prototypes on mobile devices according to the test scenario (see Section 5.1). During the prototype interaction, participants were encouraged to think aloud, providing real-time feedback on their experience, while observers documented behavioral observations and points of confusion. Afterwards, participants completed a structured questionnaire using a 5-point Likert scale (see Table A.1) to evaluate feature clarity, perceived usefulness, and engagement potential.

5.1 Test Scenario

The test scenario was structured as a series of specific tasks that participants were asked to perform, allowing for systematic evaluation of both gamification mechanics and their integration into the learning experience.

The first part of test scenario focused on the streak system understanding:

1. Participants were shown the streak status badge and asked to:
   - Interpret what the streak status badge represents
   - Explain how to activate a streak
   - Describe the requirements for maintaining a streak
   - Explain what actions would break their streak
   - Share their thoughts on how the streak system might motivate them

In the second part of test scenario participants were asked to complete a practice session with various flashcard types:

1. For each flashcard type, they were instructed to:
   - Read the instructions aloud
   - Explain how they would interact with the card
   - Complete the task while verbalizing their thought process
2. After completing several flashcards, participants were asked to:
   - Identify and explain the purpose of the element located in the top-right corner of the screen (without being told it was a progress tracker)
   - Describe stage system of the vocabulary progress tracker
   - Share how seeing their it affects their motivation
3. At the end of the practice session, participants were asked to:
   - Interpret their practice statistics
   - Explain how the session completion affects their streak
   - Share their reaction to the celebratory animations and motivational messages

After the test scenario participants were asked to fill out the questionnaire (see Table A.1).

5.2 Evaluation and Improvements

The user testing results, as detailed in Table 5.1, provide valuable insights into the effectiveness and areas for improvement of the gamification features in the proposed design. Overall, the feedback was positive, with participants mainly appreciating the variety and engagement offered by the different flashcard types. The overall average rating across all questionnaire items was 4.5 out of 5, indicating a generally favorable reception of the proposed features. However, the testing also identified several areas requiring improvement. Following iterative design adjustments and subsequent testing, these modifications successfully enhanced the user experience and resolved the initial concerns. Key improvements included:

1. The ”Can’t Speak Now” button’s visual prominence led users to misinterpret it as the primary action in pronunciation exercises. To solve this, the interface was redesigned to emphasize the microphone button as the primary call-to-action through visual hierarchy and positioning (see Figure 5.1).
2. Initial user interactions with the matching flashcard type revealed comprehension challenges. This was addressed by implementing a concise instruction ”Tap the matching pairs” to enhance task clarity.
3. Participants reported cognitive overload from the detailed word progress explanation. To address this, the explanation was simplified using visual representations and concise language.

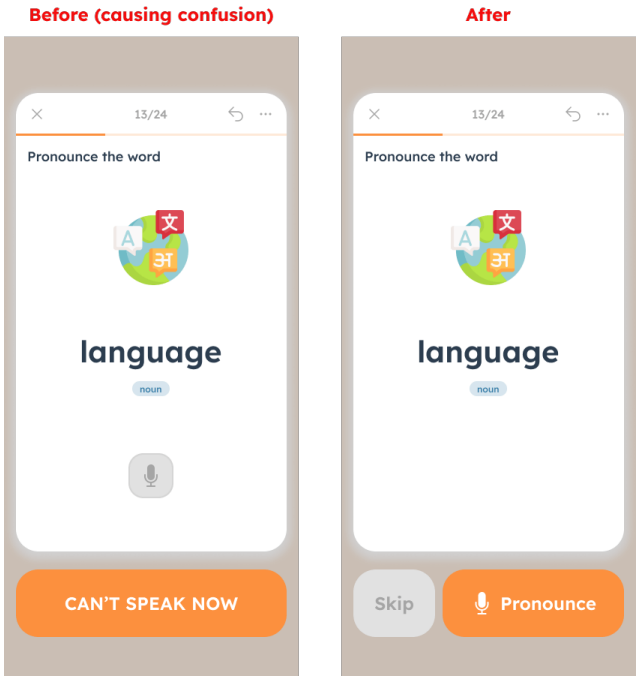


Figure 5.1: Prototype Testing - Pronounce Flashcard UI Changes

Table 5.1: Results of User Testing Questionnaire

A1, A2, A3 = Experienced Users; B1, B2, B3 = Novice Users; Avg = Average 1 = Strongly Disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, 5 = Strongly Agree							
Question	**A1**	**A2**	**A3**	**B1**	**B2**	**B3**	**Avg**
Practice Flashcards Experience							
1. The different types of flashcards made practice more engaging	5	5	5	5	5	5	5.0
2. Each flashcard type's instructions were clear and easy to understand	5	5	4	4	4	4	4.3
3. The variety in flashcard types helped me learn vocabulary more effectively	4	4	5	5	4	4	4.3
4. The distribution of different flashcard types felt well-balanced	4	5	4	5	5	5	4.7
5. The five-stage individual word progress indicator was easy to understand	5	5	3	4	4	3	4.0
6. Seeing my progress for individual words motivated me to practice more	4	3	3	4	3	3	3.3
7. The individual word progress indicator's placement was visually clear without being distracting	2	4	5	5	3	5	4.0
8. I found the progress tracking helpful for understanding my learning journey	5	4	5	5	5	4	4.7
9. The practice statistics provided useful information about my session	5	4	5	5	4	5	4.7
10. The celebratory animations made completing practice more rewarding	5	4	5	5	4	5	4.7
11. The review screen motivated me to complete future practice sessions	5	5	5	4	3	5	4.5
Streak System							
12. The streak system's requirements were clear and easy to understand	5	5	4	5	4	5	4.7
13. The visual states (active/at risk) effectively communicated streak status	5	5	3	5	4	4	4.3
14. The celebration screen for maintaining streaks felt rewarding	5	4	5	5	4	5	4.7
15. The requirement to learn one new word daily feels achievable	5	5	5	5	5	5	5.0
16. The streak system would help me build a consistent practice habit	5	4	5	4	4	5	4.5
17. I would be more likely to use the app regularly because of the streak feature	5	5	4	3	4	4	4.2
Overall Experience							
18. The gamification features enhanced my learning experience	5	4	5	4	5	5	4.7
19. The features felt well-integrated with the app's educational purpose	4	5	5	5	5	5	4.8
20. The gamification elements maintained my interest without being distracting	4	5	5	5	5	5	4.8
21. I would recommend this app to others learning English vocabulary	5	5	5	5	5	5	5.0
22. I would continue using this app for long-term vocabulary learning	5	4	5	4	4	5	4.5
Overall Average	**4.6**	**4.5**	**4.5**	**4.6**	**4.2**	**4.6**	**4.5**

Chapter 6

Implementation

This chapter presents the technical implementation of the gamification features proposed in Chapter 4 and validated through user testing in Chapter 5. The implementation process was structured to ensure both technical feasibility and alignment with the application's existing architecture while maintaining the core pedagogical principles of English Mind.

This chapter begins with an analysis of the current system architecture, providing context for the technical decisions made during implementation. Following this, a monorepo architecture is introduced that enables parallel development of both the main application and a demonstration application, allowing for the protection of the main application's intellectual property while still enabling development and testing of new features. This approach allows for rapid iteration and testing of new features without disrupting the production environment. Finally, the implementation of selected gamification features is detailed, focusing on technical considerations, integration challenges, and solutions adopted.

6.1 Existing System Architecture

English Mind is built using the Flutter framework [16], chosen for its cross-platform capabilities and robust widget system. The application follows a modular architecture inspired by Clean Architecture principles [17], ensuring separation of concerns and maintainability. This section provides a comprehensive overview of the current system architecture, detailing the technology stack, architectural patterns, and key system components.

6.1.1 Technology Stack

The application is built using the Flutter framework [16], chosen for its cross-platform capabilities and robust widget system. This choice enables the development of a single codebase that runs seamlessly on both iOS and Android platforms while maintaining native-like performance and user experience. The decision to use Flutter was driven by its ability to deliver consistent user interfaces across platforms while maintaining high performance, as demonstrated by its successful implementation in numerous production applications [18].

The data management strategy combines cloud-based and local storage solutions to balance performance, offline capability, and real-time synchronization requirements. Firebase Firestore [19] serves as the primary cloud storage solution, providing real-time data synchronization and a NoSQL document database. This choice was made to leverage Firestore's built-in offline persistence and real-time updates, which are crucial for maintaining user progress across devices. For translation services, the application integrates with the Microsoft Translator API [20], selected for its high accuracy and extensive language support. The vocabulary data is stored locally in a JSON file containing word definitions and sentence examples, a design decision that ensures immediate access to core content without network dependency and reduces server load.

The application leverages several key Flutter/Dart packages to enhance functionality and maintainability:

- **get_it** - Dependency injection framework for managing object creation and lifetime.
- **flutter_hooks** - State management solution chosen for its simplicity and performance benefits over traditional state management approaches.
- **auto_route** - Navigation and routing system that provides type-safe routing and reduces boilerplate code.
- **objectbox** - Local storage solution for efficient data management.
- **sign_in_with_apple** and **google_sign_in** - Third-party authentication providers.
- **firebase_auth** - User authentication service providing secure user management.
- **firebase_functions** - Cloud functions for serverless computing, enabling scalable backend operations.
- **firebase_firestore** - Cloud storage for real-time data synchronization and offline persistence.
- **flutter_localizations** - Localization framework enabling internationalization support.

6.1.2 Architectural Overview

The application follows Clean Architecture [17] principles with a clear separation of concerns across distinct layers, while incorporating custom architectural decisions to better suit the application's specific needs. The architecture is designed to maintain high maintainability and scalability while ensuring a unidirectional data flow. The codebase is organized into the following directory structure, each serving a specific purpose in the application's architecture:

- `core/` – Provides shared utilities and common functionality.
  - `constants/` – Application-wide constants and shared values.
  - `extensions/` – Dart language extensions providing additional functionality.
  - `injector/` – Dependency injection configuration and service registration.
  - `l10n/` – Internationalization and localization support.
  - `logger/` – Application logging functionality and utilities.
  - `routes/` – Application navigation and routing configuration.
  - `utils/` – Utility functions and helpers.
- `domain/` – Contains the core business logic and is implementation independent.
  - `entities/` – Core business objects representing the domain model.
  - `managers/` – Business logic orchestrators and state managers.
  - `repositories/` – Abstract interfaces defining data operations.
  - `usecases/` – Core business operations and rules between entities and repositories.
- `data/` – Handles data operations and implementations of repository interfaces.
  - `models/` – Data models and transformations.
  - `repositories/` – Concrete implementations of repository interfaces.
  - `mappers/` – Transformation logic between domain entities and data models.
- `ui/` – Presentation layer managing user interface components and interactions.
  - `theme/` – UI theme configuration and styling.
  - `screens/` – Screen-specific UI components and layouts.
  - `widgets/` – Reusable UI components.
  - `hooks/` – Hooks for state management and resource handling within Flutter widgets.

6.1.3 Architecture Limitations

While the current architecture provides a solid foundation for the core English Mind application, it presents several challenges when implementing new gamification features. A critical consideration is the proprietary nature of the English Mind application, which necessitates a separate demonstration application (`em_demo`) for implementing gamification features. This approach preserves intellectual property rights while providing a controlled environment for feature validation. The demonstration application serves as a proof of concept, enabling the implementation and testing of gamification elements without exposing the original application's proprietary codebase. This separation allows for independent development and validation of new features while maintaining the integrity of the production application.

The architectural challenges manifest in several key areas that impact the development of the demonstration application. First, while the current architecture follows clean architecture principles, the UI components exhibit tight coupling with application-specific concerns. This coupling creates significant challenges when attempting to reuse components in the new application.

A primary example of this coupling is the integration of multiple concerns within individual components. Specifically, UI components combine both localization logic through `flutter_localizations` and state management through `flutter_hooks`. This integration violates the single responsibility principle [21], as each component handles multiple distinct responsibilities. The resulting tight coupling makes it challenging to extract and reuse components in the new application without also migrating their dependencies, leading to potential code duplication and increased maintenance overhead.

Additionally, the lack of systematic component documentation presents a significant barrier to component reuse. Without clear documentation of component interfaces, dependencies, and usage patterns, developers face difficulties in understanding and adapting existing UI elements for the demonstration application. This issue is particularly problematic when attempting to maintain consistency between the original and demonstration applications.

These limitations collectively highlight the need for a more modular architecture that supports component isolation and documentation. Such an architecture would enable better component reuse while maintaining consistency between the original and demonstration applications.

6.2 New System Architecture

The architectural limitations identified in the previous section 6.1.3 necessitated a fundamental restructuring of the development approach. The primary challenge was the need to maintain a separate demonstration application (`em_demo`) for implementing and testing gamification features while preserving the proprietary nature of the main English Mind application. This requirement, combined with the existing architectural constraints of component coupling and documentation gaps, led to the adoption of a monorepo architecture. This new architecture provides a structured environment that enables parallel development of both applications while maintaining clear separation of concerns. The following sections detail how this architectural shift not only addresses the immediate need for separate applications but also improves overall maintainability and scalability of the development process.

6.2.1 Monorepo Solution

To address the architectural limitations identified in Section 6.1.3, we implemented a monorepo architecture that enables parallel development of both the main English Mind application and the demonstration application. This approach provides several key benefits:

- **Code Reusability** - Shared components and utilities can be extracted into separate packages, reducing code duplication and ensuring consistency across applications.
- **Unified Version Control** - All code resides in a single repository, providing a complete history of changes across all applications and packages, while maintaining clear separation between the main and demonstration applications.
- **Streamlined Development Workflow** - Developers can work on multiple applications simultaneously while maintaining a single source of truth for shared code.
- **Improved Testing and Validation** - New features can be developed and tested in the demonstration application before being integrated into the main application.

The monorepo solution was implemented using Melos [22], a tool specifically designed for managing Dart and Flutter monorepos. Melos provides essential features such as workspace management, package linking, and script execution across multiple packages. This choice was made to leverage Melos's robust support for Flutter development and its ability to handle complex dependency graphs while maintaining optimal build performance.

6.2.2 Monorepo Architecture

The monorepo architecture is organized into a clear directory structure that promotes code organization, reusability, and maintainability. The root directory contains the following key components:

- `apps/` – Contains all Flutter applications in the workspace.
  - `english_mind/` – The production version of English Mind.
  - `em_demo/` – The demonstration version of English Mind.
  - `widgetbook/` – UI components documentation.
- `packages/` – Houses all shared packages used across applications.
  - `em_theme/` – Shared UI theming and styling components.
  - `em_lints/` – Custom linting rules and analysis options.
  - `em_widgets/` – Reusable UI components.
  - `em_vocabulary/` – Core vocabulary data, models and management.
- `melos.yaml` – Melos configuration scripts.

The `em_demo` application follows the same architectural structure as the main English Mind application as described in Section 6.1.2, maintaining consistency in code organization and development patterns. This approach ensures that features developed in the demonstration application can be seamlessly integrated into the main application when ready for production deployment.

6.2.3 Widgetbook Documentation

Widgetbook [23] serves as a dedicated documentation environment for UI components, providing an interactive catalog of all shared widgets and their variations, as shown in Figure B.1. This tool enables developers to visualize and test components in isolation, ensuring consistent implementation across both applications.

The adoption of Widgetbook was driven by its ability to provide centralized, interactive documentation of UI components, making it easier for developers to understand and use shared components correctly. The tool significantly improves development efficiency by allowing developers to work on components in isolation without needing to navigate through the full application, reducing context switching and development time [24]. Furthermore, it serves as a living style guide that validates the implementation of the design system and ensures adherence to design specifications, while also facilitating better communication between designers and developers through a shared reference point for UI components [25].

6.3 Features Implementation

All gamification features proposed in Chapter 4 were successfully implemented in the demonstration application, as shown in Appendix B. The implementation followed the architectural principles established in Section 6.2, ensuring maintainability and scalability. While all features were implemented, this section focuses on the two most significant additions: the diversified flashcard system and the streak system. Detailed implementation specifics can be found in the source code.

6.3.1 Diversified Flashcard Types

The implementation of diversified flashcard types showcases a clean, type-safe approach using Dart's sealed classes. This design pattern ensures exhaustive handling of all flashcard types while maintaining a clear separation of concerns. The core of this implementation is the `FlashcardData` sealed class hierarchy:

```dart
sealed class FlashcardData {}

class RecallFlashcardData extends FlashcardData {
  final VocabularyEntry entry;
  final FlashcardSrsMetadata metadata;

  RecallFlashcardData({required this.entry, required this.metadata});
}

class SpellingFlashcardData extends FlashcardData {
  final VocabularyEntry entry;

  SpellingFlashcardData(this.entry);
}

class PronunciationFlashcardData extends FlashcardData {
  final VocabularyEntry entry;

  PronunciationFlashcardData(this.entry);
}

class MatchDefinitionsFlashcardData extends FlashcardData {
  final List<VocabularyEntry> entries;

  MatchDefinitionsFlashcardData(this.entries);
}

class MatchTranslationsFlashcardData extends FlashcardData {
  final List<VocabularyEntry> entries;

  MatchTranslationsFlashcardData(this.entries);
}
```

Code 6.1: FlashcardData sealed class hierarchy defining different flashcard types

This sealed class hierarchy provides several key benefits. The use of sealed classes ensures type safety through exhaustive pattern matching, which guarantees that all possible cases are handled at compile time. This design pattern also promotes a clear separation of concerns between different flashcard types, making the code more maintainable and easier to understand. Furthermore, the architecture is designed with extensibility in mind, allowing for straightforward addition of new flashcard types in the future while maintaining the existing type safety guarantees.

The type-safe design significantly simplifies flashcard queue creation and management. Since each flashcard type is explicitly defined in the sealed class hierarchy, the system can easily generate and manage queues containing any combination of flashcard types. The compiler ensures proper handling of all flashcard types, preventing accidental omissions during queue creation and processing. This design also enables efficient queue operations like filtering, shuffling, and type-specific transformations while maintaining type safety.

The UI implementation leverages this type-safe design through Dart's pattern matching capabilities. The `StudyScreen` widget demonstrates how this architecture enables clean, maintainable UI code:

```
@RoutePage()
class StudyScreen extends HookWidget {
  @override
  Widget build(BuildContext context) {
    // ... state management code ...

    return Scaffold(
      body: switch (flashcardData) {
        final SpellingFlashcardData data =>
          SpellingFlashcard(entry: data.entry),
        final MatchDefinitionsFlashcardData data =>
          MatchDefinitionsFlashcard(entries: data.entries),
        final MatchTranslationsFlashcardData data =>
          MatchTranslationsFlashcard(entries: data.entries),
        final PronunciationFlashcardData data =>
          PronunciationFlashcard(vocabularyEntry: data.entry),
        final RecallFlashcardData data => RecallFlashcard(
            vocabularyEntry: data.entry,
            flashcardSrsMetadata: data.metadata,
          ),
      },
    );
  }
}
```

Code 6.2: StudyScreen widget with pattern matching

This implementation demonstrates several key architectural principles. Pattern matching ensures comprehensive handling of all flashcard types, while dedicated widgets for each type promote the single responsibility principle. The sealed class hierarchy enforces a strict contract where new flashcard types require corresponding UI updates, preventing inconsistencies. The architecture also maintains flexibility for future expansion, allowing straightforward addition of new flashcard types.

6.3.2 Streak System

The streak system implementation focused on creating a reliable and engaging mechanism for tracking daily practice, with a design that prioritizes scalability and future extensibility. The core of the streak system is implemented in the `Streak` class:

```
1  class Streak {
2    final String _uid;
3    final Set<int> _activeDays; // Unix timestamps
4    // final Set<int> freezeDays; // Unix timestamps
5
6    bool get isActive;
7    int get currentStreak;
8    int get longestStreak;
9
10   Streak({required String uid, Set<int>? activeDays})
11       : _uid = uid,
12         _activeDays = activeDays ?? <int>{};
13 }
```

Code 6.3: Streak class implementation

The streak system's implementation provides a solid foundation for tracking user engagement while maintaining flexibility for future enhancements. The design decisions ensure that the system remains reliable, accurate, and scalable as the application evolves. The implementation demonstrates several key design decisions:

- **Timezone Independence** - By using Unix timestamps (milliseconds since epoch) and converting to local dates only when necessary, the system maintains consistency across different timezones. This approach ensures that streak calculations remain accurate regardless of the user's location or timezone changes.
- **Historical Data Preservation** - The system maintains a complete history of active days, enabling not only current streak tracking but also historical analysis. This data structure allows for easy calculation of both current and longest streaks, providing valuable insights into user engagement patterns its analysis.
- **Data Integrity** - Streak values are calculated on-demand rather than stored, ensuring that the data remains consistent even if the system rules change. This approach prevents data corruption and allows for flexible updates to streak calculation logic without requiring data migration.
- **Future Extensibility** - The commented `freezeDays` field (line 4 in Code 6.3) demonstrates how the system can be extended to support additional features like streak freezes. This design pattern allows for straightforward addition of new functionality without disrupting existing streak calculations.

Chapter 7

Testing

This chapter presents the testing approach and results for the implemented gamification features. The testing strategy combined both automated unit testing and user testing to ensure the reliability and usability of the new features.

7.1 Unit Testing

Given that most of the implementation features are UI components, which are not suitable for unit testing due to their interactive and visual nature, the testing efforts were primarily focused on the streak system's logic. The streak unit tests were implemented using the Flutter testing framework and covered the core functionality of streak tracking, including:

- Streak activation and maintenance
- Streak calculation across different time zones
- Streak break conditions
- Historical streak tracking

These tests ensure the reliability of the streak tracking mechanism, which is crucial for maintaining user engagement and motivation.

7.2 User Testing

The user testing was conducted over a 14-day period with six participants, evenly distributed across Android and iOS platforms (three participants each). This extended testing period allowed for thorough evaluation of the streak system's reliability and user engagement patterns. The testing focused on evaluating the usability and effectiveness of the gamification features, particularly:

- Flashcard type interactions and usability
- Streak system comprehension and engagement
- Progress tracking clarity

The testing revealed only one minor issue: the pronunciation recognition system occasionally failed to recognize isolated words. However, when users pronounced the words within a sentence context, the recognition worked correctly. This finding suggests that the speech recognition system performs better with contextual pronunciation, which aligns with natural language patterns.

All other features functioned as intended, with no additional bugs or issues reported. The positive testing outcomes validate the technical implementation and suggest that the features are ready for production deployment.

7.2.1 Limitations and Future Work

While the 14-day testing period provided valuable insights into the immediate usability and technical functionality of the gamification features, it was not sufficient to fully evaluate long-term engagement patterns and learning outcomes. Future work could include:

- Extended long-term study (3-6 months) to analyze:
  - Long-term engagement patterns and retention rates
  - Correlation between streak maintenance and vocabulary acquisition
- A/B testing of different gamification mechanics to optimize engagement
- Integration of learning analytics to measure the effectiveness of gamification on vocabulary retention
- Comparative study with a control group using a non-gamified version

These extended studies would provide more robust data on the long-term effectiveness of the implemented gamification features and their impact on language learning outcomes.

Chapter 8

Conclusion

This thesis has explored the integration of gamification elements into English Mind, a language learning application, with the aim of enhancing user engagement and learning outcomes. Through careful analysis of existing methodologies and competitor applications, key opportunities for improvement were identified and a set of carefully designed gamification features that are aligned with the application's core learning methodologies were implemented.

The success of this work is evaluated against the three principal dimensions defined in the Introduction 1.1: competitive positioning, user feedback, and implementation success.

1. **Enhancement of Competitive Positioning**
   a. The project successfully analyzed English Mind's core methodologies, including frequency-based vocabulary acquisition and spaced repetition, to understand the key principles of the application, see Chapter 2.
   b. A comprehensive analysis of three similar language learning applications was conducted, identifying several opportunities to improve the user experience and competitive positioning, see Chapter 3.
   c. Four key gamification features and enhancements were identified: diversified flashcard types, vocabulary progress tracking, post-practice analytics, and a streak system, all carefully selected to align with the application's core methodologies while addressing the competitive gap, see Chapter 4.

      The comparison of gamification features in Table 8.1 shows how these enhancements filled the identified gaps in English Mind's feature set compared to its competitors.

Table 8.1: Enhanced Competitive Positioning in Gamification Features (blue check mark "✓" indicates a newly added feature)

Gamification	English Mind	WordUp	DuoCards	Duolingo
Flashcard Types				
Exercise variety	✓	✓	✓	✓
Recall meaning	✓	✓	✓	✓
Speech recognition	✓	—	—	✓
Spelling assessment	✓	✓	✓	✓
Matching definitions	✓	✓	—	—
Matching translations	✓	—	✓	✓
Progress Tracking				
Aggregate progress metrics	✓	✓	✓	✓
Practice session progress	✓	—	—	✓
Word-level analytics	✓	✓	—	—
Post-practice analytics	✓	—	—	✓
Engagement Features				
Streak system	✓	—	✓	✓
Time-based goals	—	✓	—	—
Social comparison	—	✓	—	✓

2. **Positive User Feedback on Gamification Features**

   a. The four key gamification features were designed and carefully crafted to be on par with or surpass those of leading competitors while maintaining English Mind's unique value proposition, see Chapter 4.

   b. A prototype user testing was conducted with six participants, who provided detailed feedback through post-test questionnaires. The results showed strong user satisfaction with an average rating of 4.5/5.0 (see Table 5.1), confirming the value of the proposed features while highlighting areas for improvement, see Chapter 5.

3. **Successful Implementation of Gamification Features**

   a. The current system architecture was thoroughly analyzed, revealing limitations in component coupling and documentation, see Chapter 6.1. A new monorepo solution using Melos was proposed and implemented, enabling parallel development of the main application and the demonstration prototype while preserving intellectual property, see Chapter 6.2. All proposed gamification features were successfully implemented in the demonstration application, showcasing the effectiveness of the new architecture, see Chapter 6.3. The final implemented UI is shown in Appendix B.

b. The implementation was validated through comprehensive end-user testing, which confirmed the robustness and usability of the new features. The testing process verified the successful integration of the gamification elements and their alignment with the application's core learning methodologies, see Chapter 7.

In conclusion, this thesis has demonstrated the successful integration of gamification elements into English Mind, significantly enhancing its competitive position in the language learning application market. The implemented features not only address the identified gaps in the application's feature set but also maintain the core learning methodologies that make English Mind unique. The positive user feedback and successful implementation validate the effectiveness of the proposed solution, suggesting that gamification can indeed serve as a powerful tool for enhancing user engagement and learning outcomes in language learning applications.

Looking ahead, several promising directions for future work emerge. First, the gamification features could be expanded to include social elements, such as collaborative learning challenges or competitive leaderboards, which could further enhance user motivation and engagement. Second, a longitudinal study could be conducted to assess the long-term impact of these gamification features on vocabulary retention and learning outcomes. These future developments would not only enhance the application's capabilities but also contribute to the broader understanding of gamification's role in language learning.

Appendix A

Prototype User Testing

Table A.1: User Testing Questionnaire

Rating scale: 1 = Strongly Disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, 5 = Strongly Agree	
Question	**Rating**
Practice Flashcards Experience	
1. The different types of flashcards made practice more engaging	
2. Each flashcard type's instructions were clear and easy to understand	
3. The variety in flashcard types helped me learn vocabulary more effectively	
4. The distribution of different flashcard types felt well-balanced	
5. The five-stage individual word progress indicator was easy to understand	
6. Seeing my progress for individual words motivated me to practice more	
7. The individual word progress indicator's placement was visually clear without being distracting	
8. I found the progress tracking helpful for understanding my learning journey	
9. The practice statistics provided useful information about my session	
10. The celebratory animations made completing practice more rewarding	
11. The review screen motivated me to complete future practice sessions	
Streak System	
12. The streak system's requirements were clear and easy to understand	
13. The visual states (active/at risk) effectively communicated streak status	
14. The celebration screen for maintaining streaks felt rewarding	
15. The requirement to learn one new word daily feels achievable	
16. The streak system would help me build a consistent practice habit	
17. I would be more likely to use the app regularly because of the streak feature	
Overall Experience	
18. The gamification features enhanced my learning experience	
19. The features felt well-integrated with the app's educational purpose	
20. The gamification elements maintained my interest without being distracting	
21. I would recommend this app to others learning English vocabulary	
22. I would continue using this app for long-term vocabulary learning	

Appendix B

Implementation

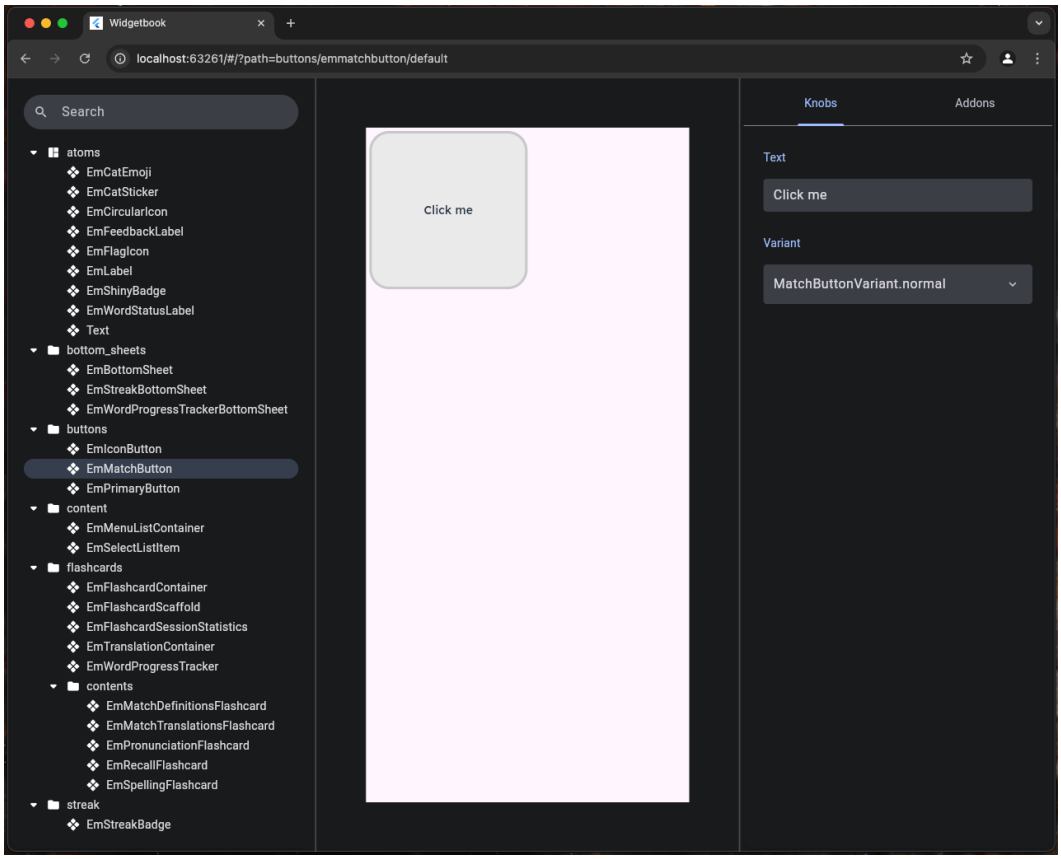


Figure B.1: Widgetbook - UI Component Documentation

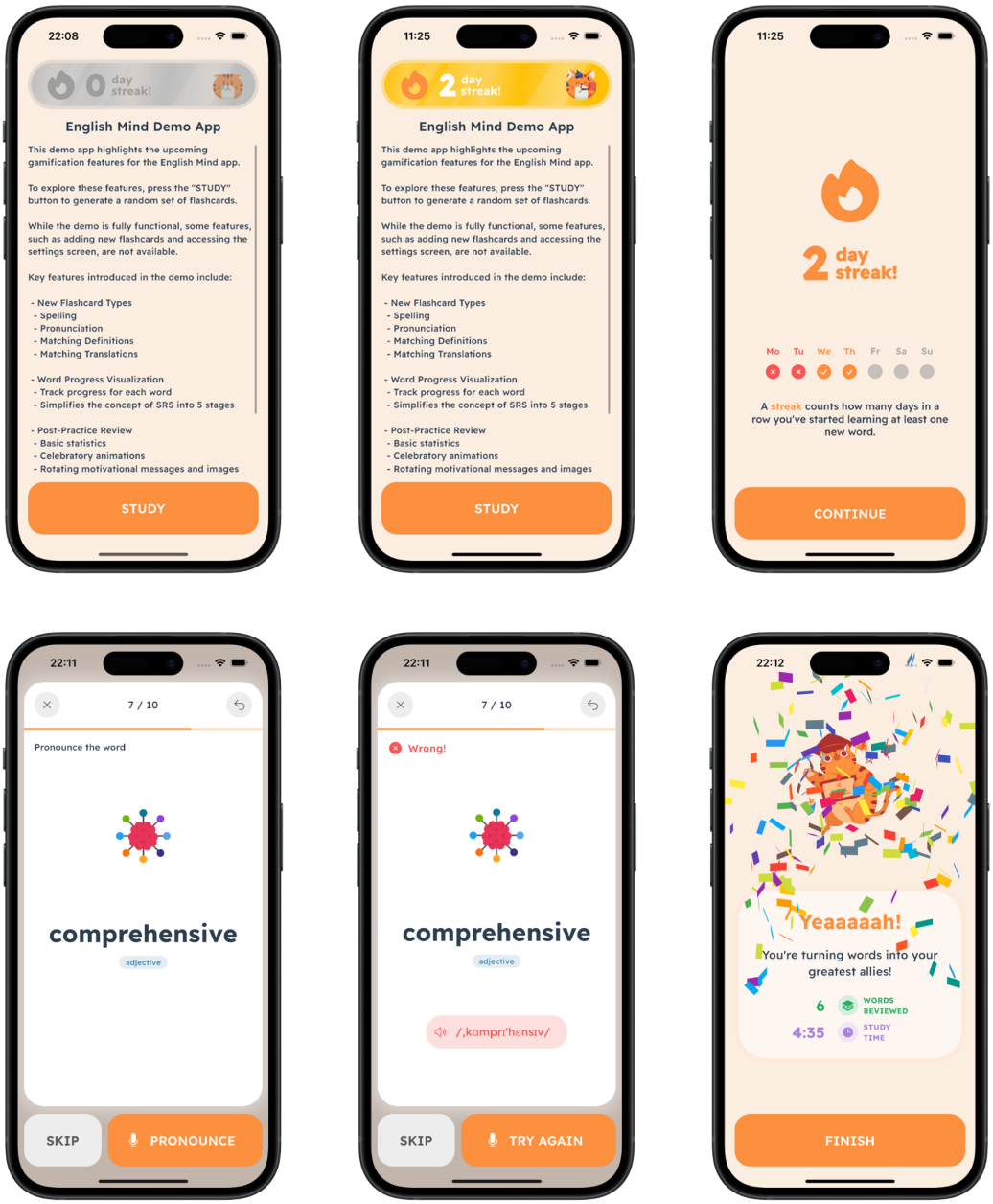


Figure B.2: Implemented UI - Streak, Pronounce Flashcard, Post-practice Analytics

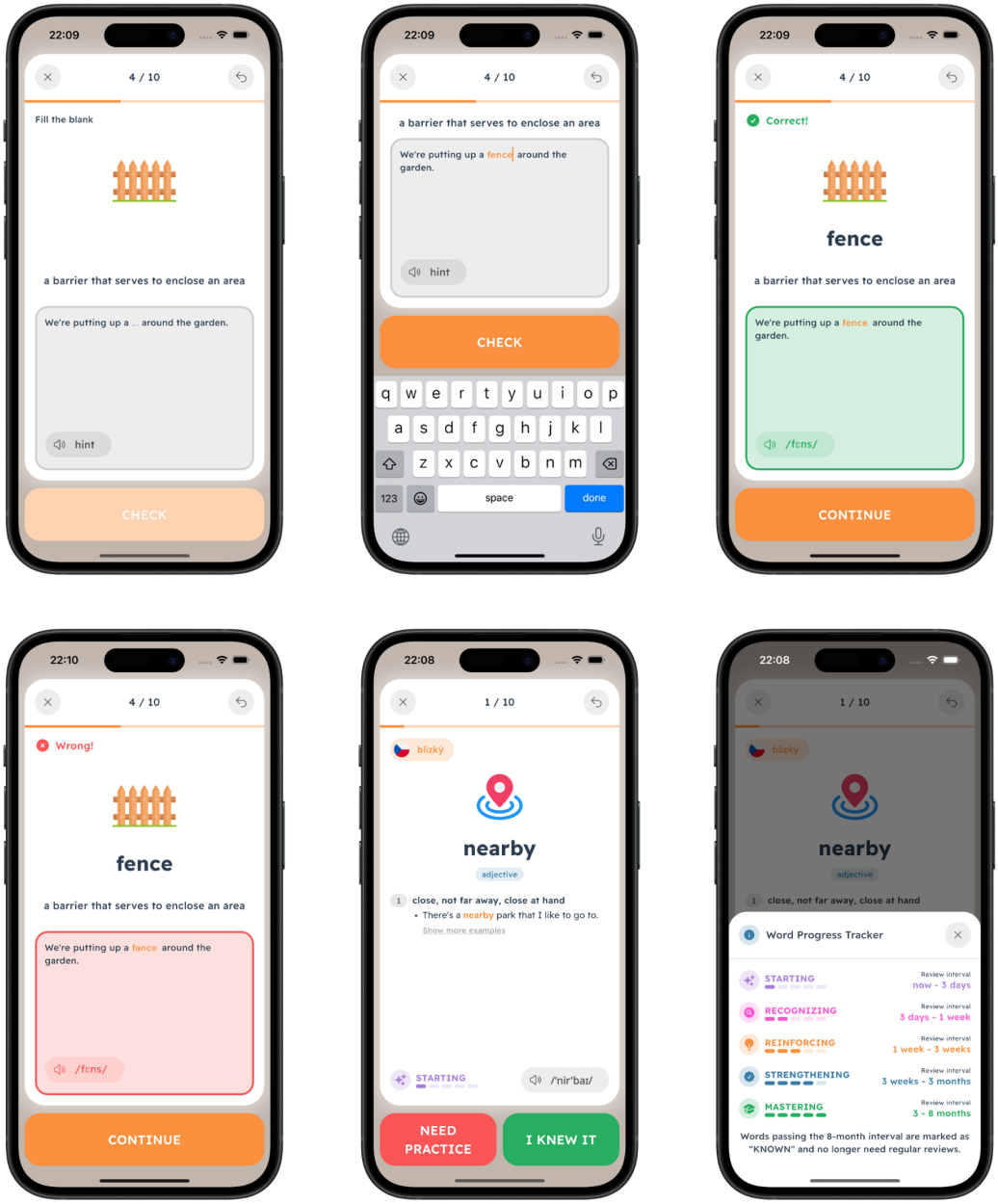


Figure B.3: Implemented UI - Spelling Flashcard, Word Progress Tracker

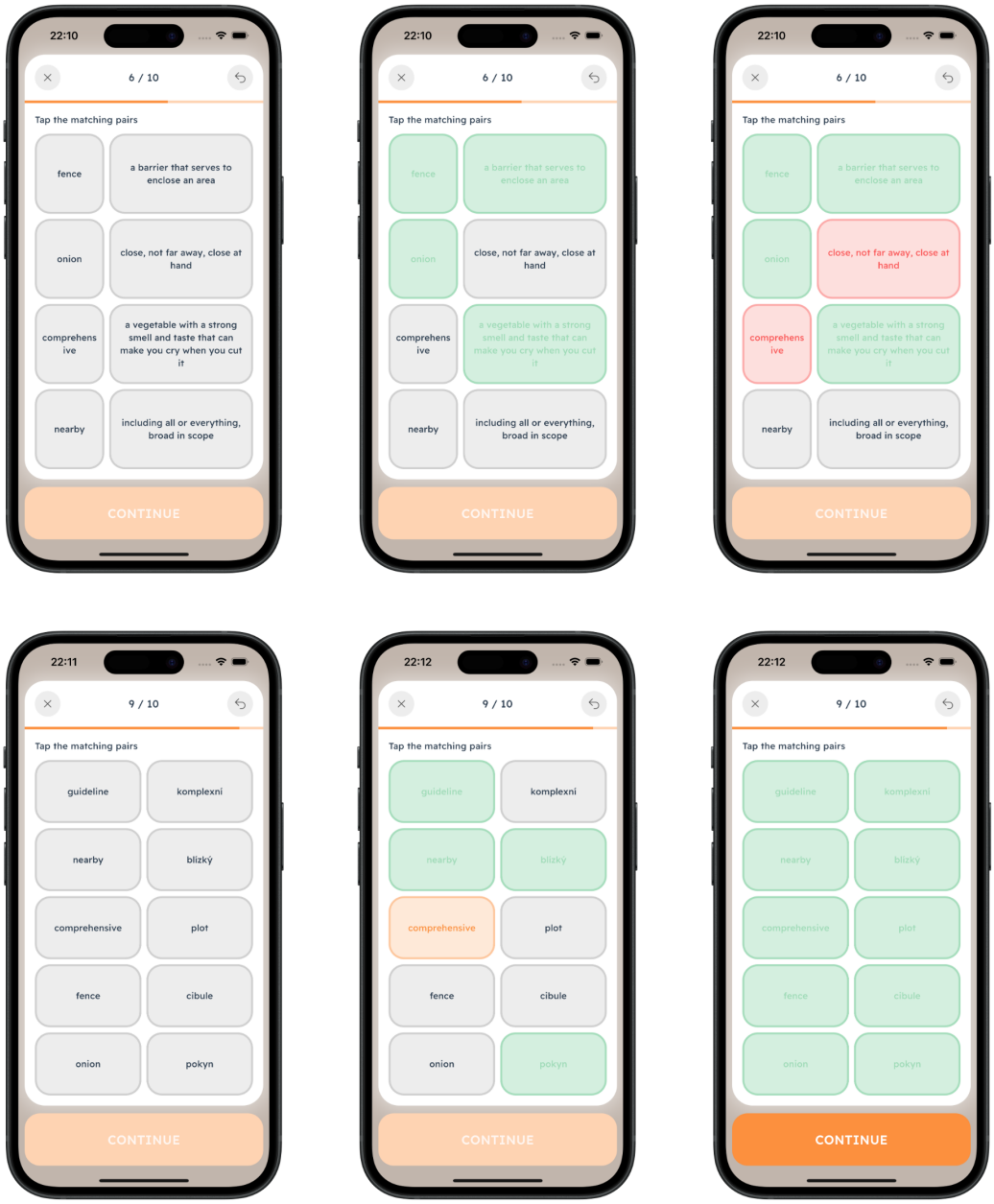


Figure B.4: Implemented UI - Match Definitions Flashcards, Match Translations Flashcard

Appendix C

Bibliography

[1] Henry L Roediger III and Jeffrey D Karpicke. Test-enhanced learning: Taking memory tests improves long-term retention. *Psychological science*, 17(3):249–255, 2006.

[2] Terence Govender and Joan Arnedo-Moreno. A survey on gamification elements in mobile language-learning applications. In *Eighth International Conference on Technological Ecosystems for Enhancing Multiculturality*, TEEM'20, page 669–676, New York, NY, USA, 2021. Association for Computing Machinery.

[3] English Mind. `https://www.englishmind.app`, 2024. Accessed: 2024-10-20.

[4] Duolingo. `https://www.duolingo.com/`, 2024. Accessed: 2024-10-24.

[5] WordUp. `https://www.wordupapp.co/`, 2024. Accessed: 2024-10-24.

[6] DuoCards. `https://duocards.com/cs/`, 2024. Accessed: 2024-10-24.

[7] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. From game design elements to gamefulness: Defining gamification. volume 11, pages 9–15, 09 2011.

[8] English Mind s.r.o. English mind on google play. `https://play.google.com/store/apps/details?id=app.englishmind.english_mind`. Accessed: 2024-10-20.

[9] English Mind s.r.o. English mind on app store. `https://apps.apple.com/cz/app/english-mind/id6478237004`. Accessed: 2024-10-20.

[10] I Nation. How large a vocabulary is needed for reading and listening? *Canadian modern language review*, 63(1):59–82, 2006.

[11] Hermann Ebbinghaus. Memory: A contribution to experimental psychology. *Annals of neurosciences*, 20(4):155, 2013.

[12] Sean HK Kang. Spaced repetition promotes efficient and effective learning: Policy implications for instruction. *Policy Insights from the Behavioral and Brain Sciences*, 3(1):12–19, 2016.

[13] Google Play. Wordup | ai vocabulary builder. `https://play.google.com/store/apps/details?id=co.wordupapp.app`, 2024. Accessed: 2024-10-24.

[14] Google Play. Duocards - language flashcards. `https://play.google.com/store/apps/details?id=com.duocards.app`, 2024. Accessed: 2024-10-28.

[15] Duolingo Inc. Duolingo hits 100m maus, reports 59% dau growth and 41% revenue growth in second quarter 2024. `https://investors.duolingo.com/node/9966/pdf`, August 2024. Press Release.

[16] Google. Flutter - build apps for any screen. `https://flutter.dev/`, 2025. Accessed: 2025-05-10.

[17] Robert C. Martin. *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Pearson, 2017.

[18] Google. Flutter case studies. `https://flutter.dev/showcase`, 2025. Accessed: 2025-05-10.

[19] Google. Cloud firestore documentation. `https://firebase.google.com/docs/firestore`, 2025. Accessed: 2025-05-10.

[20] Microsoft. Microsoft translator documentation. `https://learn.microsoft.com/en-us/azure/ai-services/translator/`, 2025. Accessed: 2025-05-10.

[21] Robert C. Martin. Solid principles. `https://web.archive.org/web/20200916100829/http://www.butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod`, 2000. Accessed: 2025-05-10.

[22] Invertase. Melos - a tool for managing dart and flutter projects with multiple packages. `https://melos.invertase.dev/`, 2025. Accessed: 2025-05-10.

[23] Widgetbook GmbH. Widgetbook - the flutter storybook. `https://widgetbook.io/`, 2025. Accessed: 2025-05-10.

[24] George T. Heineman and William T. Councill. Component-based software engineering: A survey of research issues. *Component-Based Software Engineering: Putting the Pieces Together*, pages 1–24, 2001.

[25] Richard Buchanan and Victor Margolin. A systematic review of design system research. *Design Issues*, 11(2):5–20, 1995.