

University of Tartu
Faculty of Social Sciences
Narva College
Study Program “Information Technology Systems Development”

Sergei Markov

**DEVELOPMENT OF AN APPLICATION
FOR QUICK CREATION OF ANKI CARD COLLECTION
FOR ESTONIAN LEARNING**

Final Thesis

Supervisor: Dr. Chen-Wan Lin

Narva 2026

LICENCE TO REPRODUCE THE THESIS AND MAKE IT PUBLICLY AVAILABLE

I, SERGEI MARKOV,

1. grant to the University of Tartu a free licence (non-exclusive licence) to reproduce my work
DEVELOPMENT OF AN APPLICATION FOR QUICK CREATION OF ANKI CARD
COLLECTION FOR ESTONIAN LEARNING,

supervised by

CHEN-WAN LIN,

for the purpose of preservation, including inclusion in the digital archive DSpace until the
expiry of the copyright.

2. I authorise the University of Tartu to make the work referred to in point 1 publicly available
through the University of Tartu online environment, including the digital archive DSpace,
under a Creative Commons licence CC BY NC ND 4.0, which permits reproduction, dis-
tribution and communication to the public with attribution to the author and prohibits the
creation of derivative works and the use of the work for commercial purposes, until the
copyright expires.

3. I am aware that the rights referred to in points 1 and 2 shall remain vested in the author.

4. I declare that by granting the non-exclusive licence, I do not infringe the intellectual prop-
erty rights of any other person or the rights of any person under the law on the protection of
personal data.

Sergei Markov

01.01.2026

ABSTRACT

Manual flashcard creation for spaced repetition systems such as Anki remains a time-intensive and technically demanding process, particularly when working with morphologically complex languages like Estonian. The study addresses the significant challenges associated with various methods of flashcard creation, which are often time-consuming and technically demanding due to the complexities of custom templates in Anki.

The primary objective of the research is to enhance the efficiency of flashcard creation by reducing the time and cognitive load associated with technical tasks, thereby enabling users to concentrate on educational content. The developed application, Anki Deck Creator, streamlines the process by providing an intuitive user interface, automating template formatting, and ensuring data format compatibility with Anki. Key functionalities include deck and flashcard management, structured data entry for dictionary entries and word forms, and automated export functionality.

The study evaluates the effectiveness of Anki Deck Creator through empirical testing involving 16 participants with varying levels of technical proficiency. The results of evaluation demonstrate that the automatic method significantly outperforms manual and semi-automatic approaches, reducing deck creation time by over 40% and improving user satisfaction and interface intuitiveness. Participants reported a significant reduction in cognitive and physical load, validating the application's ergonomic design and usability.

The study underscores the practical significance of the developed solution, demonstrating enhanced accessibility and efficiency of spaced repetition tools for language learners. The thesis concludes with recommendations for further research, including expanding the application's functionality for other languages and conducting larger-scale usability studies.

Keywords: spaced repetition, Anki, flashcard automation, Estonian language learning, usability evaluation.

CONTENTS

INTRODUCTION	6
Spaced Repetition	6
Objective of the Thesis	6
Tasks of the Thesis	7
1. COMPARATIVE ANALYSIS OF SPACED REPETITION	
SOFTWARE TOOLS	8
1.1. Evaluation Criteria	8
1.2. Comparative Analysis	8
1.2.1. Anki	8
1.2.2. Quizlet	9
1.2.3. SuperMemo	9
1.2.4. Memrise	9
1.3. Synthesis and Selection Rationale	9
1.4. Anki as a Spaced Repetition Tool	10
1.5. The Problem	11
2. DECK CREATION METHODOLOGIES	13
2.1. Manual Method	13
2.2. Semi-Automatic Method	14
2.3. Automatic Method	15
3. APPLICATION DEVELOPMENT	18
3.1. Justification of Development Tool Selection	18
3.1.1. Programming Language and Runtime Environment	18
3.1.2. User Interface Framework	18
3.1.3. Data Interchange Format	19
3.1.4. Comparative Analysis and Final Justification	19
3.2. Application Requirements	19
3.2.1. Functional Requirements	20
3.2.2. Non-Functional Requirements	23
3.2.3. Graphical User Interface	24
4. OPERATING PRINCIPLE	27
4.1. Selecting a Deck	27
4.2. Flashcard Creation	27
4.2.1. Entering Dictionary Entry Data	28
4.2.2. Entering Data on Word Forms	29
4.2.3. The ‘ <i>Формы слова</i> ’ Tab	30
4.2.4. The ‘ <i>Формы глагола</i> ’ Tab	30
4.2.5. Filtering the ‘ <i>Управление частью речи</i> ’ List	30

4.2.6. Entering Context of Use Data	31
4.2.7. Saving the Flashcard to the Deck	31
4.2.8. Working With the Flashcard List	32
4.2.9. Exporting And Importing the Deck	32
5. HYPOTHESES.....	36
5.1. Hypothesis 1	36
5.2. Hypothesis 2.....	36
6. STUDY METHODOLOGY	37
6.1. Target Group.....	37
6.2. Experimental Environment	37
6.2.1. Common Data Set.....	37
6.2.2. General Conditions	38
6.3. Process of Creating a Deck	38
6.4. Data Collection and Analysis	38
7. STUDY RESULT INTERPRETATION AND ANALYSIS	40
7.1. Limitations of the Study	40
Sample size and its implications	40
Non-representative skill distribution.....	40
Possible order effect	40
Single language and single deck template.....	41
Statistical testing	41
7.2. Estimating the Time to Create a Deck	41
7.2.1. Manual Method.....	42
7.2.2. Semi-automatic method.....	42
7.2.3. Automatic method	42
7.3. Evaluating user experience.....	43
7.3.1. Analyzing the intuitiveness of the user interface.....	43
7.3.2. Analysis of User Satisfaction.....	45
7.4. Result of the study	46
CONCLUSION	48
REFERENCES	49
APPENDICES	52
Appendix 1. Abbreviations and Glossary	52
Appendix 2. Overview of the formatting code for the flashcard template.....	54
Appendix 3. Evaluation Scales.....	56
Appendix 4. Test Data Set.....	58
Appendix 5. Example of CSV File Contents of Exported Deck	62

INTRODUCTION

Spaced Repetition

The rapid growth of available information has intensified the challenge of effective knowledge retention. The concept of spaced repetition originates from the pioneering work of the German psychologist Hermann Ebbinghaus, who first described the laws of forgetting in the late nineteenth century (Ebbinghaus, 2003). Based on his empirical observations of memory decay, Ebbinghaus proposed that information should be reviewed at strategically increasing time intervals – a principle that laid the foundation for what is now known as spaced repetition. This technique, grounded in cognitive psychology, has proven effective in enhancing memory retention and reducing forgetting rates, and has been widely applied across domains including foreign language acquisition and professional training.

The spaced repetition technique involves the periodic review of information at increasing time intervals, thereby activating the mechanisms of long-term memory (Cepeda et al., 2006). Empirical research confirms that distributed practice is significantly more effective than massed practice for long-term retention (Kornell, 2009).

Given the growing adoption of spaced repetition in educational practice, a number of dedicated software tools have been developed to implement this methodology. The following section examines the most widely used platforms to identify their strengths, limitations, and suitability for language learning contexts.

Objective of the Thesis

The objective of this thesis is to develop software capable of automating the creation and formatting of flashcards in Anki using a dedicated template for Estonian language study.

The application under development has been meticulously designed to address the following problems:

- Optimize the input and structuring of linguistic material through an intuitive user interface.
- Automate the application of a specialized flashcard template for Estonian language learning.
- Ensure formatting consistency with the prescribed template and export data in a format compatible with Anki.

The practical significance of the work lies in its potential to enhance the efficiency of creating educational materials by reducing the time allocated to the technical aspects of preparing flashcards. This, in turn, enables users to prioritize the content component of the educational process.

The present implementation is deliberately scoped to a single Estonian language template; however, the application architecture is designed to support extension to additional languages and templates with minimal modification to the codebase.

Tasks of the Thesis

- Conduct a comparative analysis of existing methods for creating Anki decks.
- Define the functional and non-functional requirements for the application.
- Develop the Anki Deck Creator application.
- Conduct empirical testing with a target group.
- Evaluate the effectiveness of the automated method based on time metrics and custom usability metrics (described in Appendix 3).

1. COMPARATIVE ANALYSIS OF SPACED REPETITION SOFTWARE TOOLS

This section presents a systematic comparative analysis of four spaced repetition software tools – Anki, Quizlet, SuperMemo, and Memrise – evaluated against criteria derived from software engineering quality models (ISO/IEC, 2023) and educational technology research (Nakata, 2011). The evaluation criteria are presented in the following section and draw on software engineering quality standards (ISO/IEC, 2023).

1.1. Evaluation Criteria

The selection of tools for analysis was guided by the following criteria, grounded in established software engineering principles (Sommerville, 2016):

- **Algorithmic sophistication:** Evidence-based scheduling mechanisms that support long-term retention.
- **Functional completeness:** Support for diverse content types (text, media, formulas) and customization options.
- **Usability and learnability:** Interface design adhering to user-centered principles.
- **Cross-platform compatibility and offline functionality:** Accessibility across devices and varying network conditions.
- **Extensibility and community support:** Availability of plugins, shared content, and active development.
- **Licensing and cost structure:** Transparency of pricing and alignment with open-access principles.
- **Data portability and privacy:** User control over content and independence from proprietary cloud services.

1.2. Comparative Analysis

1.2.1. Anki

Anki implements a variant of the SuperMemo SM-2 algorithm (Wozniak, 1990), with recent integration of the FSRS, which models memory states using retrievability, stability, and difficulty parameters (Ankitects Pty Ltd., n.d.). The tool supports heterogeneous content formats and extensive customization via user-developed add-ons, satisfying criteria of functional completeness and extensibility. Data are stored locally by default, ensuring user control and portability. However, the interface exhibits limited adherence to contemporary usability heuristics, resulting in a steeper learning curve (Surapaneni, 2023). The absence of integrated gamification or multimedia instructional support may reduce engagement among novice learners.

1.2.2. Quizlet

Quizlet prioritizes usability through a contemporary interface and multimodal learning activities (e.g., matching games, adaptive quizzes). Its extensive repository of user-generated content supports rapid content deployment. However, the spaced repetition implementation lacks the algorithmic granularity of Anki or SuperMemo: the platform’s own data indicate that the majority of users engage with content over too few sessions for the long-term learning algorithm to function effectively (Mooney, 2017), and the dedicated spaced repetition feature was discontinued in 2020 (Seibert Hanson & Brown, 2019; FLTMAG, 2024). Advanced features require a paid subscription, and offline functionality is restricted, potentially limiting accessibility in low-connectivity contexts.

1.2.3. SuperMemo

SuperMemo employs the proprietary SM-17 algorithm, designed to optimize inter-repetition intervals based on empirical memory research (SuperMemo World, n.d.). The platform supports complex knowledge structures and provides detailed analytics for progress monitoring. Nevertheless, the interface complexity and Windows-centric platform optimization reduce cross-platform usability. The limited availability of pre-existing content and higher cost structure may constrain adoption in educational settings with budgetary limitations.

1.2.4. Memrise

Memrise emphasizes language acquisition through curated multimedia content and mnemonic strategies, with an interface designed in accordance with modern UX standards (Nakata, 2011). However, the tool’s scope is predominantly linguistic, with limited applicability to other domains. Customization of repetition schedules is restricted, and core functionality requires an active internet connection or paid subscription. The underlying scheduling algorithm is less transparent and less configurable than those in Anki or SuperMemo.

1.3. Synthesis and Selection Rationale

Based on the evaluated criteria, Anki demonstrates the strongest alignment with requirements for a research-oriented SRS tool: it combines algorithmic transparency — demonstrated through native FSRS support — functional extensibility, data portability, and zero-cost access. While its usability presents initial barriers, these barriers are mitigated by comprehensive documentation and community support (Surapaneni, 2023). In contexts where immediate usability and multimodal engagement are prioritized over algorithmic control, Quizlet or Memrise may serve as supplementary tools. SuperMemo remains a viable option for users requiring advanced analytics, provided platform constraints are acceptable.

1.4. Anki as a Spaced Repetition Tool

For the purposes of this study, Anki was selected as one of the most flexible and functional tools that support the creation of custom card templates and for its balance of pedagogical efficacy, technical flexibility, and adherence to open-science principles (Nakata, 2011; Karpicke & Roediger, 2008).

The application supports the creation and review of digital flashcards through an evidence-based learning approach¹.

The program functions on the principle of active recall, whereby the user creates cards containing a question on one side and the answer on the other. During the learning process, the program displays the question, and the user attempts to recall the answer. The user subsequently verifies the answer by flipping the flashcard to its reverse side using the ‘*Show Answer*’ button (Figure 1).

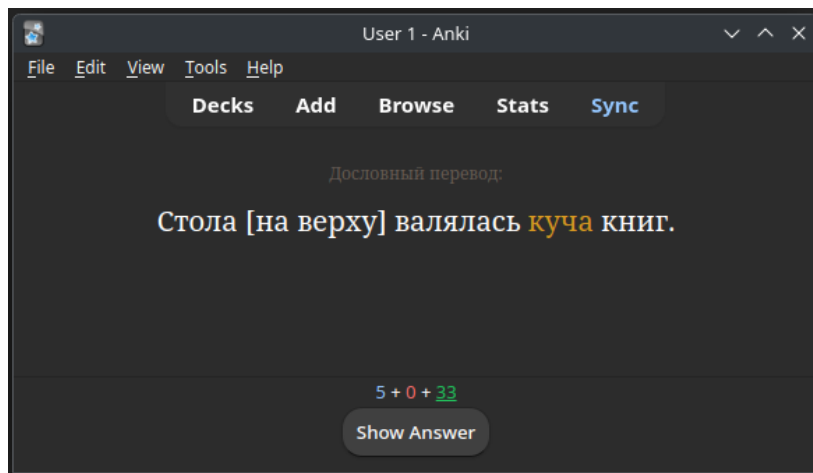


Figure 1. The front side of the flashcard with the question.

The user then evaluates the ease with which the information was recalled using the rating buttons labeled ‘*Again*’, ‘*Hard*’, ‘*Good*’, and ‘*Easy*’. The program determines the optimal time for the subsequent repetition of the flashcard, based on the user’s evaluation (Figure 2).

A core feature of Anki is the implementation of the SuperMemo SM-2 algorithm; recent versions additionally support the FSRS algorithm, which models memory states through retrievability, stability, and difficulty parameters (Ankitects Pty Ltd., n.d.). This enables users to allocate more review time to difficult material while progressively extending intervals for well-retained content. This approach thereby improves learning efficiency relative to conventional study methods.

¹ Evidence-based learning approach refers to instructional methods grounded in empirical research on human memory and cognition, as opposed to intuition-based or traditional practice.

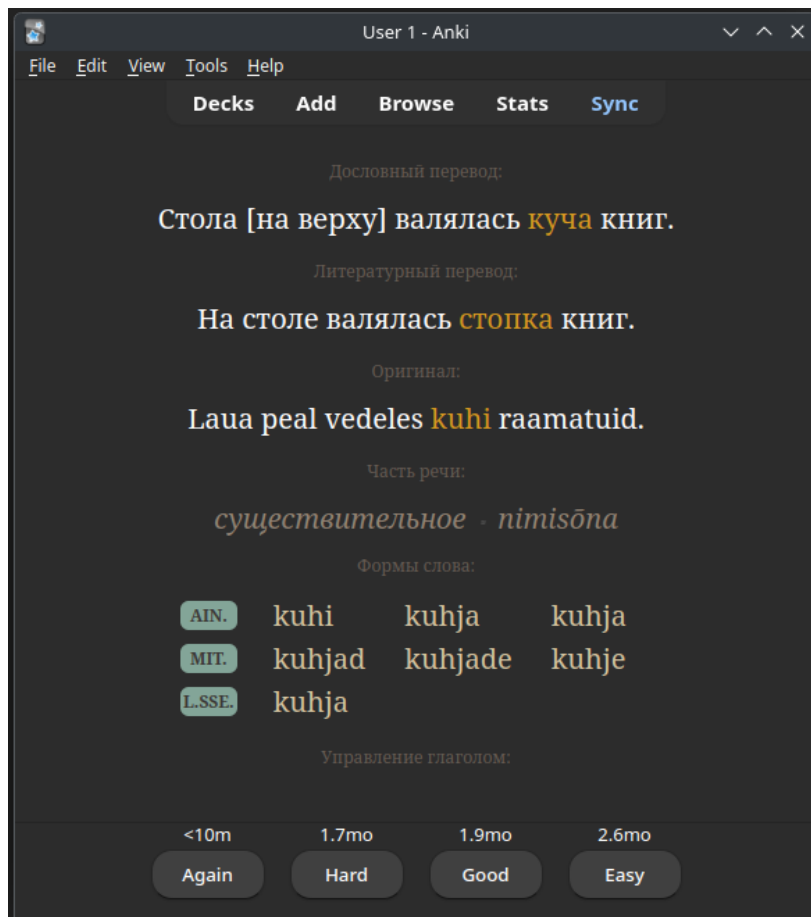


Figure 2. The back side of the flashcard with the evaluation buttons.

1.5. The Problem

Despite the extensive functionality of the Anki application, there are a number of significant limitations in working with custom card templates that considerably reduce the efficiency of using this tool compared to using standard templates.

The primary challenge associated with custom templates pertains to the substantial time investment required for their customization and maintenance. This limitation stems from the following factors:

1. The process of creating and modifying custom templates necessitates a foundation of knowledge in web technologies, specifically HTML and CSS. Individuals lacking proficiency in web development may encounter substantial challenges when attempting even minor alterations to card design, often requiring additional training.
2. The built-in editor has significant functional limitations. Despite the integration of a basic visual editor, its functionality does not provide sufficient tools for creating complex

types of cards². This forces users to edit template code directly, substantially increasing the application's complexity of use.

3. The automation of processes is inadequate. The absence of integrated automation tools for custom template updates and edits necessitates manual execution of routine operations, thereby increasing error rates and operational overhead.
4. The complexity of the content fulfillment process is a significant challenge. The utilization of custom templates frequently gives rise to numerous interrelated fields, accompanied by intricate display logic. This introduces significant challenges during data entry, requiring constant verification of each field's content.

In the context of this project, a custom template is employed, incorporating a field containing data on word forms of the lexeme under study. The arrangement of word forms in the template is carefully structured, ensuring a consistent sequence that is paramount for maintaining the uniformity of all cards. Users are required to verify and adjust the placement of specific word forms within the designated sections of the template. Erroneous placement of word forms necessitates further correction, which entails time and labor costs.

Furthermore, the fields for literal and literary translations, as well as for the original phrase, are structured identically. To ensure accuracy, users are cautioned not to enter, for instance, the text of the literal translation in the original phrase field.

5. The software is deficient in its lack of a data validation system. The software does not provide built-in mechanisms for checking the correctness of data entered into user templates, which can lead to incorrect card rendering and complicate the process of detecting and correcting errors, especially when working with large amounts of data.

The aforementioned factors combine to result in a marked decline in the efficiency of work with the program, thereby diverting the user's attention from the educational process to the resolution of technical issues. This effect is particularly pronounced among novice users, potentially undermining their motivation to continue using Anki.

Addressing these limitations has the potential to substantially improve the accessibility of custom templates and, consequently, enhance the functionality of Anki software as an educational tool.

² For the purposes of this study, a "complex type" of flashcards is defined as a configuration retaining the standard two-sided structure but employing complex formatting elements on the answer side, including tabular data, hierarchical lists, part-of-speech markers, and multi-level content structure.

2. DECK CREATION METHODOLOGIES

2.1. Manual Method

The manual method of flashcard creation involves sequentially entering data into the fields of a flashcard template using Anki tools. The process requires the user to perform repetitive operations: entering formatting commands into the appropriate fields (Figure 3) and populating those fields with lexical data by copying from a linguistic source (e.g., a dictionary, textbook, or language corpus) or typing manually. This procedure must be repeated for each field of every flashcard to be created.

The primary limitation of this method is the high time expenditure associated with repetitive manual data entry for each individual flashcard field (Figure 4).

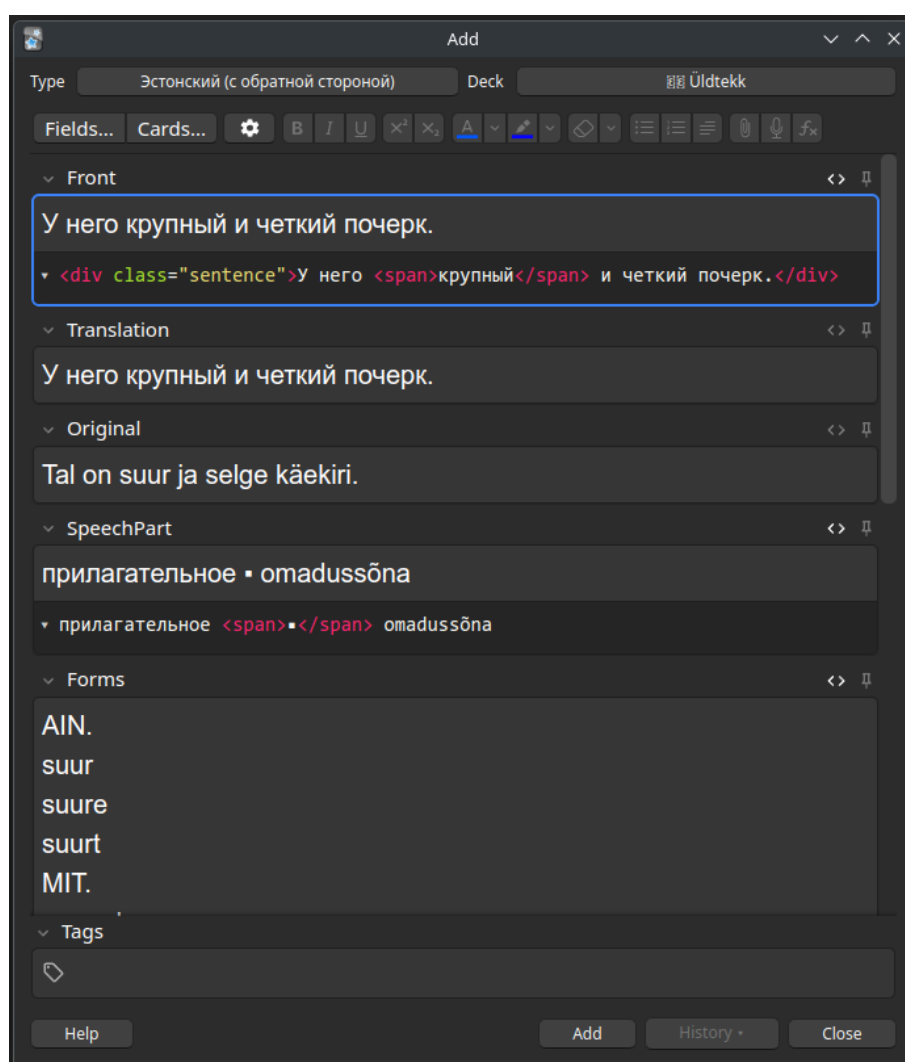


Figure 3. The Add screen prompting the user to input the pertinent fields ('Front', 'Translation', 'Original', 'SpeechPart', 'Forms', and others). If HTML formatting is deemed necessary, it can be applied to the content of the fields (such a field – Front – is highlighted in blue in this figure).

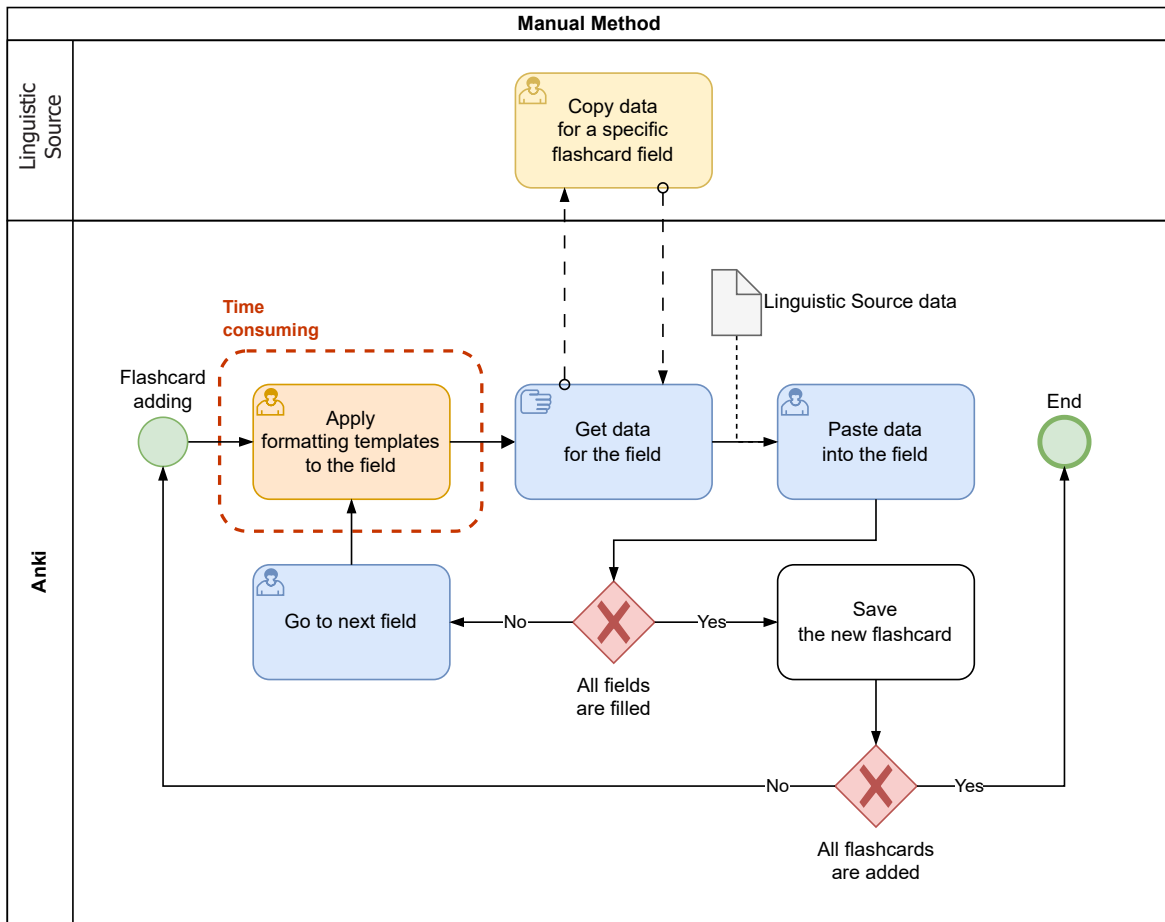


Figure 4. The diagram shows the process of adding a new flashcard using the Manual Method. The red border marks the process that consumes the most time and effort.

2.2. Semi-Automatic Method

This method is similar to the manual method, but differs in that the fields of the card are organized in a tabular format (Figure 5). Both the data and the formatting code for each field are entered manually into the corresponding cells of the table.

This method requires the use of a spreadsheet application (Figure 6).

The data structure is organized as follows:

1. Each row in the spreadsheet corresponds to a row in the exported CSV file. The first two rows constitute the file header, containing Anki's field delimiter and HTML formatting directives. The remaining rows represent the individual flashcards in the deck.
2. Each column in the spreadsheet row that describes a flashcard corresponds to a separate field of the flashcard template.

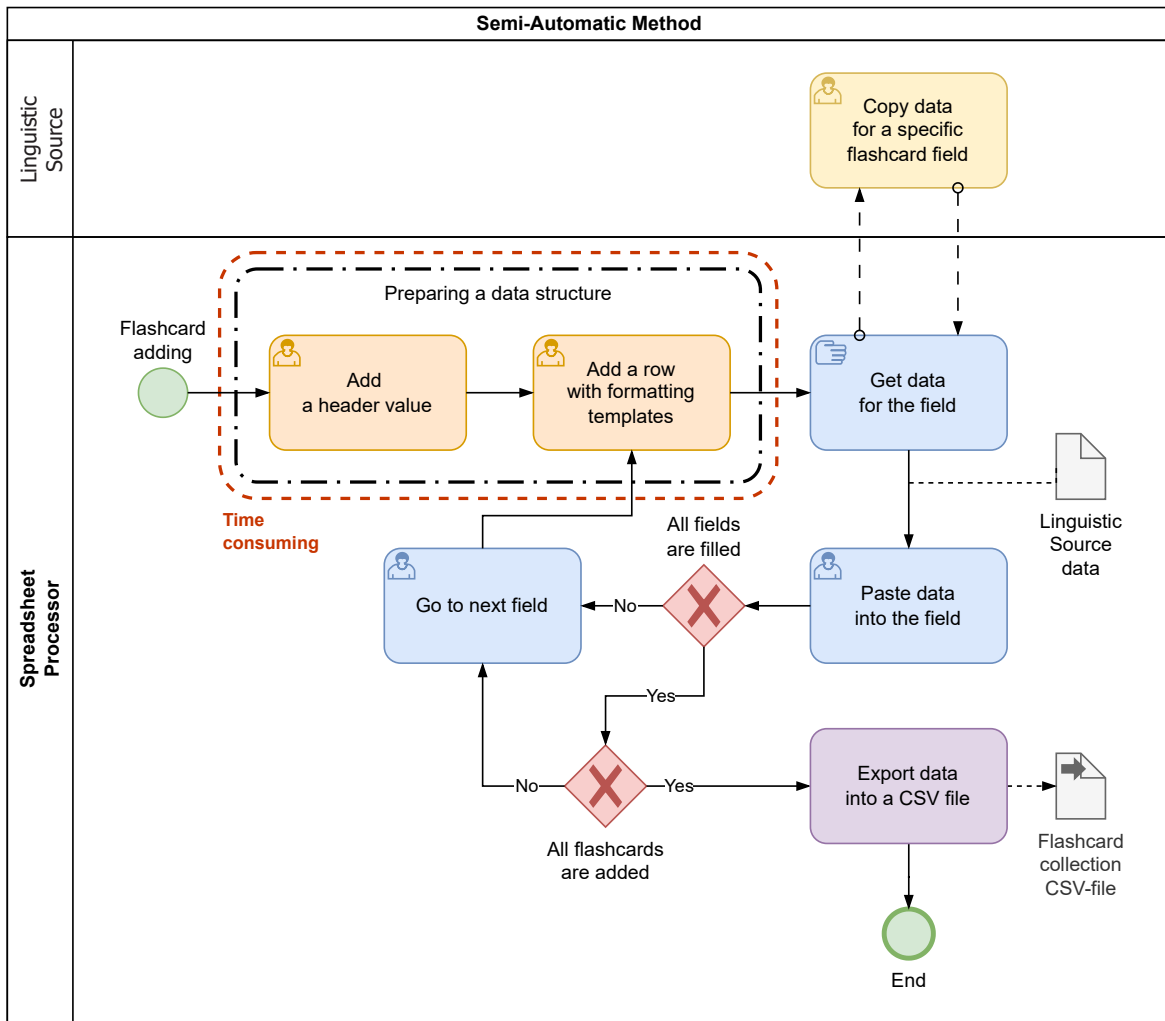


Figure 6. The diagram shows the process of adding a new flashcard using the Semi-Automatic Method. The red border marks the process that consumes the most time and effort.

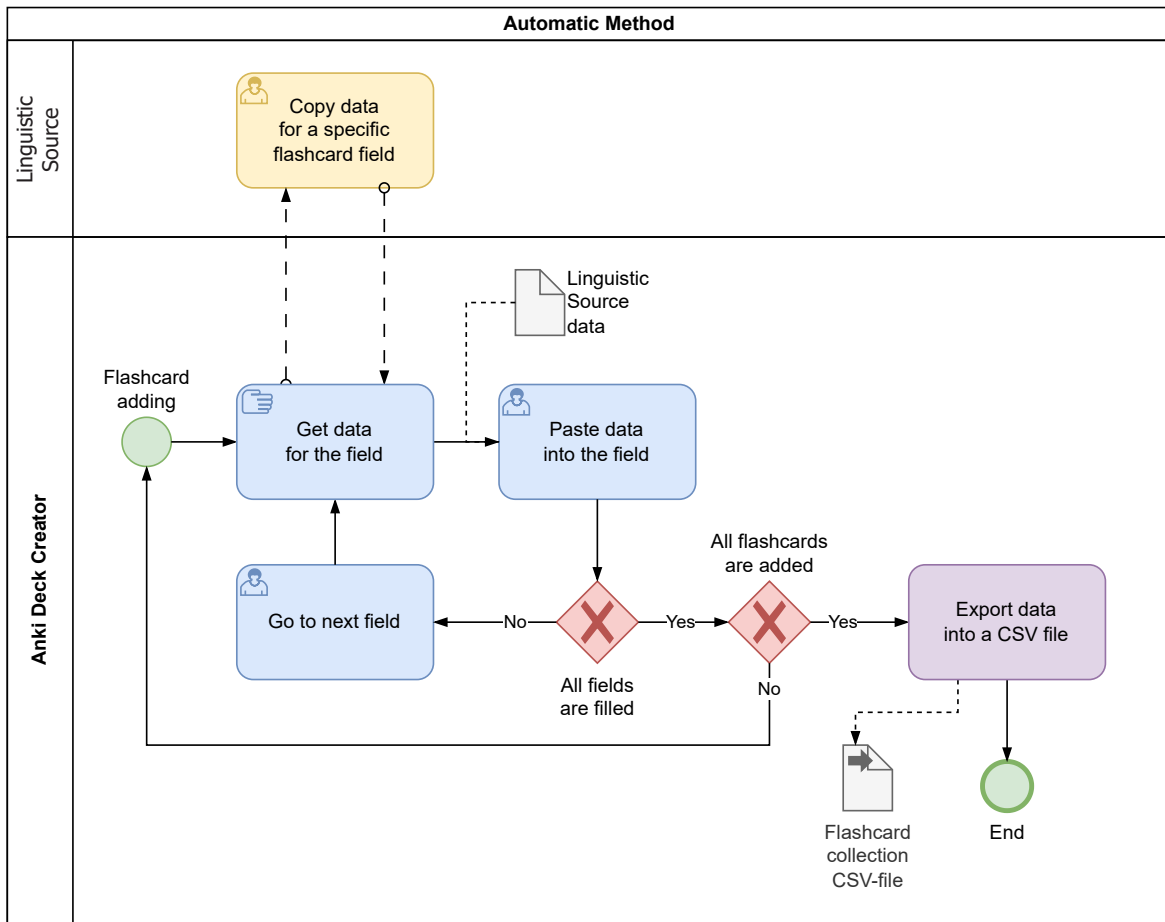


Figure 7. The diagram shows the process of adding a new flashcard using the Automatic Method.

3. APPLICATION DEVELOPMENT

3.1. Justification of Development Tool Selection

The selection of software development tools is governed by the project requirements, which prioritize cross-platform compatibility, long-term maintainability, type safety, and efficient data interchange. The evaluation criteria were derived from established software engineering practices and include ecosystem maturity, community support, licensing model, architectural alignment with proven design patterns, and interoperability with external systems (Sommerville, 2016). Based on these criteria, the development stack comprising C# with the .NET runtime, the Avalonia UI framework, and JSON is justified as follows.

3.1.1. Programming Language and Runtime Environment

C# was selected as the primary programming language due to its strong static typing, object-oriented paradigm, and structured exception-handling mechanisms, which collectively enhance code reliability, testability, and static analysis capabilities (Microsoft, 2023). The language's tight integration with the .NET platform provides a standardized base class library, significantly reducing implementation overhead for routine operations such as file system interaction and data serialization. Furthermore, .NET's transition to a unified, cross-platform runtime enables deployment across Windows, Linux, and macOS without platform-specific code branches, directly satisfying the project's portability requirements (Microsoft, 2023). The framework's permissive MIT/Apache licensing and sustained vendor stewardship ensure predictable long-term ecosystem stability.

3.1.2. User Interface Framework

Avalonia UI is adopted for desktop UI development to fulfill the requirement for a unified, cross-platform presentation layer. The framework enforces the MVVM architectural pattern, ensuring strict separation between business logic and interface components, which improves modularity, unit testing coverage, and long-term maintainability (Negi et al., 2023). Avalonia utilizes XAML for declarative UI definition, offering a development experience consistent with WPF while extending hardware-accelerated rendering across multiple operating systems and desktop environments (AvaloniaUI, 2024). Its actively maintained codebase, comprehensive control library, and extensibility through third-party plugins reduce implementation time and mitigate technical debt.

3.1.3. Data Interchange Format

JSON is designated as the primary data serialization and exchange format. Standardized as a lightweight, text-based notation (RFC 8259), JSON provides human-readable syntax and native parsing support across modern runtimes, facilitating debugging and interoperability with heterogeneous systems (Bray, 2017). Compared to verbose alternatives such as XML, JSON reduces payload size and parsing computational overhead, which is advantageous for network transmission and configuration management. Its schema-flexible design supports iterative development and seamless integration with RESTful APIs, aligning with the project's roadmap for third-party service integration.

3.1.4. Comparative Analysis and Final Justification

When evaluated against alternative stacks, the selected tools demonstrate optimal alignment with the project constraints. While languages such as Python or JavaScript offer rapid prototyping, C# provides superior compile-time type checking and performance predictability for business logic. Compared to native UI frameworks (WinUI, Cocoa, GTK), Avalonia delivers true cross-platform parity with a single codebase, avoiding the maintenance burden of platform-specific branches. In data serialization, JSON balances developer ergonomics and processing efficiency more effectively than binary formats — which require strict schema versioning — or XML, which incurs higher parsing overhead. Consequently, the C#/NET + Avalonia UI + JSON stack satisfies all defined selection criteria, ensuring sustainable development, reduced integration friction, and compliance with industry-standard architectural practices. This ensures that the resulting application remains accessible, maintainable, and extensible in the context of Estonian language flashcard automation.

3.2. Application Requirements

Requirements for the application being developed were structured in accordance with established software engineering practices (Wiegiers & Beatty, 2013; Sommerville, 2015; International Institute of Business Analysis, 2015).

All user interface element names retain their original Russian-language form, as implemented in the application. English translations are provided in the Glossary (Appendix 1).

3.2.1. Functional Requirements

This section presents the functional requirements that define the core behaviour of the application.

3.2.1.1. Deck and Flashcard Management

1. The system should allow the user to select a deck to edit.
2. The system shall provide for the creation of a new flashcard through a dedicated data entry interface.
3. The system should allow the user to initiate editing of a new flashcard.
4. The system shall provide storage of a list of flashcards in the deck.
5. The system shall allow the user to select one or more previously created flashcards from the list of flashcards in the deck.
6. The system should allow the user to add a new flashcard to the deck.
7. The system shall allow the user to remove an individual flashcard from the deck.
8. The system shall allow the user to remove all flashcards from the deck.
9. The system shall prompt the user to confirm the removal of an individual flashcard from the deck before executing the operation.
10. The system shall prompt the user to confirm the removal of all flashcards from the deck before performing the operation.
11. The system shall allow the user to confirm a request to remove an individual flashcard from the deck.
12. The system shall allow the user to reject a request to remove an individual flashcard from the deck.
13. The system shall allow the user to confirm the request to remove all flashcards from the deck.
14. The system shall allow the user to decline the request to remove all flashcards from the deck.

3.2.1.2. Work With the Flashcard Template Data

1. The system shall provide the ability for the user to change the value of the *‘Изучаемое слово или выражение’* field of the flashcard template.
2. The system shall provide the ability for the user to change the value of the *‘Часть речи’* field of the flashcard template.
3. The system shall limit the selection of values for the *‘Часть речи’* flashcard template field to a single value.
4. The system shall allow the user to change the value of the *‘Именительный падеж’* template field for singular number.
5. The system shall allow the user to change the value of the *‘Родительный падеж’* word card template field for singular number.
6. The system shall provide the ability for the user to change the value of the *‘Партитив’* flashcard template field for the singular.
7. The system shall provide the ability for the user to change the value of the *‘Именительный падеж’* flashcard template field for the plural.
8. The system shall provide the ability for the user to change the value of the *‘Родительный падеж’* flashcard template field for the plural.
9. The system shall provide the ability for the user to change the value of the *‘Партитив’* template field for plural.
10. The system shall allow the user to change the value of the *‘Короткая форма иллатива’* field of the flashcard template.

11. The system shall provide provide the ability for the user to change the value of the '*МА-инфинитив*' flashcard template field.
12. The system shall allow the user to change the value of the '*ДА-инфинитив*' flashcard template field.
13. The system shall provide the ability for the user to change the value of the flashcard template field '*Изъявительное наклонение (3 лицо)*'.
14. The system shall allow the user to change the value of the '*Причастие (страдат. прои.)*' field of the flashcard template.
15. The system shall provide the ability for the user to change the value of the flashcard template field '*Изъявительное наклонение (3 лицо, прои.)*'.
16. The system shall provide the ability for the user to change the value of the '*Причастие (действ. прои.)*' field of the flashcard template.
17. The system shall provide the ability for the user to change the value of the flashcard template field '*Повелительное наклонение (ед.ч.)*'.
18. The system shall allow the user to change the value of the '*Страдательные залог (наст.)*' field of the flashcard template.
19. The system shall provide the user with the ability to select one or more values for the '*Управление частью речи*' field of the flashcard template.
20. The system shall allow the user to select multiple values for the '*Управление частью речи*' field of the flashcard template.
21. The system shall provide filtering of the list of values of the flashcard template field '*Управление частью речи*' by the first letter of the value.
22. The system must display all list values for the '*Управление частью речи*' flashcard template field if the user has not selected a filtering criterion for the list of values.
23. The system shall provide the ability for the user to reset all applied filters for the flashcard template field '*Управление частью речи*'.
24. The system shall allow the user to change the value of the '*Дословный перевод*' field of the flashcard template.
25. The system should allow the user to change the value of the '*Литературный перевод*' field of the flashcard template.
26. The system should allow the user to change the value of the '*Оригинал*' field of the flashcard template.

3.2.1.3. User Interface Requirements

1. The system should provide a special interface in the '*Контекст*' tab where users can mark key words within the learned word's usage context.
2. The interface for marking the learned word in the text of the context of use should provide a choice of one or more words for formatting.
3. The interface of marking the learned word in the text of the context of use should ensure that previously selected words are deselected when the marked word is clicked again.
4. The system should center the main application window in the center of the screen after the initial boot.
5. The system should allow the user to resize the main application window.
6. The system should allow the user to minimize the main application window.
7. The system should allow the user to maximize the main application window.
8. The user interface should be divided into two screens: the Deck Selection Screen, the Edit Screen.

9. The system shall display the deck list on the Deck Selection Screen as horizontally arranged buttons.
10. The deck selection button within the Deck Selection Screen must display the national flag corresponding to the language being learned.
11. The system should place the Button Bar on the left side of the Edit Screen.
12. The system should place Tabs Bar on the right side of the Edit Screen.
13. The system should display the following buttons on the Button Bar in the following order from top to bottom: *‘Новая’*, *‘Добавить’*, *‘Удалить’*, *‘Очистить’*, *‘Экспорт’*, *‘Выход’*.
14. The system should arrange the tabs on the Tabs Bar in the following order: *‘Словарная статья’*, *‘Формы слова’*, *‘Формы глагола’*, *‘Контекст’*, *‘Список карточек’*.
15. The system should hide the *‘Формы слова’*, *‘Формы глагола’* tabs if the user has not selected any value for the *‘Часть речи’* field of the flashcard template.
16. The system must display the *‘Формы слова’* tab if the user selects any value from the list of values for the *‘Часть речи’* field of the flashcard template other than *‘Глагол’* or *‘Глагол составной’*.
17. The system must display the *‘Формы глагола’* tab if the user has selected *‘Глагол’* or *‘Глагол составной’* from the list of values in the *“Часть речи”* field of the flashcard template.
18. The system should display the following tabs after the Edit Screen is displayed: *‘Словарная статья’*, *‘Контекст’*, *‘Список карточек’*.
19. The system should group the following fields of the flashcard template on the *‘Словарная статья’* tab: *‘Изучаемое слово или выражение’*, *‘Часть речи’*.
20. The system shall group the following fields of the flashcard template on the *‘Формы слова’* tab: *‘Именительный падеж (единственное число)’*, *‘Именительный падеж (множественное число)’*, *‘Родительный падеж (единственное число)’*, *‘Родительный падеж (множественное число)’*, *‘Партитив (единственное число)’*, *‘Партитив (множественное число)’*, *‘Короткая форма иллатива’*.
21. The system shall group the following fields of the word card template on the *‘Формы глагола’* tab: *‘Да-инфинитив’*, *‘Изъявительное наклонение (3 лицо)’*, *‘Причастие (страд. прош.)’*, *‘Изъявительное наклонение (3 лицо, прош.)’*, *‘Причастие (действ. прош.)’*, *‘Повелительное наклонение (ед.ч.)’*, *‘Страдательный залог (наст.)’*, *‘Управление частью речи’*.
22. The system should group the following fields of the word card template on the *‘Контекст’* tab: *‘Дословный перевод’*, *‘Литературный перевод’*, *‘Оригинал’*.
23. The system must provide for each text field responsible for editing template field data a text label identifying the type of input data.
24. The system must provide for each text field responsible for editing of template field data the presence of control element for inserting data from the system clipboard in one action. The system shall display the list of part-of-speech names on the *‘Словарная статья’* tab as independent switches.
25. The system shall provide the elements of the list of values of the *‘Управление частью речи’* flashcard template field with independent switches, so that the user can select the elements only by mouse without using the keyboard.
26. The system shall provide the element of the list of values of the field value list of the *‘Управление частью речи’* flashcard template with a text label displaying the current value of the list element.
27. The system shall display the text values of the value list items of the *‘Управление частью речи’* flashcard template field in lower case.

28. The system shall sort the elements of the list of values of the *‘Управление частью речи’* flashcard template field alphabetically in descending order.
29. The system should display all flashcards of the deck as a list on a separate tab of the user interface – *‘Список карточек’*.
30. The system shall display an information label with the text *‘Двойной щелчок для правки карточки’* below the list of flashcards in the current deck on the *‘Список карточек’* tab.
31. The system shall display the quantity count of the current deck’s flashcards in the header of the *‘Список карточек’* tab.
32. The system should sort the items on the *‘Список карточек’* tab in the order in which they are added to the deck: earlier cards are displayed first, then later cards.

3.2.1.4. System Requirements

1. The system shall allow the user to paste text from the system clipboard into text fields in a single action.
2. The system must sanitize unwanted characters from data pasted from the system clipboard into text fields.
3. The system shall store the flashcard template structure in a separate file.
4. The system must store the values for the *‘Часть речи’* flashcard template field in a separate file.
5. The system must store the values for the flashcard template field *‘Управление частью речи’* in a separate file.
6. The system must load the list of values for the *‘Часть речи’* flashcard template field when the application is loaded.
7. The system must load the list of values for the flashcard template field *‘Управление частью речи’* when the application is loaded.
8. The system shall allow the user to correctly end the application through the button on the Deck Selection Screen.
9. The system must allow the user to correctly end the application on the Edit Screen through the button on the Button Bar.
10. The system should prompt the user to export the deck to a file when exiting the application if the deck has not been exported previously.
11. The system should notify the user when the deck has been successfully exported. The system must support exporting decks to a CSV file.
12. The system should allow the user to select the file path where the current deck will be exported.

3.2.2. Non-Functional Requirements

Given the defined scope of this study, the primary methodological focus was placed on user interaction scenarios and functional requirements. Non-functional requirements including performance, security, and reliability – were addressed at a baseline level in accordance with standard software engineering practices (Sommerville, 2016).

The application is required to operate on Windows, macOS, and Linux; no authentication mechanism is implemented, as the application processes locally stored data exclusively.

3.2.3. Graphical User Interface

In accordance with the principles of cognitive ergonomics and human-machine interaction theory, the application user interface design incorporates several key tenets (Nielsen, 1993; Norman, 2013). The application architecture is predicated on a two-level navigation model, incorporating the Deck Selection Screen (Figure 9) and the Edit Screen (Figure 8).

The layout of the Edit Screen is predicated on the binary division of functional zones: the Button Bar (left panel) and the information-manipulative area (right panel). This segmentation aligns with the principles of Gestalt psychology concerning the aggregation of related elements, thereby supporting the formation of a coherent mental model of system interaction.

The left panel of the Edit Screen consolidates data manipulation functions at both the individual flashcard and deck levels. It also includes export and session termination controls. The right-hand area, organised into thematic tabs, provides access to the various fields of the flashcard template.

The segmentation of input fields into thematic tabs represents an implementation of the principle of cognitive decomposition, which serves to minimize the mental load on the user. This approach establishes an intuitive correlation between the semantic elements of the dictionary and the corresponding interface areas, such as morphological forms or the contextual use of lexical units.

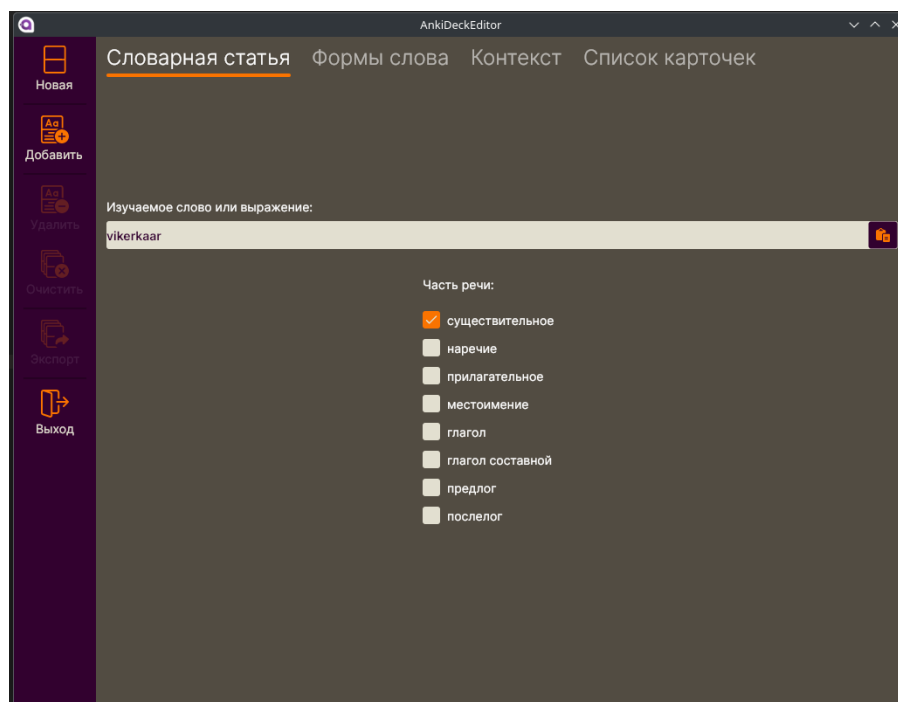


Figure 8. Edit Screen with Button Bar and the Tabs of the editing steps.

Each input field is equipped with a dedicated clipboard paste mechanism. This mechanism eliminates the need to use the context menu of the input field and reduces the physical effort required to interact with the interface. This solution is based on the principles of minimizing physical user actions.

The visual design of the interface is characterized by minimalism and high contrast, which corresponds to the principles of perceptual ergonomics. The minimalist design reduces cognitive noise, allowing the user to concentrate on the primary task. Concurrently, the high-contrast design optimizes visual search for controls.

The initialization of the editor in a state of readiness to create a new flashcard is predicated on the principle of anticipation of user actions. Removing the requirement to manually activate the new flashcard mode via the '*Новая*' button eliminates a redundant interaction step.

The interface architecture thus reflects a consistent application of ergonomic design principles aimed at maximising interaction efficiency while minimising cognitive and physical load.

3.2.3.1. Colors

Dark gray-brown (the main window background): The reduction of text contrast to a comfortable level has demonstrated minimized eye strain during extended periods of visual task performance (Hall & Hanna, 2004). Unlike a white background, which increases luminance-related eye strain, this approach reduces visual discomfort during prolonged use (Hall & Hanna, 2004). This feature enables users to distinctly highlight controls and work areas, enhancing visibility and ease of use.

Rich orange (icons on the buttons): this color's high visibility against the Button Bar's dark background draws attention to controls, making them easily identifiable. Furthermore, rich orange is associated with positive affect, activity, and energy (Valdez & Mehrabian, 1994). In addition, unlike red, the color does not elicit comparable visual tension.

Dark purple (the Button Bar background): The visual separation of the Button Bar from the main workspace is achieved by implementing a color scheme that differentiates these two elements. Dark purple generates sufficient contrast with the orange button icons, does not compete with the main content for attention, and establishes a coherent visual hierarchy within the interface.

The combination of the colors listed above provided several benefits in a professional setting:

- Facilitates effective navigation within the interface.
- Reduces visual fatigue during extended use.
- Enables users to swiftly locate essential controls.
- Fosters a comfortable work environment by avoiding excessive brightness and contrast.

The selected color scheme aims to balance functionality and visual comfort. Significant attention was devoted to mitigating visual strain during extended work periods, facilitating intuitive navigation, and establishing a clear visual hierarchy of interface elements. A notable feature of this scheme is its accessibility for users with color vision deficiencies, achieved through emphasis on luminance contrast rather than hue differentiation.

4. OPERATING PRINCIPLE

4.1. Selecting a Deck

Prior to launching the application, the user must select the '*Eesti Keel*' (est., 'Estonian Language') deck from the list present on the Deck Selection Screen (Figure 9). The Deck Selection Screen also contains a button that allows the user to exit the application.

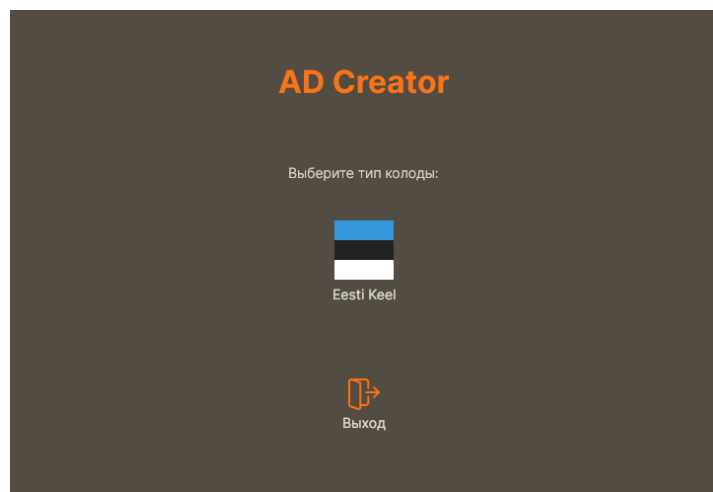


Figure 9. The Deck Selection Screen of the app with a list of decks (currently the only deck is '*Eesti Keel*').

4.2. Flashcard Creation

Following deck selection on the Deck Selection Screen, the application transitions to the Edit Screen.

The process of creating a flashcard entails a series of steps, each requiring the user to enter data into specific fields of the flashcard template. The sequence and execution of these steps may be adjusted at the user's discretion. Each step is delineated on a designated tab of the user interface (Figure 10).

Users may navigate between tabs in any order. Each tab contains a dedicated set of UI controls.

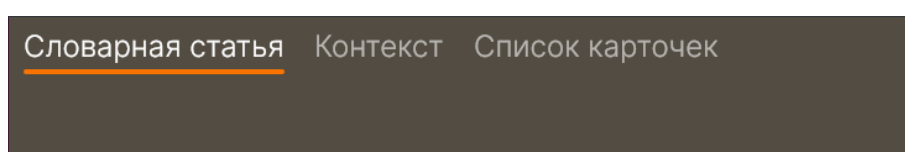


Figure 10. The flashcard editing steps tabs.

The editing process involves entering the following data:

- Data about the dictionary entry.
- Data on word forms (optional).
- Data on context of use.

4.2.1. Entering Dictionary Entry Data

At this stage of the process, the user is required to enter the word or expression that is the subject of learning in the '*Изучаемое слово или выражение*' field (Figure 11).

This field is required by Anki for the following purposes:

- as a value when displaying the list of flashcards in the deck;
- as a criterion when searching for a flashcard in decks.

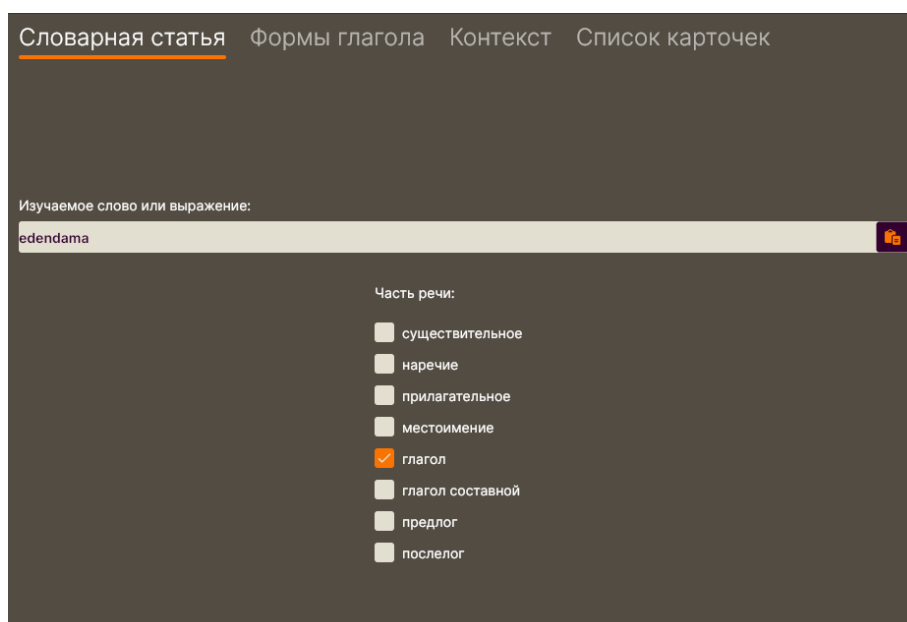
The screenshot shows the 'Словарная статья' (Dictionary entry) tab in the Anki editor. At the top, there are four tabs: 'Словарная статья' (selected), 'Формы глагола' (Verb forms), 'Контекст' (Context), and 'Список карточек' (List of cards). Below the tabs, there is a label 'Изучаемое слово или выражение:' (Word or expression to be learned:) followed by a text input field containing the word 'edendama'. To the right of the input field is a small red icon. Below the input field, there is a section titled 'Часть речи:' (Part of speech:). It contains a list of grammatical categories with corresponding checkboxes: 'существительное' (noun), 'наречие' (adverb), 'прилагательное' (adjective), 'местоимение' (pronoun), 'глагол' (verb) - which is checked with an orange checkmark, 'глагол составной' (phrasal verb), 'предлог' (preposition), and 'послелог' (postposition).

Figure 11. The data about the learned word 'edendama' (est, 'develop', 'promote') is introduced and the part-of-speech type '*Глагол*' is selected.

When a user creates a flashcard for a single word, it is recommended that they select a value from the '*Часть речи*' list. Conversely, if a flashcard is created for an entire expression, selecting a value from this list is unnecessary, as the expression as a whole – comprising heterogeneous grammatical elements – is memorized as a single unit.

The data entered in this step will be displayed on the front and back of the flashcard (Figure 13 and Figure 15 respectively).

4.2.2. Entering Data on Word Forms

If any value from the ‘*Часть речи*’ list was selected in the preceding step (“4.2.1. Entering Dictionary Entry Data”), the application displays an additional tab for the corresponding data entry step.

Depending on the part of speech selected by the user, the following tabs will be displayed:

- ‘*Формы слова*’ tab when selecting any values except ‘*Глагол*’ and ‘*Глагол составной*’ (Figure 12).
- ‘*Формы глагола*’ tab when selecting ‘*Глагол*’ and ‘*Глагол составной*’ (Figure 14).

Словарная статья Формы слова Контекст Список карточек

ЕДИНСТВЕННОЕ ЧИСЛО МНОЖЕСТВЕННОЕ ЧИСЛО

Именительный падеж

abinõu abinõud

Родительный падеж

abinõu abinõude

Партитив

abinõu abinõusid

Короткая форма иллатива

Figure 12. The ‘*Формы слова*’ tab is displayed.

Дословный перевод:

Проблемы [на решении] [было бы] [в суд] обращение крайний мера.

Литературный перевод:

Обращение в суд было бы крайней мерой для решения проблемы.

Оригинал:

Probleemi lahendamisel oleks kohtusse pöördumine äärmuslik abinõu.

Часть речи:

существительное · nimisõna

Формы слова:

AIN.	abinõu	abinõu	abinõu
MIT.	abinõud	abinõude	abinõusid

Управление глаголом:

Figure 13. Forms of the word ‘*abinõu*’ (6 meanings) at the bottom of the back of the flashcard.

Словарная статья **Формы глагола** Контекст Список карточек

Управление частью речи:

Фильтр: **Е** К М Reset

МА-инфинитив:
edendama

DA-инфинитив:
edendada

Изъявительное наклонение (3 лицо):
edendab

Причастие (страд., прош.):
edendatud

Изъявительное наклонение (3 лицо, прош.):
edendas

Причастие (действ. прош.):
edendanud

Повелительное наклонение (ед.ч.):
edenda

Страдательный залог (наст.):
edendatakse

Контрольный список:

- ☐ kui palju
- ☐ kuidas
- ☐ kus
- ☐ kust
- ☐ kust + kuhu
- ☒ mida
- ☐ mida + millega
- ☐ mida tegema
- ☐ mida tegemas
- ☐ mida tegemast
- ☐ mida tegevat
- ☐ mida teha
- ☐ midagi tegemata
- ☐ mille all
- ☐ mille alla
- ☐ mille eest
- ☐ mille järel
- ☐ mille järgi
- ☐ mille käes

Figure 14. Data on verb-type word forms.

4.2.3. The ‘*Формы слова*’ Tab

In this tab, users are prompted to enter data concerning forms of words that do not constitute verbs. All fields in this tab are optional. The data from this tab is displayed on the back of the flashcard (Figure 15).

4.2.4. The ‘*Формы глагола*’ Tab

In this tab, the user enters data on the forms of the word being a verb. All fields in this tab are optional. In addition to the verb forms, the user may select one or more values for the verb government type from the ‘*Управление частью речи*’ list, if applicable.

The data obtained from this step will be displayed on the back of the flashcard (Figure 15).

4.2.5. Filtering the ‘*Управление частью речи*’ List

If needed, the user may filter the ‘*Управление частью речи*’ list using a dedicated filter control (Figure 16). Filtering is performed by the first letter of the list entry. The user applies the filter by selecting the relevant criterion via the filter button. In the absence of any specified filter criteria, the list will present all available values. The ‘*Сброс*’ button resets the current filter criteria. The filtered list displays only entries whose first letter matches the selected criterion.



Figure 15. The verb forms displayed on the back of the flashcard.

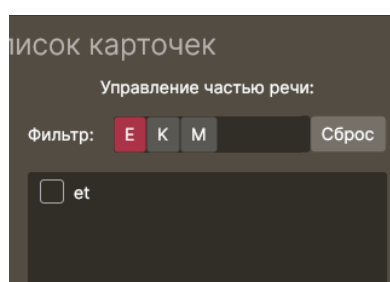


Figure 16. The list is filtered by the first letter ('E') of its values.

4.2.6. Entering Context of Use Data

At this stage, the user enters data regarding the context of use of the word or expression being learned (Figure 17).

Subsequent to entering data concerning the context of use in the designated text field, the user may highlight one or more instances of the studied word within the context using the 'Маркер' control. The selected words will be highlighted in a distinct color on the flashcard (Figure 18).

4.2.7. Saving the Flashcard to the Deck

After entering the required data, the user adds a new flashcard to the deck by activating the 'Добавить' button located on the Button Bar. The application then adds the new flashcard to the list displayed on the 'Список карточек' tab.

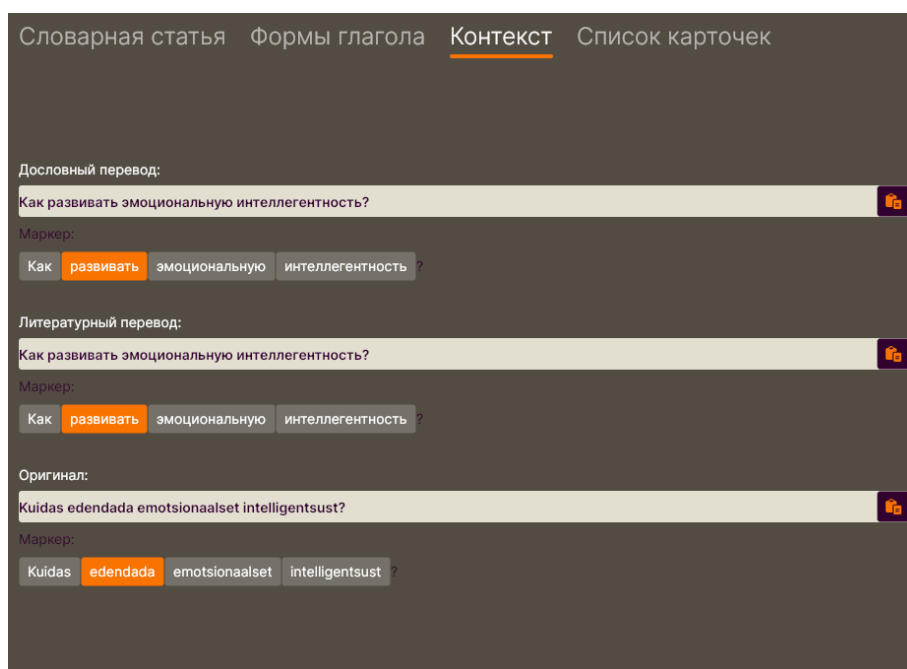


Figure 17. Context of use of the studied word with marked meanings (highlighted in orange).

4.2.8. Working With the Flashcard List

Once added, flashcards are accessible via the ‘Список карточек’ tab. Flashcards added to the deck are displayed in the list on the ‘Список карточек’ tab (Figure 19).

The ‘Удалить’ and ‘Очистить’ buttons on the Button Bar (Figure 8) facilitate the removal of a flashcard from the deck and the clearing of the entire list, respectively. Furthermore, the list items are accompanied by checkboxes, enabling users to select one or more required list items. By double-clicking a list item, users can initiate the editing process for a specific flashcard, as indicated by a status label at the bottom of the list.

4.2.9. Exporting And Importing the Deck

To add the flashcards created in the application to Anki, the user must perform a sequence of export and import operations. The first step involves exporting the deck to a file. During the export process, the user is presented with the option to specify the target directory for the exported file (Figure 20). Thereafter, the exported file must be imported into Anki (Figure 21), enabling the deck’s flashcards to be reviewed within Anki (Figure 22).

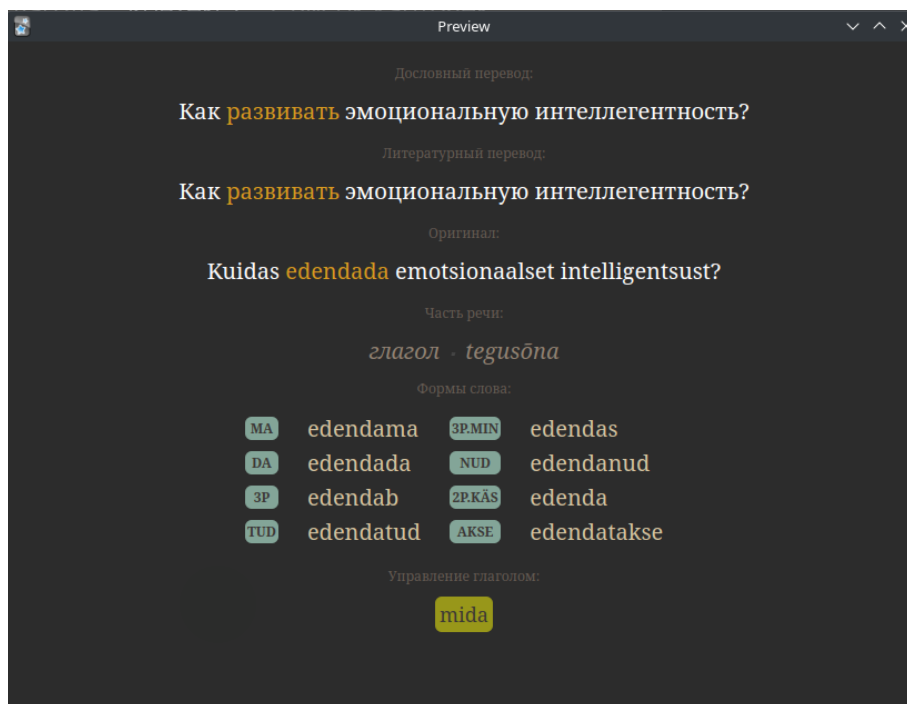


Figure 18. The yellow highlighted learned word is on the back of the flashcard at the top.

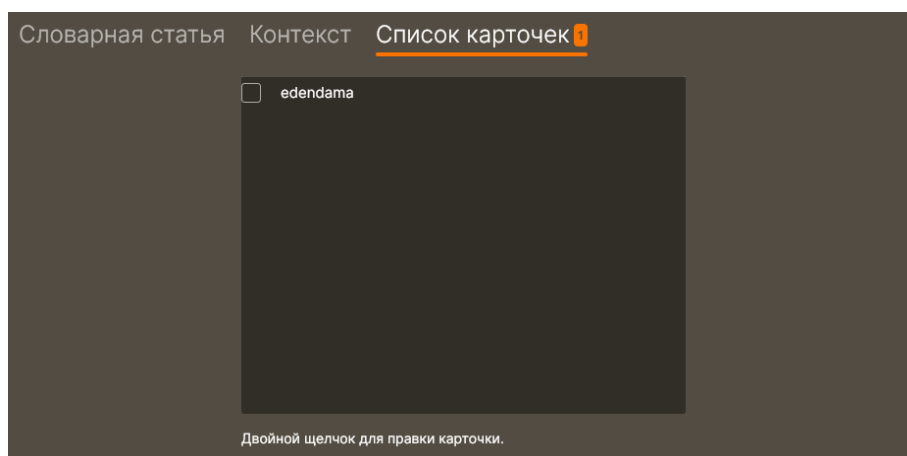


Figure 19. List of flashcards in the collection on the ‘Список карточек’ tab.

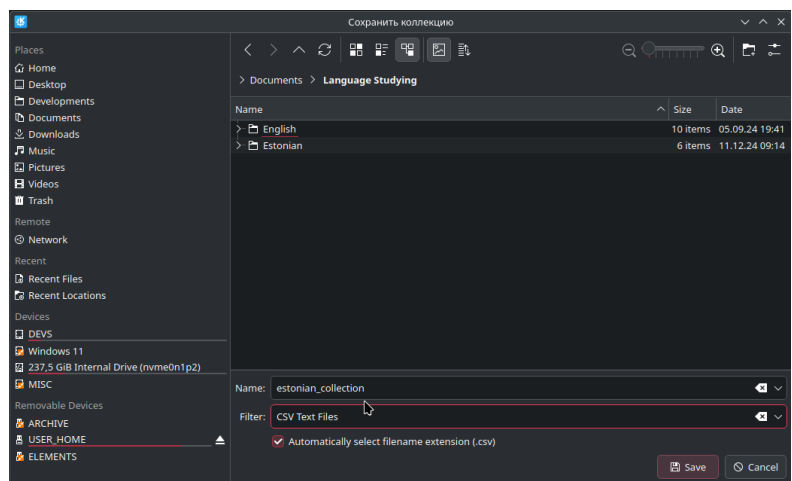


Figure 20. The dialog for selecting a path for exporting a deck.

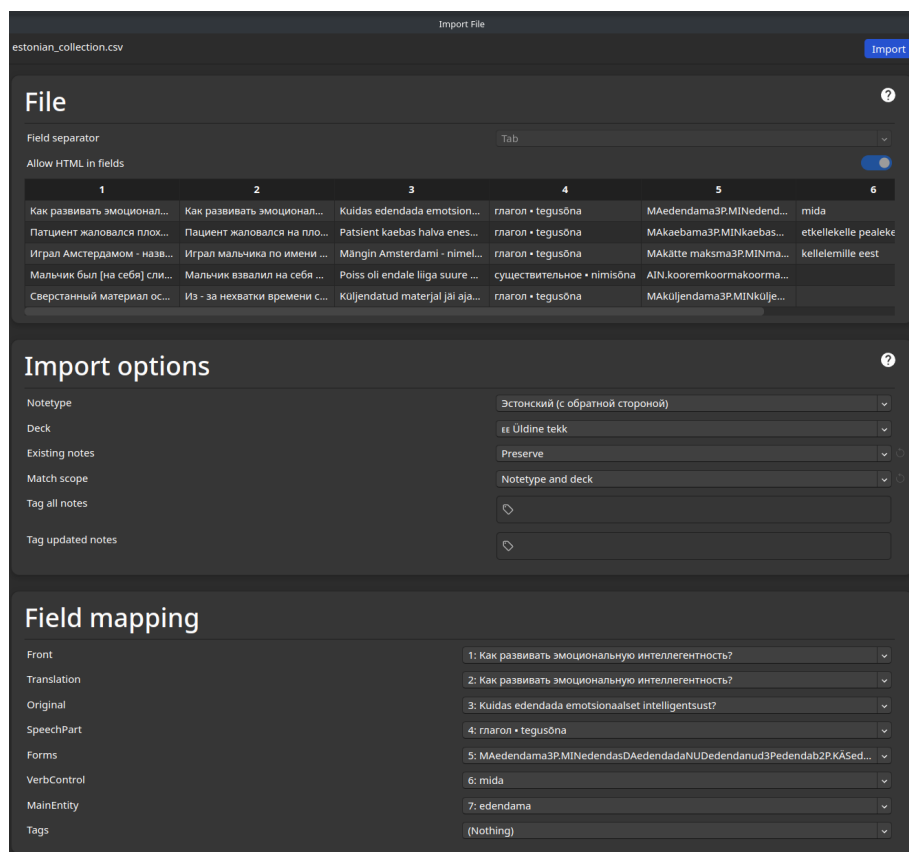


Figure 21. The dialog for importing a deck of the Anki program file.

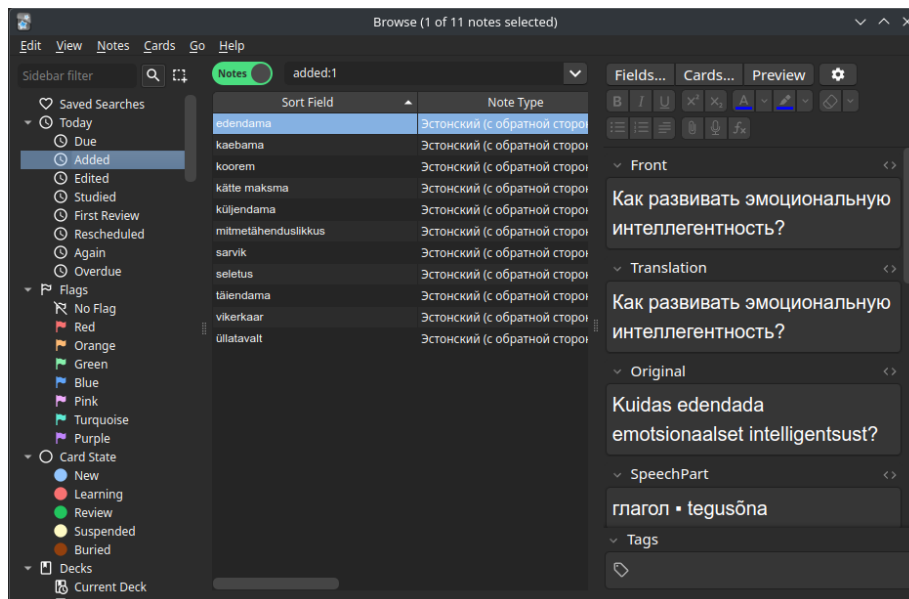


Figure 22. Flashcards successfully imported from a file.

5. HYPOTHESES

The formulation of hypotheses for the study was based on an analysis of deck formation methods, including manual, semi-automatic, and automatic methods.

5.1. Hypothesis 1

Anki Deck Creator reduces the time required to create a deck of 10 flashcards by at least 40% compared to manual and semi-automatic methods.

Rationale: The time required for flashcard creation can be substantially reduced by automating the following steps in the process:

- Applying a uniform card template, thereby eliminating the need for manual formatting of each card.
- Automatically allocating input data to the corresponding card fields.

5.2. Hypothesis 2

The application reduces the cognitive and physical load on the user when creating a deck of 10 cards as measured by the custom usability metrics (Appendix 3).

Rationale. Anki Deck Creator provides the following advantages:

- simplifies deck creation through a structured workflow, reducing both physical and cognitive load on the user;
- facilitates vocabulary card creation, thereby reducing user fatigue and improving motivation to maintain decks regularly;
- improves the overall user experience of working with Anki.

6. STUDY METHODOLOGY

6.1. Target Group

The survey participants were Estonian language learners who use Anki in their educational practice.

The study included 16 participants representing diverse age groups (Table 1). The respondents were characterized predominantly by a high level of technical competence.

The mean age of the participants was 29 years.

With respect to their experience with computers, participants demonstrated varying levels of proficiency, as shown in Table 2 (Appendix 3).

Table 1: Distribution of participants by age.

Age	Number
Up to 30	10
31-45	5
46 and over	1

Table 2: Distribution of participants by computer skills.

Level	Number
Basic	1
Junior	0
Middle	4
Senior	7
Expert	4

6.2. Experimental Environment

All participants provided informed consent prior to the study. No personally identifiable data were recorded; results were anonymised at the point of collection.

6.2.1. Common Data Set

In accordance with the experimental protocol, each participant was provided with a pre-generated data set (Appendix 4), comprising vocabulary items derived from an Estonian language lexicographic source.

This methodological decision enabled the exclusion of the following variables from the temporal measurements:

1. The influence of individual differences in input device skills, such as speed and manner of typing on a keyboard and interaction with a mouse.
2. Interaction with the lexicographic data source (electronic dictionary or printed edition).
3. The time required to search for information to be entered into the corresponding flash-

card fields. In particular, it refers to the selection of authentic contextual example (for the ‘*Original*’ flashcard field) of the lexeme under study in case of its absence in the dictionary source.

6.2.2. General Conditions

Standardized experimental conditions were established for all participants to ensure comparability of results. The work environment was configured to minimize cognitive and physical load when switching between applications. To ensure a seamless experience, participants were permitted to utilize personal peripheral input devices, such as keyboards and mice, promoting familiarity, particularly in face-to-face sessions.

The study, contingent upon the participants’ capabilities, was conducted in two formats: in-person and remote. The remote format was implemented using remote desktop software (TeamViewer, RustDesk) to provide access to the test environment. During the preparatory stage, all participants were acquainted with the structure of the flashcard and instructed on the utilization of template tags to format its elements.

Participants with no prior experience with spreadsheet software received brief training in basic cell and sheet editing operations, reducing the time required to complete the semi-automatic method tasks.

In order to prevent potential distortion of results and to minimize the number of errors due to rushed performance, participants were not informed prior to the test that the time taken to create and compile the deck was being recorded.

6.3. Process of Creating a Deck

Each participant created three identical flashcard decks from the provided data set using each of the three methods: manual (Anki), semi-automatic (spreadsheet software), and automatic (Anki Deck Creator).

All participants performed the three methods in the same sequence: manual, semi-automatic, automatic. This introduces a potential order effect, which is acknowledged as a limitation of the study (Section 7.1).

6.4. Data Collection and Analysis

The time spent on flashcard creation was measured using an independent timing device. However, this functionality was not integrated into the spreadsheet software or Anki Deck Creator due to technical limitations of the Anki application itself. The Anki application does not currently support such measurement capabilities.

When calculating the time cost of creating flashcards, the duration of the preparatory phase related to the configuration of software for processing user input is not taken into account, since this indicator varies significantly depending on the level of user competence in working with the relevant tools.

In the manual method, the preparatory phase consists of pasting the test formatting code from the pre-defined vocabulary card template into the corresponding fields of the Anki application. The preparatory step in implementing the semi-automatic method consists of formulating a tabular data structure in which the number of rows corresponds to the expected number of flashcards in the deck. The cells of this structure contain formatting code for each flashcard field, in accordance with the prevailing flashcard template.

Furthermore, qualitative user experience testing was conducted to evaluate the usability and functional effectiveness of the considered deck creation methods.

User interviews were conducted to assess two key aspects of interaction with the system: the intuitiveness of the user interface and the degree of user satisfaction with the flashcard creation process.

7. STUDY RESULT INTERPRETATION AND ANALYSIS

7.1. Limitations of the Study

Several limitations of the present study should be acknowledged when interpreting its findings.

Sample size and its implications

The study involved 16 participants, which represents a relatively small sample for quantitative comparative research. While prior work in usability evaluation has demonstrated that small samples can be sufficient for identifying usability problems in qualitative assessments (Nielsen, 1993), Virzi (1992) notes that quantitative comparisons of performance metrics require larger samples to achieve adequate statistical power. Consequently, the numerical results reported in this study should be treated as indicative rather than conclusive, and replication with a larger sample is recommended.

Non-representative skill distribution

The participant pool was characterised by a pronounced overrepresentation of technically proficient users: 68.75% of respondents were classified as Senior or Expert level (Section 7.3), with no Junior-level participants. This distribution does not reflect the broader population of Estonian language learners, which may include users with limited technical skills. As Creswell (2014) notes, convenience sampling – the method employed in this study – frequently results in samples that are not representative of the target population, thereby limiting the generalisability of findings.

Possible order effect

All participants performed the three methods in the sequence: manual, semi-automatic, automatic. In within-subjects experimental designs, repeated exposure to similar tasks can introduce practice effects, whereby performance on later tasks is improved due to familiarity with the task structure rather than the method itself (Shadish et al., 2002). This may have contributed to the lower time scores observed for the automatic method, which was performed last.

Single language and single deck template

The application was developed and evaluated exclusively in the context of Estonian language learning, using a single, predefined flashcard template. The results therefore reflect the performance of the automatic method under these specific conditions and may not generalise to other languages or template structures. As noted in the Introduction section (“Objective of the Thesis”, p. 6), the application architecture supports the future addition of further decks, which would enable comparative evaluation across different linguistic contexts.

Statistical testing

A Wilcoxon signed-rank test was applied to the time measurements, confirming the statistical significance of the observed differences (Section 7.2.3). However, the analysis of user experience data (Sections 7.3.1—7.3.2) remains descriptive. The custom ordinal scales developed for this study (Appendix 3) have not been formally validated, and the sample size (n=16) further limits the applicability of inferential statistics to those measures

7.2. Estimating the Time to Create a Deck

The experimental study revealed significant differences in the temporal characteristics of the flashcard creation process when applying the three methodological approaches. The charts in Figure 23 (p. 41) and Figure 24 (p. 42) depict the time it took each study participant to create a deck using the different methods, as well as showing the average time required to complete this task within each method.

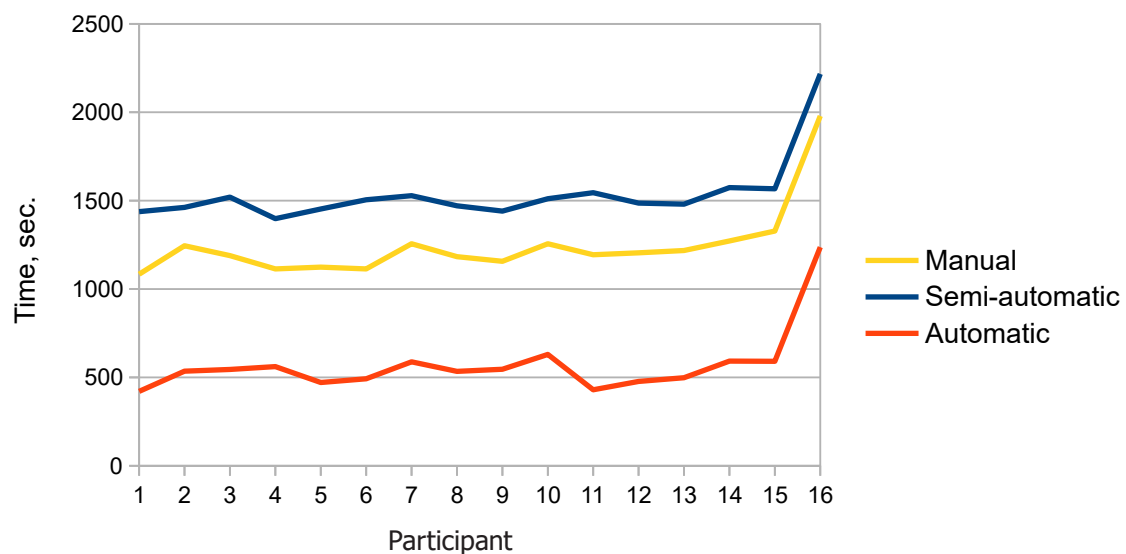


Figure 23. Total time to create a deck for each participant.

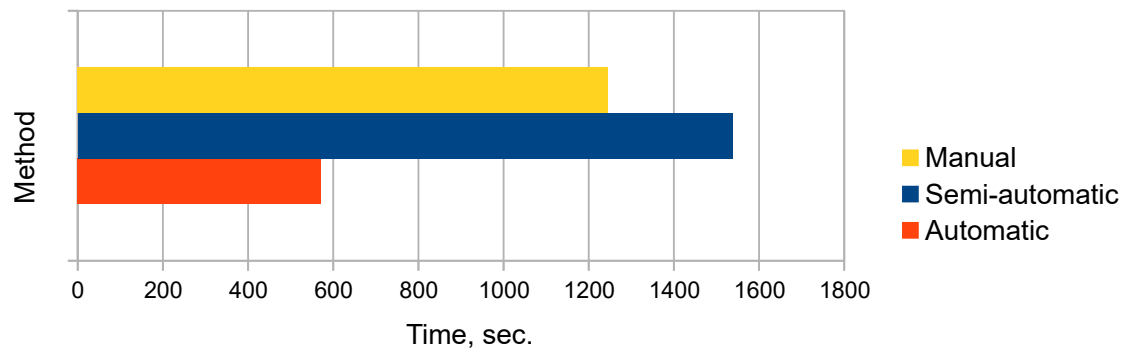


Figure 24. The mean time required to create a deck.

7.2.1. Manual Method

Average deck creation time: 1244.875 sec., 00:20:44.

This method is characterised by high levels of manual effort and time expenditure. It involves manual handling of flashcard data, which is prone to errors, and requires sustained user engagement with template formatting strings of the flashcard template fields, thereby increasing task completion time and cognitive load on the user.

7.2.2. Semi-automatic method

Average deck creation time: 1537.25 sec., 00:25:37.

Contrary to expectations, the automated approach exhibited a prolonged duration in comparison with the manual method. This can be attributed to the complexity of data preparation in the spreadsheet software and the necessity of executing numerous sequential operations. Of particular note is the probability of errors in this process is comparable to that of manual data entry, which underscores the need for rigorous attention to detail in both methods.

7.2.3. Automatic method

Average deck creation time: 571.75 sec., 00:09:31.

Within the scope of this study, the automatic method was determined to be the most time-efficient among the three approaches evaluated. The principal advantage of this approach is the minimization of manual intervention and a significant reduction of time costs relative to alternative methodological approaches. Among the study participants, the manual method required 117.73% more time and the semi-automatic method 168.88% more time to complete the task compared to the automatic method.

The observed reduction of 54.1% exceeds the 40% threshold stated in Hypothesis 1.

To assess the statistical significance of the observed differences, a Wilcoxon signed-rank test was applied to the paired time measurements ($n=16$). The test confirmed a statistically significant advantage of the automatic method over the manual method ($W=0.0$, $p=0.0000305$), and over the semi-automatic method ($W=0.0$, $p=0.000434$). The value $W=0.0$ indicates that in all 16 cases without exception, the automatic method required less time than either alternative. These results provide statistical support for Hypothesis 1.

Hypothesis 1 is therefore supported.

7.3. Evaluating user experience

User experience was assessed using a custom ordinal scale developed for this study; its scope and limitations are described in Appendix 3.

The distribution of respondents according to their computer competence level demonstrates a significant predominance of experienced users (Figure 25): precisely 11 respondents (68.75%) were classified as either Senior or Expert. A single case is represented by a Basic-level user, with a complete absence of representatives of the Junior category.

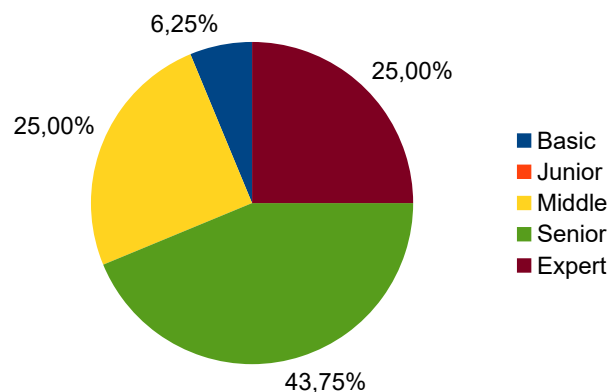


Figure 25. The distribution of respondents by computer usage experience.

7.3.1. Analyzing the intuitiveness of the user interface

Intuitiveness was assessed using a custom ordinal scale developed for this study; its scope and limitations are described in Appendix 3.

The provided diagrams – Figure 26 and Figure 27 on p. 44 – illustrate the findings from an evaluation of user interface intuitiveness across various deck creation methods. The first diagram displays the distribution of individual user ratings for each method, enabling an

analysis of the variability in perceived intuitiveness. The second diagram presents the mean scores for each method, offering a consolidated measure of their overall intuitiveness as assessed by users.

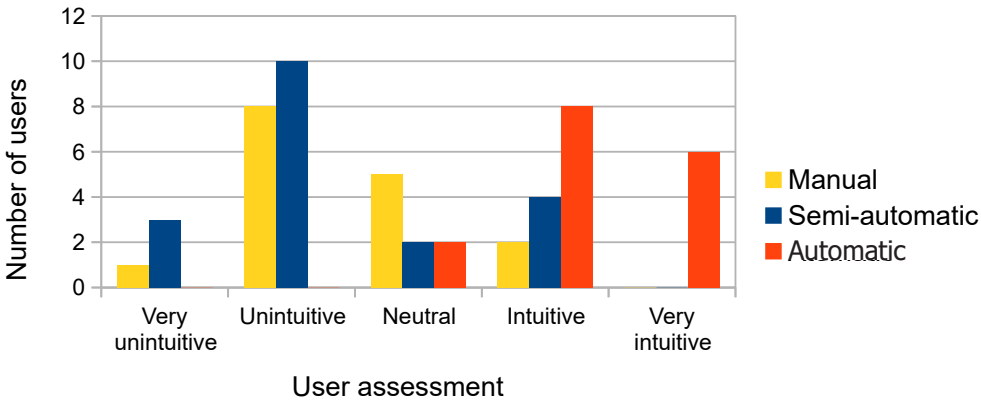


Figure 26. Evaluation of user interface intuitiveness.

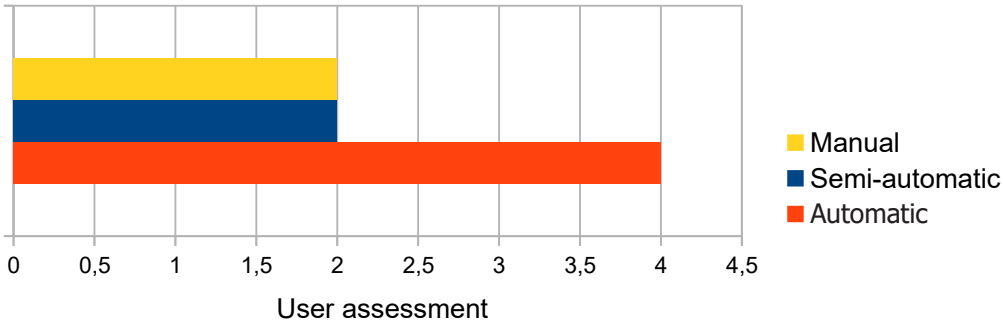


Figure 27. The mean evaluation of the intuitiveness of the user interface.

7.3.1.1. Manual method user interface

The interface posed challenges for the majority of users: Precisely 56.25% of respondents (9 respondents) characterized it as unintuitive. Conversely, a mere 12.5% of respondents (2 respondents) found the interface acceptable.

7.3.1.2. Semi-automatic Method User Interface

This interface received the least favourable ratings: 13 respondents (81.25%) characterised it as unintuitive. While 25% (4 respondents) offered positive ratings, this did not offset the predominance of negative evaluations.

7.3.1.3. Automatic Method User Interface

This method received the highest rating in terms of intuitiveness: specifically, 87.5% of respondents (14 out of 16) characterized the interface as intuitive. The absence of negative ratings further supports the effectiveness of the implemented approach.

7.3.2. Analysis of User Satisfaction

User satisfaction was assessed using a custom ordinal scale developed for this study; its scope and limitations are described in Appendix 3.

Figure 28 presents a distribution of users based on their satisfaction with the deck creation process for each method, while Figure 29 illustrates the mean user satisfaction score.

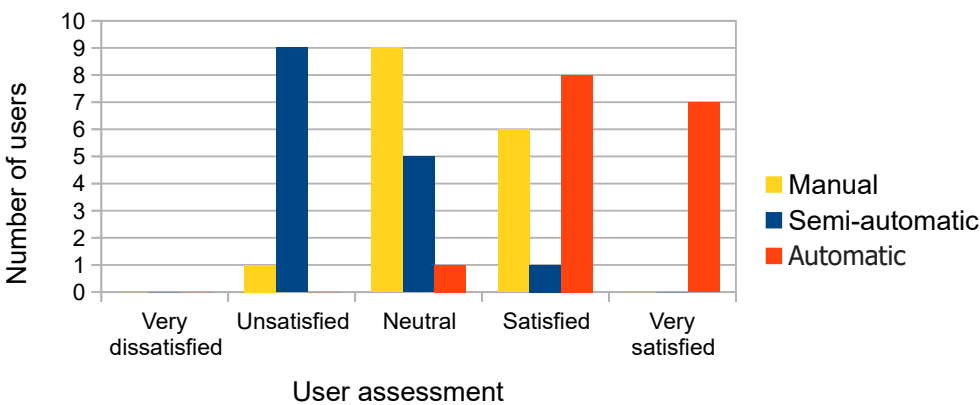


Figure 28. Evaluation of user satisfaction with the deck creation process.

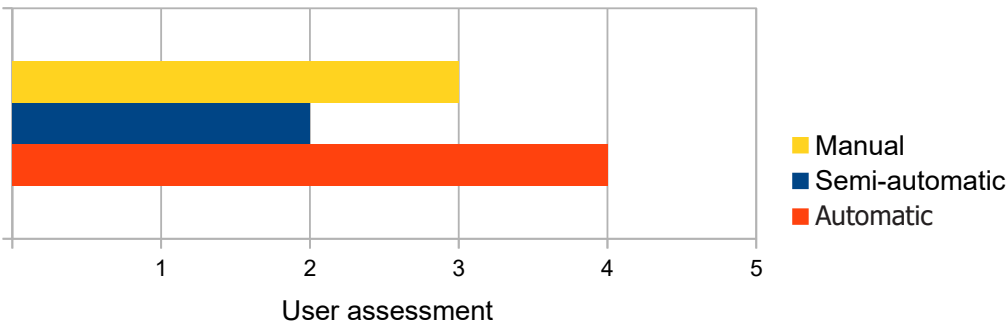


Figure 29. The average score of user satisfaction with the deck creation process.

7.3.2.1. Manual Method

The method is characterized by predominantly neutral evaluation: 56% of respondents (9 respondents) expressed a neutral attitude, while 37.5% (6 respondents) gave a positive evaluation. A single instance of dissatisfaction was recorded, amounting to 6.25% of the total. This method can be characterized as functionally acceptable, but not optimal.

7.3.2.2. Semi-Automatic Method

The method received a predominantly critical reception, with 56.25% of respondents (9 respondents) expressing dissatisfaction. Conversely, a mere 6.25% of respondents offered a favorable evaluation. These findings suggest that there are substantial deficiencies in the implementation of this approach.

7.3.2.3. Automatic Method

Within this study's sample, the automatic method received notably high user satisfaction ratings. A substantial majority of participants rated the method positively: 93.75% (15 out of 16) expressed satisfaction. Notably, 43.75% of participants (7 out of 16) selected the highest possible satisfaction rating. The complete absence of negative ratings further indicates the approach's effectiveness.

7.4. Result of the study

The findings of the analysis demonstrate a direct correlation between the time required to create the deck and the mastery of digital information processing tools, such as the ability to work with text and a spreadsheet processor. A particularly salient factor complicating the process of working with the spreadsheet software was the lack of functionality for syntax highlighting of HTML tags of the flashcard field formatting, which necessitated additional cognitive and temporal resources from users for preliminary adaptation of the visual data structure (using colour-coding and cell formatting styles). Conversely, Anki tools with integrated formatting support enhance the user experience by reducing the time-consuming manual manipulation of pre-formatting the word card field code.

Analysis of the observational data revealed the predominant cognitive challenges encountered by participants. The predominant error observed pertained to the erroneous placement of information blocks within the flashcard field structure. Users inserted some data into the fields intended for other data. It is noteworthy that the prior familiarity of some participants with the principles of HTML markup enabled them to adapt more rapidly and to focus more intently on the content aspects of the work.

The results of the empirical study indicate a consistent superiority of the automatic method across all investigated parameters among the participants of this study. The ergonomic design of the user interface and the high satisfaction scores of the automatic method demonstrate a significant reduction in the cognitive and physical load on users, thereby supporting Hypothesis 2. The substantial difference in quantitative scores (approximately 2 points) between the automatic and semi-automatic methods suggests that the semi-automatic approach requires fundamental revision or discontinuation.

CONCLUSION

The present thesis is primarily applied in nature, contributing a working software tool and an empirical comparison of deck creation methodologies, rather than advancing theoretical knowledge in the fields of HCI or educational technology.

Specifically, Hypothesis 1 was supported by a 54.1% reduction in deck creation time, and Hypothesis 2 was supported by consistently high usability ratings for the automatic method.

The results not only support the initial hypotheses of the study, but also demonstrate the significant potential of automating the process of creating flashcards. Within this study, the application demonstrated effectiveness in reducing task completion time and cognitive load among participants.

The present study involved a diverse sample of participants with varying levels of computer competence and experience with the Anki application. Despite the variation in skill levels, the automatic method demonstrated a consistent advantage in flashcard creation efficiency.

The findings provide a basis for recommending the application as an effective tool for creating learning materials for Estonian language learning using the principle of spaced repetition, particularly when regular preparation of large flashcard sets is required.

In light of the study's findings, the following recommendations can be made:

1. Adopt the application in educational practice for Estonian language learning.
2. Refine the automation algorithms to further improve flashcard creation efficiency.
3. Conduct additional research to validate the results with a larger and more representative participant sample.
4. Conduct dedicated usability testing with entry-level (Basic) users to assess accessibility across all proficiency levels.

Future research could focus on extending the application's functionality and adapting it for additional languages.

REFERENCES

- Ankitects Pty Ltd. (n.d.). *What spaced repetition algorithm does Anki use?* <https://faqs.ankiweb.net/what-spaced-repetition-algorithm> (accessed April 27, 2025)
- AvaloniaUI. (2024). *Avalonia documentation*. <https://docs.avaloniaui.net/> (accessed April 27, 2025)
- Bray, T. (2017). *The JavaScript Object Notation (JSON) Data Interchange Format* (RFC 8259). Internet Engineering Task Force. <https://www.rfc-editor.org/rfc/rfc8259>
- Brooke, J. (1996). *SUS: A “quick and dirty” usability scale*. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, & I. L. McClelland (Eds.), *Usability evaluation in industry* (pp. 189—194). Taylor & Francis.
- Cepeda, N. J., Pashler, H., Vul, E., Wixted, J. T., & Rohrer, D. (2006). Distributed practice in verbal recall tasks: A review and quantitative synthesis. *Psychological Bulletin*, 132(3), 354–380. <https://doi.org/10.1037/0033-2909.132.3.354> (accessed April 27, 2025)
- Creswell, J. W. (2014). *Research design: Qualitative, quantitative, and mixed methods approaches* (4th ed.). SAGE Publications.
- Ebbinghaus, H. (2003). Memory: A contribution to experimental psychology. In M. Wertheimer & C. W. King (Eds.), *The history of psychology: Fundamental questions* (pp. 168–177). Thomson/Wadsworth.
- Field, A. (2013). *Discovering statistics using IBM SPSS statistics* (4th ed.). SAGE Publications.
- Hall, R. H., & Hanna, P. (2004). The impact of web page text-background colour combinations on readability, retention, aesthetics and behavioural intention. *Behaviour and Information Technology*, 23(3), 183–195. <https://doi.org/10.1080/01449290410001669932> (accessed April 27, 2025)
- International Institute of Business Analysis. (2015). *BABOK: A guide to the business analysis body of knowledge* (3rd ed.). <https://www.iiba.org/babok-guide> (accessed April 27, 2025)
- ISO/IEC. (2023). *ISO/IEC 25010:2011 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*. International Organization for Standardization. <https://www.iso.org/standard/78176.html> (accessed April 27, 2025)
- Karpicke, J. D., & Roediger, H. L. (2008). *The critical importance of retrieval for learning*. *Science*, 319(5865), 966–968. <https://doi.org/10.1126/science.1152408> (accessed April 27, 2025)

- Kılıçkaya, F. (2014). Technology & language learning. *AATSEEL Newsletter*, 57(3), 12–14.
- Kornell, N. (2009). Optimising learning using flashcards: Spacing is more effective than cramming. *Applied Cognitive Psychology*, 23(9), 1297–1317. <https://doi.org/10.1002/acp.1523> (accessed April 27, 2025)
- Microsoft. (2023). *.NET documentation*. Microsoft Learn. <https://learn.microsoft.com/en-us/dotnet/> (accessed April 27, 2025)
- Microsoft. (2023). *Welcome to C# 12*. Microsoft Learn. <https://learn.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-12> (accessed April 27, 2025)
- Nakata, T. (2011). Computer-assisted second language vocabulary learning in a paired-associate paradigm: A critical investigation of flashcard software. *Computer Assisted Language Learning*, 24(1), 17–38. <https://doi.org/10.1080/09588221.2010.520675> (accessed April 27, 2025)
- Negi, A. S. & Vaidya, H. & Chauhan, R. (2023) *Design and Development of a LMS with MVVM Architecture*, 2023 International Conference on Sustainable Emerging Innovations in Engineering and Technology (ICSEIET), Ghaziabad, India, pp. 820-823, doi: 10.1109/ICSEIET58677.2023.10303333 (accessed April 27, 2025)
- Nielsen, J. (1993). *Usability engineering*. Morgan Kaufmann.
- Norman, D. A. (2013). *The design of everyday things* (Revised and expanded ed.). The MIT Press.
- Schafer, S. M. (2005). *Web standards programmer's reference: HTML, CSS, JavaScript, Perl, Python, and PHP*. Wrox.
- Seibert Hanson, A. E., & Brown, C. M. (2019). Enhancing L2 learning through a mobile assisted spaced-repetition tool: An effective but bitter pill? *Computer Assisted Language Learning*, 33(1–2), 1–23. <https://doi.org/10.1080/09588221.2018.1552975> (accessed April 27, 2025)
- Shadish, W. R., Cook, T. D., & Campbell, D. T. (2002). *Experimental and quasi-experimental designs for generalized causal inference*. Houghton, Mifflin and Company.
- Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson.
- SuperMemo World. (n.d.). *SuperMemo algorithm*. https://help.supermemo.org/wiki/SuperMemo_Algorithm (accessed April 27, 2025)
- Surapaneni, K. M. (2023). Factors to consider while implementing spaced repetition software applications like Anki as a learning tool in medical education. *Medical Science Educator*, 33, 1297. <https://doi.org/10.1007/s40670-023-01864-2> (accessed April 27, 2025)

Valdez, P., & Mehrabian, A. (1994). Effects of color on emotions. *Journal of Experimental Psychology: General*, 123(4), 394–409. <https://doi.org/10.1037/0096-3445.123.4.394> (accessed April 27, 2025)

Virzi, R. A. (1992). *Refining the test phase of usability evaluation: How many subjects is enough?* *Human Factors*, 34(4), 457–468. <https://doi.org/10.1177/001872089203400407> (accessed April 27, 2025)

Wharton, Ch. & Wozniak, P.A. (1994). SuperMemo. *CALICO Journal*. Vol. 12, No 1, p. 66-70.

Wiegers, K., & Beatty, J. (2013). *Software requirements* (3rd ed.). Microsoft Press.

Wozniak, P. A. (1990). *Optimization of learning* [Master's thesis, University of Poznan].

APPENDICES

Appendix 1. Abbreviations and Glossary

Abbreviations

1. API – Application Programming Interface
2. CSS – Cascading Style Sheets
3. CSV – Comma-Separated Values
4. GUI – Graphical User Interface
5. JSON – JavaScript Object Notation
6. FSRS – Free Spaced Repetition Scheduler
7. HCI – Human Computer Interaction
8. HTML – HyperText Markup Language
9. IEC – International Electrotechnical Commission
10. ISO – International Organization for Standardization
11. MVVM – Model-View-ViewModel
12. REST – Representational State Transfer
13. SRS – Spaced Repetition Scheduler
14. UI – User Interface
15. UX – User Experience
16. XAML – eXtensible Application Markup Language
17. WPF – Windows Presentation Foundation

Glossary

1. *ДА-инфинитив* – DA-infinitive
2. *МА-инфинитив* – MA-infinitive
3. *Выберите тип колоды* – Choose deck type
4. *Выход* – Exit
5. *Глагол* – Verb
6. *Глагол составной* – Compound verb
7. *Двойной щелчок для правки карточки* – Double-click to edit card
8. *Добавить* – Add
9. *Дословный перевод* – Literal translation
10. *Единственное число* – Singular
11. *Изучаемое слово или выражение* – Word or phrase being studied
12. *Изъявительное наклонение (3 лицо, прош.)* – Indicative mood (3rd person, past)
13. *Именительный падеж* – Nominative case
14. *Контекст* – Context
15. *Короткая форма иллатива* – Short illative form
16. *Литературный перевод* – Free translation
17. *Маркер* – Highlight marker
18. *Множественное число* – Plural
19. *Наречие* – Adverb

20. *Новая* – New
21. *Оригинал* – Original
22. *Партитив* – Partitive
23. *Повелительное наклонение (ед.ч.)* – Imperative mood (singular)
24. *Послелог* – Postposition
25. *Предлог* – Preposition
26. *Причастие (действ. прош.)* – Participle (active, past)
27. *Причастие (страда., прош.)* – Participle (passive, past)
28. *Прилагательное* – Adjective
29. *Родительный падеж* – Genitive case
30. *Словарная статья* – Dictionary entry
31. *Список карточек* – Card list
32. *Страдательный залог (наст.)* – Passive voice (present)
33. *Существительное* – Noun
34. *Удалить* – Delete
35. *Управление частью речи* – Grammatical government
36. *Фильтр* – Filter
37. *Формы глагола* – Verb forms
38. *Формы слова* – Word forms
39. *Часть речи* – Part of speech
40. *Экспорт* – Export

Appendix 2. Overview of the formatting code for the flashcard template

The code presented in this appendix is part of the Anki application and defines the visual layout and appearance of flashcards. A simplified version of the code is provided below.

The HTML and CSS syntax used follows established web standards (Schafer, 2005).

The Front Side

```
<div class="front-container">
  <p class="section-description"> Дословный перевод: </p>
  {{Front}}
</div>
```

The Back Side

```
<div class="front-container">
  <p class="section-description"> Дословный перевод: </p>
  {{Front}}
</div>
<div class="translation-container">
  <p class="section-description"> Литературный перевод: </p>
  {{Translation}}
</div>
<div class="translation-container">
  <p class="section-description"> Оригинал: </p>
  {{Original}}
</div>
<div class="speech-part-container">
  <p class="section-description"> Часть речи: </p>
  <div class="speech-part"> {{SpeechPart}} </div>
</div>
<div class="forms-container">
  <p class="section-description"> Формы слова: </p>
  {{Forms}}
</div>
<div class="verb-control-container">
  <p class="section-description"> Управление глаголом: </p>
  {{VerbControl}}
</div>
```

The Styles Table

```
.card {
  font-family: "Roboto Slab", "Noto Serif";
  font-size: 20px;
  text-align: center;
  color: black;
  background-color: red;
}
.front-container, .translation-container, .word-forms-container {
  margin-bottom: 16px;
}
.section-description {
  text-align: center;
  font-size: 0.6em;
  color: #665C54;
}
.sentence span { color: #D79921; }
.sentence i {
  color: #D3869B;
  font-style: normal;
}
.word-forms-container {
  display: grid;
  column-gap: 8px;
  grid-template-columns:
    max-content max-content max-content max-content;
  justify-content: center;
}
.grid-item {
  padding: 2px;
  margin-right: 16px;
  color: black;
  text-align: left;
}
.grid-item.form, .grid-item.form-short-into { color: #D5C4A1; }
```

Appendix 3. Evaluation Scales

The following scales were developed specifically for this study. They serve two distinct purposes.

The Computer Expertise scale is a classificatory instrument used to characterise participants' self-reported level of technical proficiency. It is an ordinal scale comprising five levels and was employed solely to describe the skill distribution of the participant pool (Section 7.3), not to evaluate the deck creation methods. As a self-report measure, it reflects participants' own assessment of their competence rather than an independently verified skill level.

The Degree of Satisfaction and Degree of Intuitiveness scales are custom five-point ordinal instruments designed to capture participants' subjective experience of the deck creation process and the user interface respectively. As self-report measures, they reflect perceived experience rather than objective behavioural indicators such as error rates or time-to-first-success. Neither scale has been formally validated; their use is therefore exploratory in nature. In particular, the intuitiveness scale represents a single self-reported dimension of a multi-faceted construct that in validated instruments – such as the System Usability Scale (Brooke, 1996) – is typically assessed through multiple items. Future studies are encouraged to employ validated usability instruments to enable cross-study comparison.

Computer Expertise

This scale assesses the user's level of proficiency in working with computer programs and devices.

Basic – The user is in the early stages of computer use. Their knowledge is limited to fundamental actions: turning the device on, using a mouse and keyboard, and accessing basic applications such as web browsers and text editors. Ongoing assistance is typically required.

Junior – The user can operate basic software applications, including text editors, web browsers, and email clients, and perform simple tasks such as creating documents, using social media, and searching the internet. More complex tasks require prompts or guidance.

Middle – The user confidently uses computers for a range of everyday tasks, including file management, configuring programs and system settings, and using advanced software features. While occasional difficulties may arise, the user is generally able to resolve them independently.

Senior – The user has a thorough understanding of operating systems and can customise programs and system settings, troubleshoot technical issues, and optimise computer performance. The user is comfortable multitasking across multiple applications and performing advanced operations such as creating backups and configuring security settings.

Expert – The user demonstrates a high level of proficiency across specialised programs and systems. They are capable of resolving complex technical issues, optimising system performance, and applying computers to advanced tasks such as programming and data analysis. They frequently provide guidance and training to other users.

The Degree of Satisfaction with the Flashcard Creation Process

This scale assesses the user's perceived convenience and effectiveness of the flashcard creation process.

Very dissatisfied (1) – The flashcard creation process was complicated, inconvenient, and time-consuming, with frequent difficulties.

Unsatisfied (2) – The process was cumbersome and presented notable challenges, though the task was ultimately completed.

Neutral (3) – The process was neither particularly convenient nor inconvenient. No significant difficulties were encountered, but no notable comfort was perceived either.

Satisfied (4) – The flashcard creation process was efficient, with no significant issues encountered.

Very satisfied (5) – The process was fast and intuitive, with no obstacles encountered. The overall experience was positive.

The Degree of Intuitiveness of the User Interface

This scale assesses the ease with which the user interface can be understood and used without additional instructions.

Very unintuitive (1) – The interface was difficult to understand and required frequent assistance or guidance.

Unintuitive (2) – The interface was initially difficult to understand; however, after several attempts, the user was able to grasp its functionality.

Neutral (3) – The interface was not particularly challenging, but lacked clear visual cues for immediate comprehension.

Intuitive (4) – The interface was simple and intuitive, facilitating ease of use.

Very intuitive (5) – The interface was entirely intuitive, with all functions readily apparent at first glance, requiring no explanation.

Appendix 4. Test Data Set

Study participants were provided with the following datasheets to facilitate the formation of decks.

Карточка 1	
Изучаемое слово или выражение	küljendama
Часть речи	глагол
МА-инфинитив	Изъявительное наклонение (3 лицо, прош.)
küljendada	küljendas
ДА-инфинитив	Причастие (действ. прош.)
küljendada	küljendanud
Изъявительное наклонение (3 лицо)	Повелительное наклонение (ед.ч.)
küljendab	küljenda
Причастие (страда., прош.)	Страдательный залог (наст.)
küljendatud	küljendatakse
Дословный перевод	Сверстанный материал остался(üv) [в отсутствие времени] [не прочитан].
Литературный перевод	Из-за нехватки времени сверстанный материал не был прочитан.
Оригинал	Küljendatud materjal jäi ajapuudusel üle lugemata.
Карточка 2	
Изучаемое слово или выражение	kätte maksma
Часть речи	глагол составной
МА-инфинитив	Изъявительное наклонение (3 лицо, прош.)
kätte maksma	maksis kätte
ДА-инфинитив	Причастие (действ. прош.)
kätte maksta	kätte maksnud
Изъявительное наклонение (3 лицо)	Повелительное наклонение (ед.ч.)
maksab kätte	maksa kätte
Причастие (страда., прош.)	Страдательный залог (наст.)
kätte makstud	makstakse kätte
Управление частью речи	kellele; mille eest
Дословный перевод	Играл Амстердамом-названного мальчиком, кто платит [в руки] своего отца смерть за.
Литературный перевод	Играл мальчика по имени Амстердам, который мстит за смерть своего отца.
Оригинал	Mängin Amsterdami-nimelist poissi, kes maksab kätte oma isa surma eest.

Карточка 3		
Изучаемое слово или выражение		seletus
Часть речи		существительное
	Единственное число	Множественное число
Именительный падеж	seletus	seletused
Родительный падеж	seletuse	seletuste
Партитив	seletust	seletusi
Короткая форма иллатива		seletusse
Дословный перевод	Мужчина попросил [с него] письменное объяснение.	
Литературный перевод	Мужчина попросил от него письменное объяснение.	
Оригинал	Mees palus talt kirjalikku seletust.	
Карточка 4		
Изучаемое слово или выражение		sarvik
Часть речи		существительное
	Единственное число	Множественное число
Именительный падеж	sarvik	sarvikud
Родительный падеж	sarviku	sarvikute
Партитив	sarvikut	sarvikuid
Короткая форма иллатива		
Дословный перевод	[Сына Калева] битва [с бесами].	
Литературный перевод	Битва Калева с бесами.	
Оригинал	Kalevipoja võitlus sarvikuga.	
Карточка 5		
Изучаемое слово или выражение		mitmetähenduslikkus
Часть речи		существительное
	Единственное число	Множественное число
Именительный падеж	mitmetähenduslikkus	mitmetähenduslikkused
Родительный падеж	mitmetähenduslikkuse	mitmetähenduslikkuste
Партитив	mitmetähenduslikkust	mitmetähenduslikkusi
Короткая форма иллатива		mitmetähenduslikkusse
Дословный перевод	Ситуации неоднозначность было трудно недооценить.	
Литературный перевод	Неоднозначность ситуации трудно было недооценить.	
Оригинал	Olukorra mitmetähenduslikkust oli raske alahinnata.	
Карточка 6		
Изучаемое слово или выражение		üllatavalt
Часть речи		прилагательное
	Единственное число	Множественное число
Именительный падеж	üllatavalt	
Родительный падеж		
Партитив		

Короткая форма илллатива			
Дословный перевод	удивительно		
Литературный перевод			
Оригинал			
Карточка 7			
Изучаемое слово или выражение		koorem	
Часть речи		существительное	
	Единственное число	Множественное число	
Именительный падеж	koorem	koormad	
Родительный падеж	koorma	koormate	
Партитив	koormat	koormaid	
Короткая форма илллатива			
Дословный перевод	Мальчик был [на себя] слишком большой груз спину загрузил.		
Литературный перевод	Мальчик взвалил на себя слишком большую ношу.		
Оригинал	Poiss oli endale liiga suure koorma selga ladunud.		
Карточка 8			
Изучаемое слово или выражение		kaebama	
Часть речи		глагол	
МА-инфинитив		Изъявительное наклонение (3 лицо, прош.)	
kaebama		kaebas	
ДА-инфинитив		Причастие (действ. прош.)	
kaevata		kaevanud	
Изъявительное наклонение (3 лицо)		Повелительное наклонение (ед.ч.)	
kaebab		kaeba	
Причастие (страда., прош.)		Страдательный залог (наст.)	
kaevatud		kaevatakse	
Управление частью речи		mille üle; mida; et; kelle; kuhu; kelle peale; kellele	
Дословный перевод	Пациент жаловался плохое самочувствие на.		
Литературный перевод	Пациент жаловался на плохое самочувствие.		
Оригинал	Patsient kaebas halva enesetunde üle.		
Карточка 9			
Изучаемое слово или выражение		täiendama	
Часть речи		глагол	
МА-инфинитиф		Изъявительное наклонение (3 лицо, прош.)	
täiendama		täiendas	
ДА-инфинитив		Причастие (действ. прош.)	
täiendada		täiendanud	

Изъявительное наклонение (3 лицо)		Повелительное наклонение (ед.ч.)
täiendab		täienda
Причастие (страда., прош.)		Страдательный залог (наст.)
täiendatud		täiendakse
Управление частью речи		mida + millega
Дословный перевод	Закон дополнили новым параграфом.	
Литературный перевод	В закон был добавлен новый пункт.	
Оригинал	Seadust täiendati uue paragrahviga.	
Карточка 10		
Изучаемое слово или выражение		edendama
Часть речи		глагол
МА-инфинитив		Изъявительное наклонение (3 лицо, прош.)
edendama		edendas
ДА-инфинитив		Причастие (действ. прош.)
edendada		edendanud
Изъявительное наклонение (3 лицо)		Повелительное наклонение (ед.ч.)
edendab		edenda
Причастие (страда., прош.)		Страдательный залог (наст.)
edendatud		edendatakse
Управление частью речи		mida
Дословный перевод	Как развивать эмоциональную интеллектность?	
Литературный перевод	Как развивать эмоциональную интеллектность?	
Оригинал	Kuidas edendada emotsionaalset intelligentsust?	

Appendix 5. Example of CSV File Contents of Exported Deck

A fragment of a text file generated by the application and imported into Anki. The lines are numbered. Long lines are wrapped and indicated with the ‘↵’ symbol.

The excerpt shows the file header (lines 1 and 2), containing the card field delimiter specification (the ‘|’ character) and a directive instructing Anki to apply HTML formatting to card fields; and one flashcard’s data (line 3).

1. #separator:|
2. #html:true
3. “<div class=“”sentence“”>Брат дал ключи мои [в ↵
руки].</div>”|”<div class=“”sentence“”>Брат дал мне ↵
ключи.</div>”|”<div class=“”sentence“”>Vend andis ↵
võtmed minu kätte.</div>”|”Глагол ■ Tegusõna”|”<div ↵
class=“”word-forms-container“”><div class=“”grid-item declension“” ↵
>MA</div><div class=“”grid-item form“”>andma</div><div class=“”griditem ↵
declension“”>3P.MIN</div><div class=“”grid-item form“”>andis</div><div ↵
class=“”grid-item declension“”>DA</div><div class=“”grid-item form“”> ↵
anda</div><div class=“”grid-item declension“”>NUD</div><div class=“”grid-↵
item form“”>andnud </div><div class=“”grid-item declension“”>3P</div> ↵
<div class=“”grid-item form“”>annab</div><div class=“”grid-item declension ↵
“”>2P.KÄS</div><div class=“”grid-item form“”>anna</div><div class=“”grid-↵
item declension“”>TUD</div><div class=“”grid-item form“”>antud ↵
</div><div class=“”grid-item declension“”>AKSE</div> <div class=“”grid-↵
item form“”>antakse</div></div>”|””|”andma”