# Extending Diamond: A Web Application for Tree Testing and Card Sorting

Mohamed Amine El Kaouakibi

# Extending Diamond: A Web Application for Tree Testing and Card Sorting

Mohamed Amine El Kaouakibi

**Bachelor's Thesis**

to achieve the university degree of

Bachelor of science

Bachelor's Degree Programme: Computer Science

submitted to

Graz University of Technology

Supervisor

Ao.Univ.-Prof. Dr. Keith Andrews
Institute of Interactive Systems and Data Science (ISDS)

Graz, 27 Sep 2023

# Ausbau von Diamond: Eine Webanwendung für Baumtests und Kartensortierung

Mohamed Amine El Kaouakibi

**Bachelorarbeit**

für den akademischen Grad

Bachelor

Bachelorstudium: Informatik

an der

Technischen Universität Graz

Begutachter

Ao.Univ.-Prof. Dr. Keith Andrews
Institute of Interactive Systems and Data Science (ISDS)

Graz, 27 Sep 2023

Diese Arbeit ist in englischer Sprache verfasst.

## Statutory Declaration

*I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The document uploaded to TUGRAZonline is identical to the present thesis.*

## Eidesstattliche Erklärung

*Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Dokument ist mit der vorliegenden Arbeit identisch.*

Date/Datum

Signature/Unterschrift

# Abstract

Information architecture (IA) seeks to organize and label content for a web site or application using information hierarchies and metadata. Two of the main techniques in IA involve testing with representative test users, namely card sorting and tree testing. In card sorting, users are asked to sort concepts into groups and then give the groups names. This helps the information architect to craft a corresponding information hierarchy. In tree testing, a proposed information hierarchy is presented to test users and they are asked to find particular items in the hierarchy.

Diamond is a free, open-source web application for card sorting and tree testing. It is written in TypeScript using the MEAN stack (MongoDB, Express.js, Angular, Node). Diamond is designed to be self-hosted by potential study owners. It provides the essential functionality required to set up, carry out, and assess the outcomes of card sorting and tree testing studies.

# Kurzfassung

Informationsarchitektur (IA) zielt darauf ab, Inhalte für eine Website oder Anwendung mithilfe von Informationshierarchien und Metadaten zu organisieren und zu kennzeichnen. Zwei der Haupttechniken in der IA beinhalten Tests mit repräsentativen Testbenutzern, nämlich Card Sorting und Tree Testing. Beim Card Sorting werden Benutzer gebeten, Konzepte in Gruppen zu sortieren und diesen Gruppen Namen zu geben. Dies hilft dem Informationsarchitekten dabei, eine entsprechende Informationshierarchie zu erstellen. Beim Tree Testing wird eine vorgeschlagene Informationshierarchie Testbenutzern präsentiert, die gebeten werden, bestimmte Elemente in der Hierarchie zu finden.

Diamond ist eine kostenlose Open-Source-Web Anwendung für Card Sorting und Tree Testing. Sie ist in TypeScript geschrieben und verwendet den MEAN-Stack (MongoDB, Express.js, Angular, Node). Diamond ist so konzipiert, dass sie von potenziellen Studienleitern selbst gehostet werden kann. Sie bietet die grundlegende Funktionalität, die für die Einrichtung, Durchführung und Auswertung von Card Sorting- und Tree Testing-Studien erforderlich ist.

# Contents

**Contents** ii

**List of Figures** iv

**List of Listings** v

**Acknowledgements** vii

**Credits** ix

**1 Introduction** 1

**2 Information Architecture** 3
- 2.1 Card Sorting . . . 3
- 2.2 Online Card Sorting Tools . . . 5
  - 2.2.1 OptimalSort [2004] . . . 5
  - 2.2.2 Proven By Users [2017] . . . 7
  - 2.2.3 Maze [2018] . . . 7
  - 2.2.4 PlaybookUX [2018] . . . 7
  - 2.2.5 UXtweak [2019] . . . 7
  - 2.2.6 KardSort [2020] . . . 9
- 2.3 Tree Testing . . . 10
- 2.4 Online Tree Testing Tools . . . 11
  - 2.4.1 Treejack [2008] . . . 11
  - 2.4.2 PlaybookUX [2018] . . . 11
  - 2.4.3 UXtweak [2019] . . . 11

**3 Modern Web Technology** 15
- 3.1 Responsive Web Design . . . 15
- 3.2 Single Page Applications . . . 15
- 3.3 MEAN Stack . . . 17

**4 Diamond Version 1.0** 19
- 4.1 Limitations . . . 19
- 4.2 Capabilities . . . 19

**5 Extending Diamond** **27**

5.1 Global Changes . . . . . . . . . . . . . . . . . . . . . . . . . . 27

5.2 Card Sorting . . . . . . . . . . . . . . . . . . . . . . . . . . 29

5.3 Tree Testing . . . . . . . . . . . . . . . . . . . . . . . . . . 31

**6 Future Work** **35**

**A Card Sorting Study Export Format** **37**

**B Tree Testing Study Export Format** **41**

**Bibliography** **45**

# List of Figures

2.1 IA in a Web Site . . . . . 4

2.2 Open Card Sorting . . . . . 4

2.3 OptimalSort: Setting up a Card Sorting Study . . . . . 6

2.4 OptimalSort: Sorting Cards . . . . . 6

2.5 Proven By Users: Sorting Cards . . . . . 7

2.6 Maze: Sorting Cards . . . . . 8

2.7 PlaybookUX: Sorting Cards . . . . . 8

2.8 UXtweak: Sorting Cards . . . . . 9

2.9 KardSort: Sorting Cards. . . . . . 10

2.10 Treejack: Setting up a Tree Test Study. . . . . . 11

2.11 Treejack: Doing a Tree Test . . . . . 12

2.12 PlaybookUX: Doing a Tree Test . . . . . 12

2.13 PlaybookUX: Analyzing a Tree Test Study . . . . . 13

2.14 UXtweak: Doing a Tree Test . . . . . 14

3.1 Responsive Web Site . . . . . 16

3.2 SPA Lifecycle . . . . . 17

3.3 MongoDB Compass . . . . . 18

4.1 Diamond v1.0: Study Owner Landing Page . . . . . 20

4.2 Diamond v1.0: Creating a Tree Testing Study . . . . . 21

4.3 Diamond v1.0: Creating a Tree for a Tree Testing Study . . . . . 21

4.4 Diamond v1.0: Defining Tasks for a Tree Testing Study . . . . . 22

4.5 Diamond v1.0: Results Page for a Tree Testing Study . . . . . 22

4.6 Diamond v1.0: Participants in a Tree Testing Study . . . . . 23

4.7 Diamond v1.0: Participants of a Card Sorting Study. . . . . . 23

4.8 Diamond v1.0: Results of One Participant in a Card Sorting Study . . . . . 24

4.9 Diamond v1.0: Task Analysis Overview for a Tree Testing Study . . . . . 24

4.10 Diamond v1.0: Path Tree for a Tree Testing Task . . . . . 25

4.11 Diamond v1.0: Destinations Table for a Tree Testing Study . . . . . 25

5.1 Error Page Before and After . . . . . 28

5.2 CardSort Studies Page Before and After . . . . . 30

5.3 CardSort Overview Excluded Participants . . . . . . . . . . . . . . . . . . . . . . 31
5.4 TreeTest Studies Page Before and After . . . . . . . . . . . . . . . . . . . . . . 32
5.5 TreeTest Overview Excluded Participants . . . . . . . . . . . . . . . . . . . . . . 33
5.6 Skipped Tasks in Destinations Table . . . . . . . . . . . . . . . . . . . . . . 33
5.7 Absolute Numbers of Tests for Task Analysis . . . . . . . . . . . . . . . . . . . . 34

# List of Listings

A.1 Card Sorting Study Export Schema . . . . . . . . . . . . . . . . . . . . . . . 38
A.2 Card Sorting Study Export Example . . . . . . . . . . . . . . . . . . . . . . . 39

B.1 Tree Testing Study Export Schema . . . . . . . . . . . . . . . . . . . . . . . 42
B.2 Tree Testing Study Export Example . . . . . . . . . . . . . . . . . . . . . . . 43

# Acknowledgements

I am profoundly grateful to my advisor, Keith Andrews, for his exceptional guidance, his endless hours in supporting me, and his insightful feedback throughout the entirety of this thesis. His expertise, patience, and tireless dedication to reviewing multiple drafts of this thesis have been immensely valuable in shaping the direction of this work.

I extend my heartfelt appreciation to my fiance, Manar, whose understanding and patience during long hours of work were a constant source of strength.

Last but not least, I am indebted to my family for their unending love, their sacrifices and their continuous support during all my period as a student. And my friends, for their support.

Mohamed Amine El Kaouakibi

Graz, Austria, 27 Sep 2023

# Credits

I would like to thank the following individuals and organizations for permission to use their material:

- The thesis was written using Keith Andrews' skeleton thesis [Andrews 2021].
- Figure 2.1 was adapted by Keith Andrews from Figure 2-2 of Spencer [2014, page 28], and used with kind permission of Keith Andrews.
- Figure 2.2 was adapted by Keith Andrews from Figure 26 of Schilb [2006, page 40] and used with kind permission of Keith Andrews.

# Chapter 1

# Introduction

Diamond is an open-source web application for card sorting and tree testing [Diamond 2023]. This thesis describes various extensions and improvements which were implemented for Diamond. The work on this thesis started by analyzing some limitations that Diamond has and explaining more how its functionalities were improved to cover some of those limitations.

In Chapter 2, the thesis first explains the concepts of information architecture, and in particular card sorting and tree testing. Chapter 3 gives an overview of some of the technologies used in modern web development, including those used in the development of Diamond. Chapter 4 describes the Diamond project, including its history and status before commencement of this work.

The extensions and improvements implemented for Diamond during this thesis work are presented in Chapter 5. In particular, facilities to export and import Diamond studies were implemented. Chapter 6 presents potential future work that could be done to further improve Diamond. Finally, two appendices explain the formats used to export and import Diamond studies: Appendix A specifies the export format for a card sorting study, while Appendix B specifies the export format for a tree testing study.

# Chapter 2

# Information Architecture

Information architecture (IA) is “a professional practice and field of studies focused on solving the basic problems of accessing, and using, the vast amounts of information available today.” [Resmini and Rosati 2011]. It is part of the broader field of user experience (UX). Information architecture involves organizing and labeling content using information hierarchies and metadata. Information hierarchies group and label content to support navigation, while metadata is structured data attached to individual items to support search and faceted browsing [Andrews 2022].

In a web site, an information hierarchy is used to structure content into categories and subcategories, as shown in Figure 2.1. In a software application, an information hierarchy might be used to structure menus and sub-menus. Card sorting and tree testing are methods involving representative test users to assist in the design and evaluation of an information hierarchy, respectively.

## 2.1 Card Sorting

*Card sorting* is a design technique used to help construct a usable information hierarchy in a bottom-up approach. Participants are given a set of cards with content topics and are asked to sort them into groups (and possibly subgroups) which make sense to them, providing insights into how they understand and categorize the topics [Sherwin 2018]. When they are satisifed with their groupings, users are asked to label each group, as shown in Figure 2.2. The results over many users are aggregated into a canonical grouping, which essentially corresponds to the bottom level of an information hierarchy. The labels chosen for each group can form the concepts for the next higher level of the hierarchy. This kind of card sorting is sometimes called *open* card sorting.

In contrast, *closed* card sorting is a technique used to validate a predefined grouping. Users sort the concept cards into a given predefined set of categories (groups) and may be asked at the start what they think each category label means. Closed card sorting tests the fit of concepts to an existing group of categories, but does not correspond to how users would naturally navigate an information hierarchy. Tree testing is almost always preferable to closed card sorting to test navigation in an information hierarchy.

A mixture of open and closed sorting, where some categories are predefined, but users are free to change their names or create new categories, is called *hybrid* card sorting. In this thesis, the phrase card sorting, without further qualification, always refers to open card sorting.

In this thesis, the term *card sorting test* (or *card sort*) refers to a single user’s grouping of concepts. The term *card sorting study* is used to refer to a study involving mutliple participants. Typically, three to five participants are enough for the study owner to gain some insight into how users think. However, many more users (50, 100, or more) are required to gain a comprehensive understanding of how to best structure and label an information hierarchy. The commonly available statistics-based analysis tools also require this many participants [Andrews 2022, page 48].

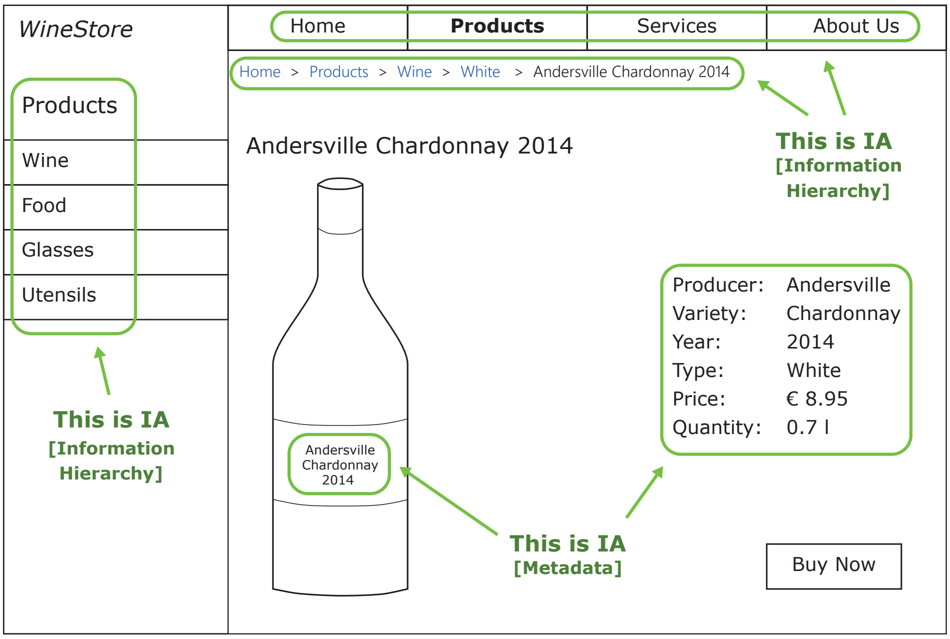


**Figure 2.1:** An example of how IA might be part of a product web site. Information hierarchies are used in navigation. Metadata is used for search and faceted browsing. [Adapted by Keith Andrews from Figure 2-2 of Spencer [2014, page 28], and used with kind permission of Keith Andrews.]

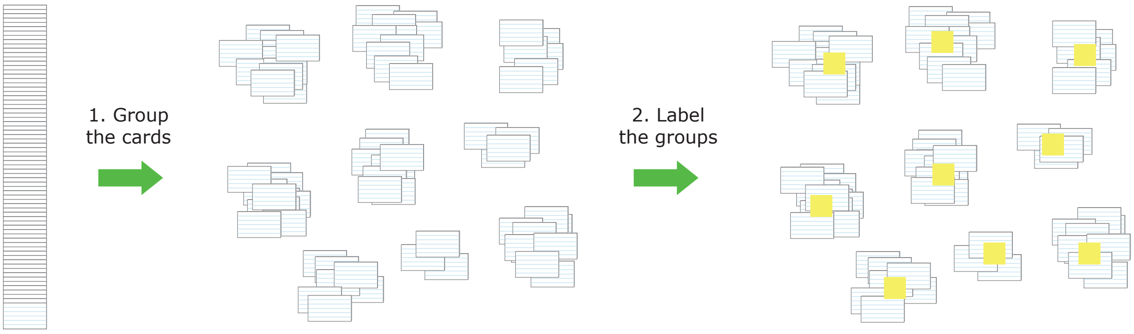


**Figure 2.2:** The process of open card sorting. The user first divides the cards into groups and then labels each group. [Adapted by Keith Andrews from Figure 26 of Schilb [2006, page 40] and used with kind permission of Keith Andrews.]

Card sorting can be done either face-to-face or using an online tool such as Diamond. Online tools have the advantage of scaling easily to dozens or hundreds of participants, but unsupervized card sorts are harder to interpret. Ultimately, card sorting provides an external viewpoint that aids in constructing an information hierarchy likely to benefit the majority of users [Spencer 2009, Chapter 1].

Different users sometimes use different strategies to group the cards. For example, some users group products according to where the might by found in a supermarket, others by what might be needed to make a particular meal. These strategies or organizing principles are called *mindsets* [Andrews 2022, page 48]. Online tools, such as Diamond, usually ask the participant at the end of test (feedback page) to describe the organizing principle (mindset) they used during the test. For meaningful analysis, it is important to first group the individual sorts by mindset, and only aggregate sorts from the same mindset. Typically, the study owner chooses the largest mindset to analyze first.

Next, within a mindset, it is necessary to standardize the names given to particular groups by different users. For example, “Alcohol”, “Alcoholic Beverages”, and “Alcoholic Drinks” might be standardized into “Alcoholic Drinks”.

After that, card sorting applications typically provide numerous tools to help the study owner analyze the results and form aggregate groupings [Spencer 2009, Chapter 10; Albert and Tullis 2022, Section 10.2]. These include:

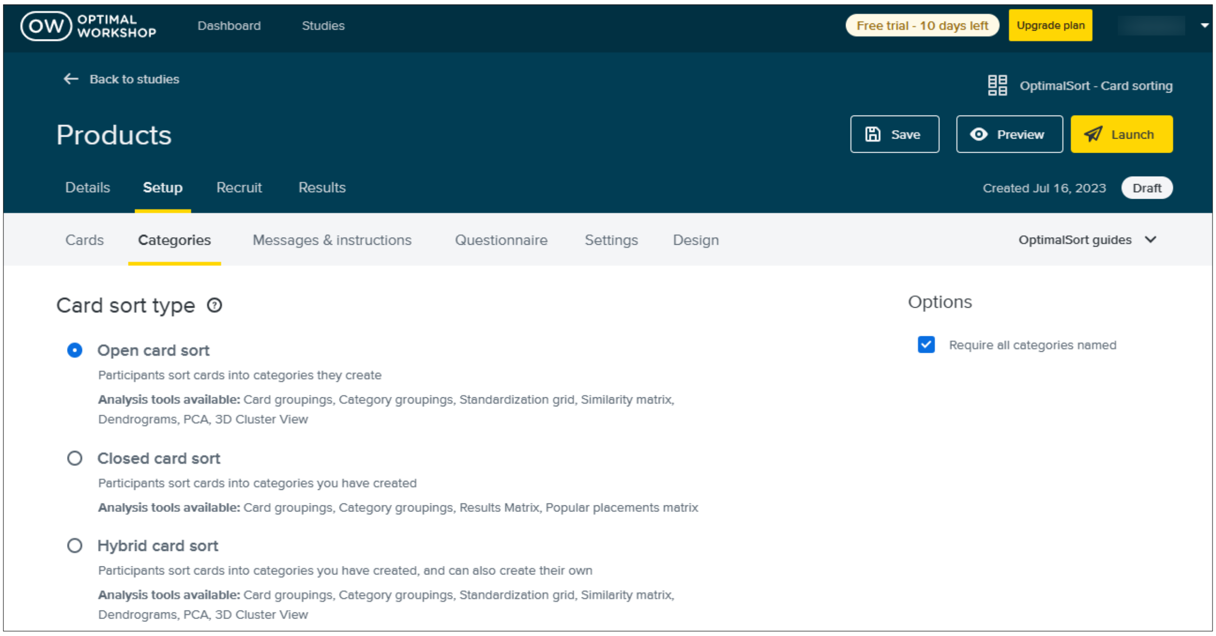
- *Similarity Matrix*: A visual representation of how often each pair of cards was placed together in the same category. This allows the study owner to identify clusters of cards which belong together, regardless of the specific categories they were sorted into [OW 2023b].
- *Dendrogram*: A visual tree-like diagram constructed from the similarity matrix using either bottom-up or top-down hierarchical clustering methods. The branches in the dendrogram indicate cards grouped together by a certain number of respondents. Moving the cut-off line higher up the tree results in fewer groups being formed and vice versa.
- *Similarity Map*: A visual clustering of similar items, based on projection techniques such as Principle Component Analysis (PCA), Multidimensional Scaling (MDS), or t-SNE.

## 2.2 Online Card Sorting Tools

Online card sorting tools are used to conduct card sorting studies remotely. These tools support the creation of digital concept cards representing information items, support the management of participants, and provide a graphical interface for the sorting itself. They collect and analyze data on participants’ sorting choices, provide visual representations, and offer collaboration and reporting features. A selection of online card sorting tools are described below.

### 2.2.1 OptimalSort [2004]

Optimal Workshop provide a suite of online UX tools. OptimalSort is their online card sorting tool [OW 2023a]. It is a paid service, but offers a free plan that comes with some limitations in number of cards and number of participants per study. When creating a card sorting study, the study owner can choose the type of study: open, closed, or hybrid, as shown in Figure 2.3. The study can be configured and concept cards can be imported in bulk or added one by one. After creating the study, the web site provides a link to send to participants to do the sort. OptimalSort supports managing participants, conducting studies, and analyzing the results with a variety of tools. Figure 2.4 shows an example of a user performing an open card sort.



Products

Card sort type

● Open card sort

Participants sort cards into categories they create

**Analysis tools available:** Card groupings, Category groupings, Standardization grid, Similarity matrix, Dendrograms, PCA, 3D Cluster View

○ Closed card sort

Participants sort cards into categories you have created

**Analysis tools available:** Card groupings, Category groupings, Results Matrix, Popular placements matrix

○ Hybrid card sort

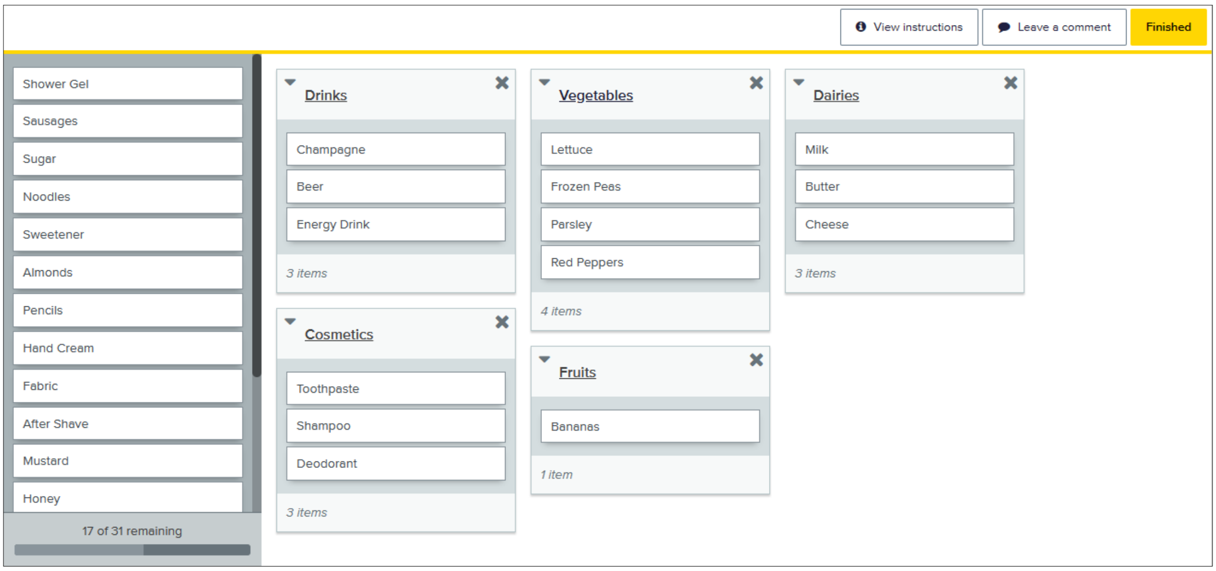
Participants sort cards into categories you have created, and can also create their own

**Analysis tools available:** Card groupings, Category groupings, Standardization grid, Similarity matrix, Dendrograms, PCA, 3D Cluster View

Options

☑ Require all categories named

**Figure 2.3:** OptimalSort: An example of setting up a card sorting study and selecting its type before launching. [Screenshot made by the author of this thesis.]



Shower Gel
Sausages
Sugar
Noodles
Sweetener
Almonds
Pencils
Hand Cream
Fabric
After Shave
Mustard
Honey

17 of 31 remaining

Drinks: Champagne, Beer, Energy Drink — 3 items

Cosmetics: Toothpaste, Shampoo, Deodorant — 3 items

Vegetables: Lettuce, Frozen Peas, Parsley, Red Peppers — 4 items

Fruits: Bananas — 1 item

Dairies: Milk, Butter, Cheese — 3 items

**Figure 2.4:** OptimalSort: A user sorting a set of product cards in an open card sorting test. [Screenshot made by the author of this thesis.]

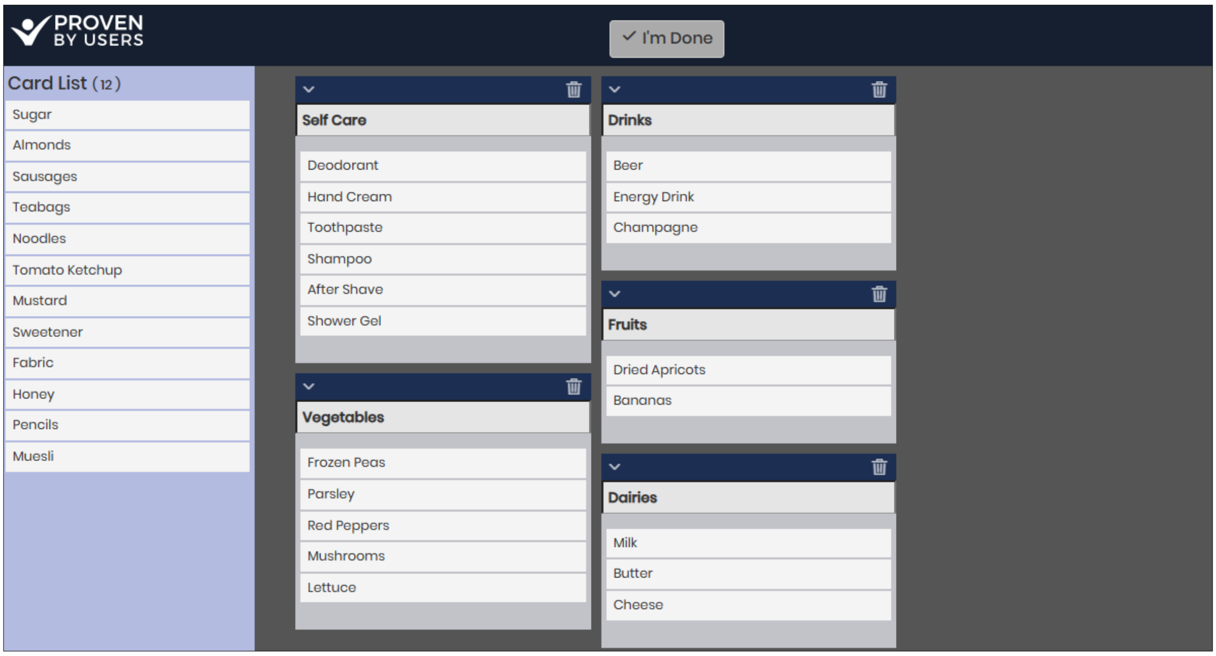


**Figure 2.5:** Proven By Users: A user sorting a set of product cards in an open card sorting test. [Screenshot made by the author of this thesis.]

### 2.2.2 Proven By Users [2017]

Proven By Users provides a suite of online UX tools as a paid service. Its card sorting tool supports all three types of card sorting [PBU 2023]. It has a free version, where the researcher can import unlimited cards, but the number of participants is limited. The study can be configured and concept cards can be imported in bulk or added one by one. After creating the study, the web site provides a link to send to participants to do the sort. Proven By Users supports managing participants, conducting studies, and analyzing the results with a variety of tools. Figure 2.5 shows a user performing an open card sort.

### 2.2.3 Maze [2018]

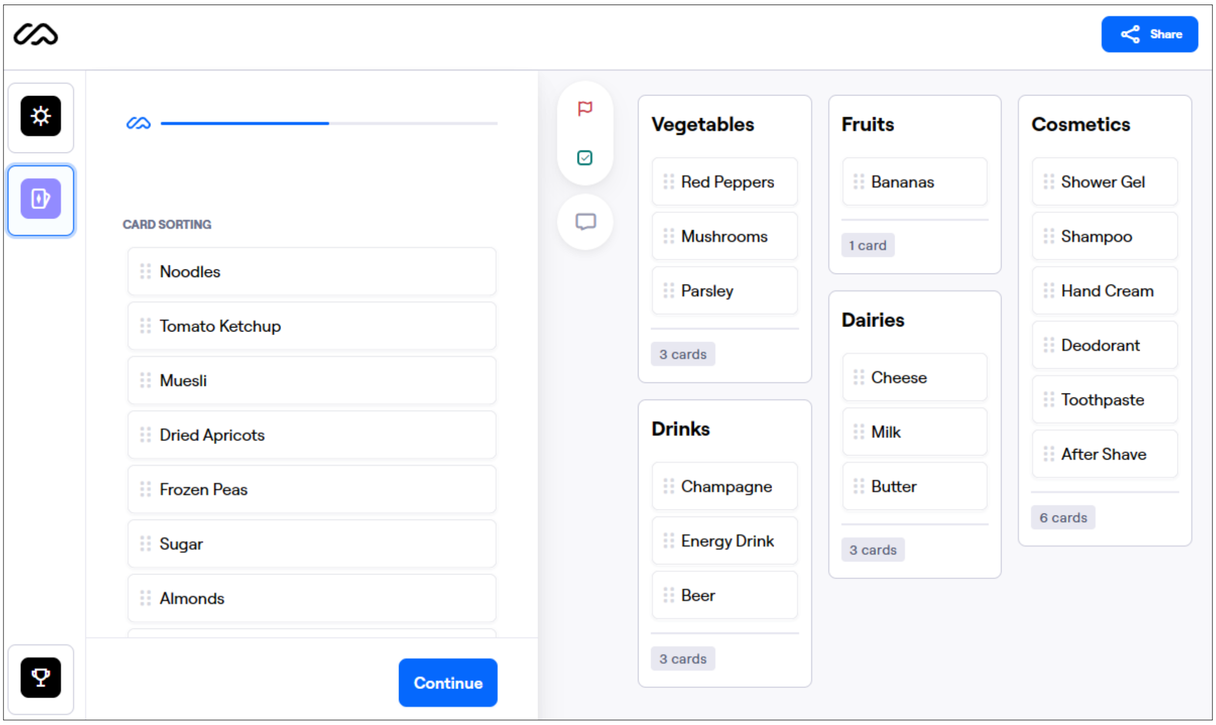
Maze provides a suite of online UX tools as a paid service. Its card sorting tool [Maze 2023] supports both open and closed card sorting. However, the open card sorting tool is only available in a paid plan. The platform provides tools to set up a study, conduct tests, and analyze the resulting data. Figure 2.6 shows a user performing a closed card sort.

### 2.2.4 PlaybookUX [2018]

PlaybookUX provides a suite of online UX tools as a paid service. Its card sorting tool [PBUX 2023a] has a free plan with a limited number of participants. The tool offers all three types of card sorting, but does not have the option to import the cards in bulk. The platform provides tools to set up a study, conduct tests, and analyze the resulting data. Figure 2.7 shows an example of a user performing an open card sorting test.

### 2.2.5 UXtweak [2019]

UXtweak provides a suite of online UX tools as a paid service. Its card sorting tool [UXtweak 2023a] supports all three kinds of card sorting: open, closed, and hybrid. Its free plan offers only one active study at a time with some other limitations on the number of cards and number of participants. The platform provides tools to set up a study, conduct tests, and analyze the resulting data. The platform also



Noodles
Tomato Ketchup
Muesli
Dried Apricots
Frozen Peas
Sugar
Almonds

**Vegetables**
Red Peppers
Mushrooms
Parsley
3 cards

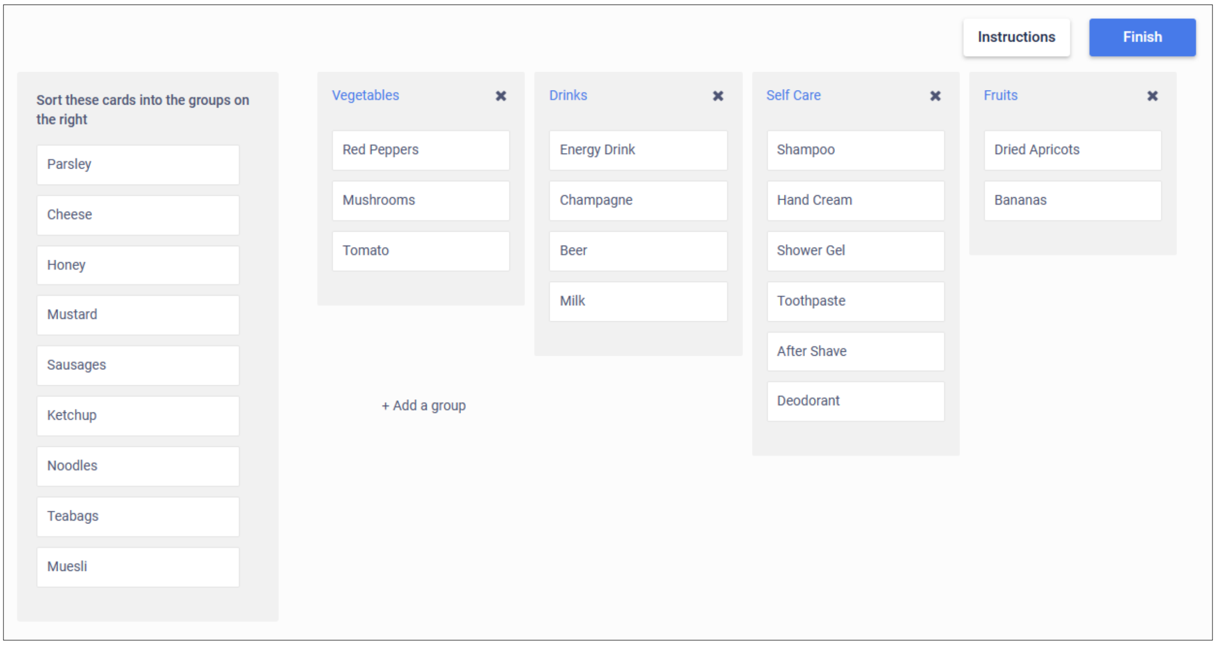
**Drinks**
Champagne
Energy Drink
Beer
3 cards

**Fruits**
Bananas
1 card

**Dairies**
Cheese
Milk
Butter
3 cards

**Cosmetics**
Shower Gel
Shampoo
Hand Cream
Deodorant
Toothpaste
After Shave
6 cards

**Figure 2.6:** Maze: A user sorting a set of product cards into predefined categories in a closed card sort. [Screenshot made by the author of this thesis.]



Sort these cards into the groups on the right
Parsley
Cheese
Honey
Mustard
Sausages
Ketchup
Noodles
Teabags
Muesli

Vegetables ✖
Red Peppers
Mushrooms
Tomato

Drinks ✖
Energy Drink
Champagne
Beer
Milk

Self Care ✖
Shampoo
Hand Cream
Shower Gel
Toothpaste
After Shave
Deodorant

Fruits ✖
Dried Apricots
Bananas

**Figure 2.7:** PlaybookUX: A user sorting a set of product cards in an open card sorting test. [Screenshot made by the author of this thesis.]

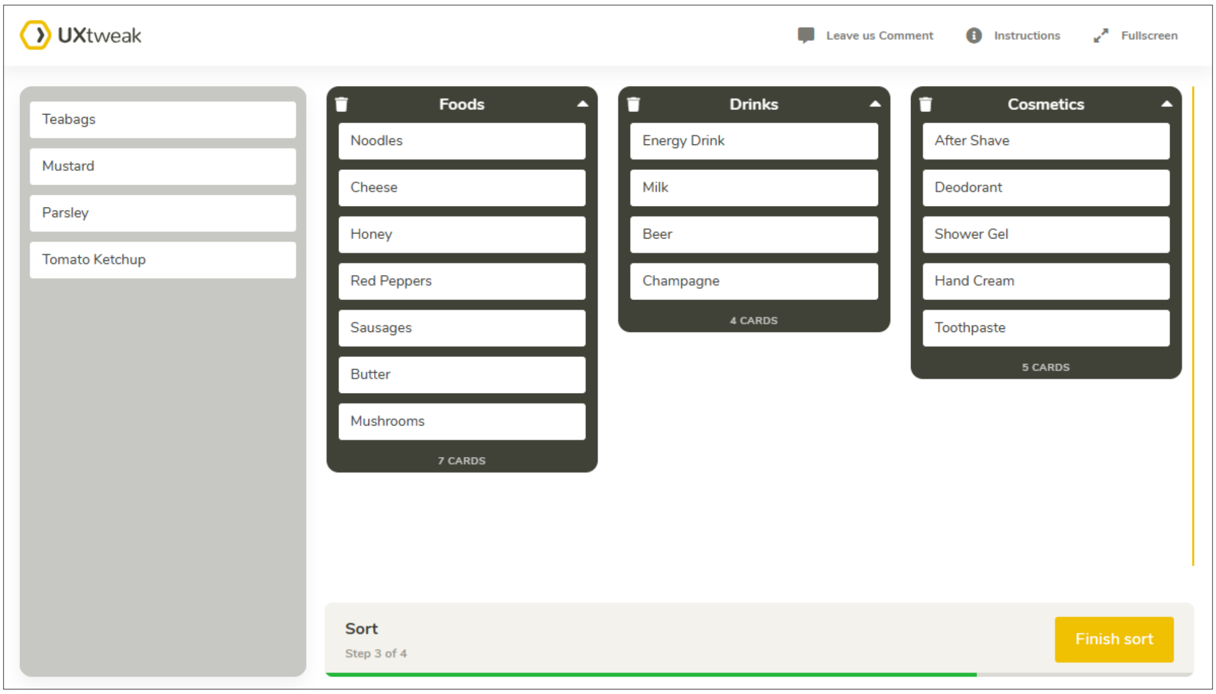
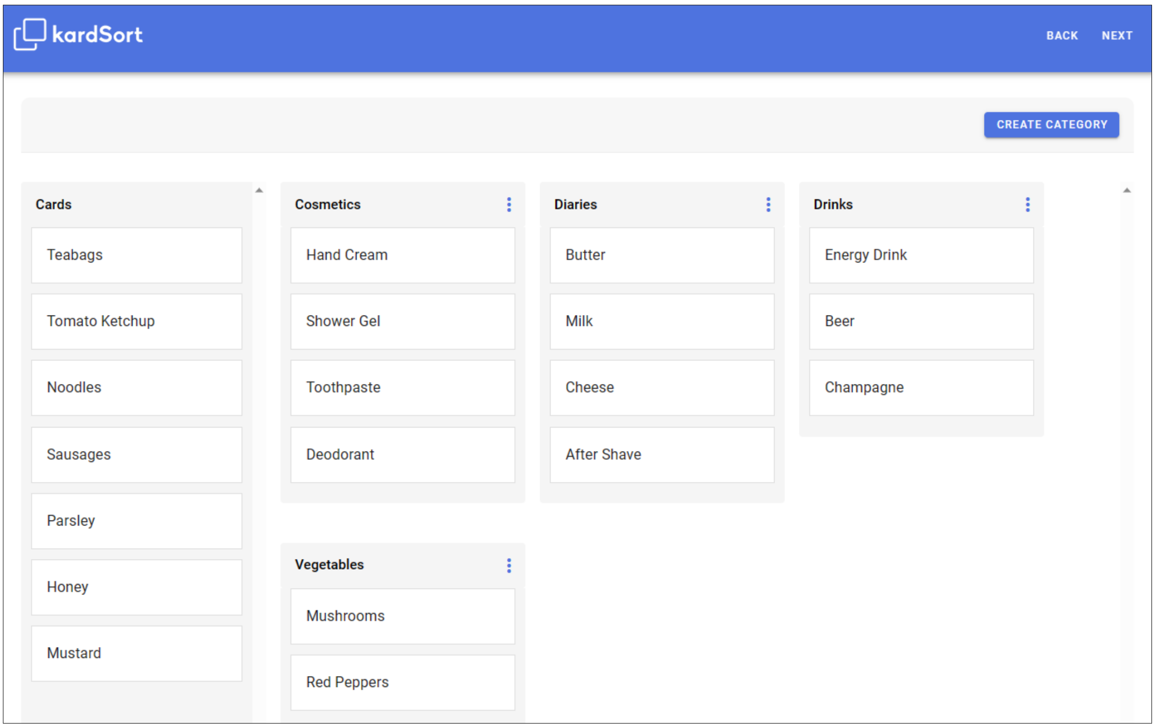


**Figure 2.8:** UXtweak: A user sorting a set of product cards in an open card sort. [Screenshot made by the author of this thesis.]

offers to recruit participants on behalf of the study owner. Figure 2.8 shows an example of test taken by a participant using the card sorting tool by UXtweak.

### 2.2.6 KardSort [2020]

KardSort is an online free-to-use tool to perform card sorting tests [Balachandran 2023]. It has a limitation of 20 cards, which can be removed for one year by making a donation to help cover some of the server costs. KardSort supports all three types of card sorting. KardSort does not provide any analysis tools itself: the results need to be exported for analysis with an external tool. In addition to CSV format, the platform supports export to SynCaps or Casolysis format too. Figure 2.9 shows a user performing an open card sorting test.



Cards
Teabags
Tomato Ketchup
Noodles
Sausages
Parsley
Honey
Mustard

Cosmetics
Hand Cream
Shower Gel
Toothpaste
Deodorant

Vegetables
Mushrooms
Red Peppers

Diaries
Butter
Milk
Cheese
After Shave

Drinks
Energy Drink
Beer
Champagne

**Figure 2.9:** KardSort: A user sorting a set of product cards in an open card sorting test. [Screenshot made by the author of this thesis.]

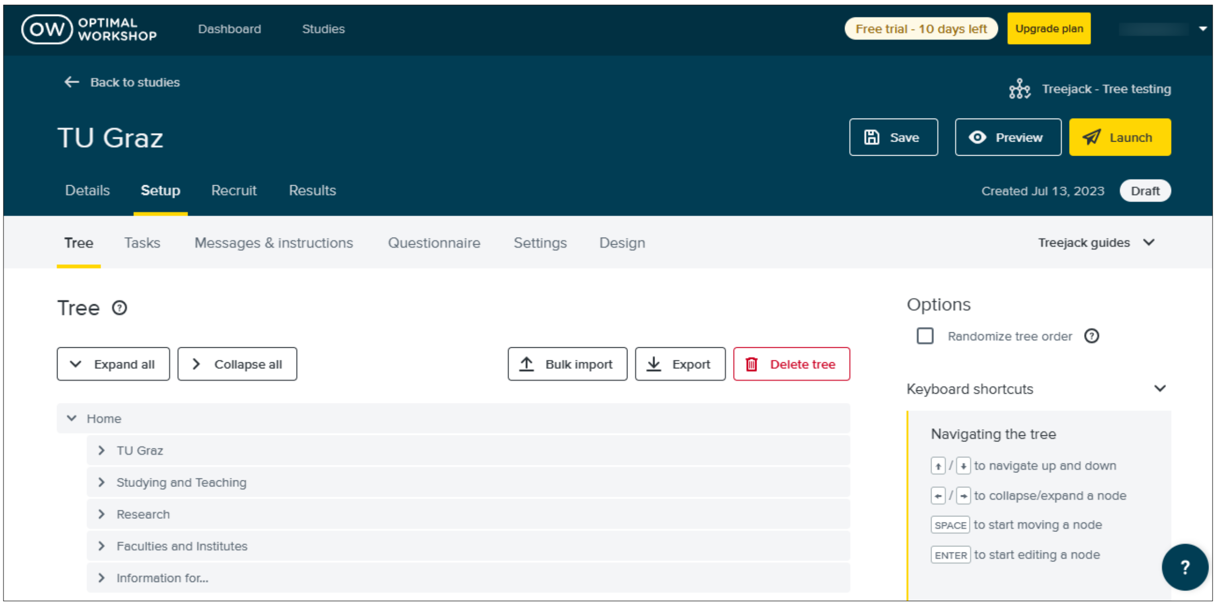
## 2.3 Tree Testing

*Tree testing*, also known as hierarchy validation or reverse card sorting, is a method used to validate the navigation hierarchy of a web site or application. The process involves presenting a plain, bare-bones version of the existing or proposed navigation hierarchy to users and asking them where they would expect to find certain items (known-item search). This tests the basic hierarchy, without the influence of visual or navigation design. The benefits of tree testing include the ability to determine if the labels make sense to users, if the content is grouped logically, and if users look where expected. Tree testing can be done using index cards (paper-based), a computer-based tree testing tool (locally installed or online), or through either face-to-face (supervised) or unsupervised methods [Andrews 2022].

In this thesis, the term *tree test* refers to a single user's participation in a series of tree testing tasks. The term *tree testing study* is used to refer to a study involving mutliple participants. Typically, around 20 to 30 participants are required to provide good results. Online tree testing tools make it easy to conduct studies with large numbers of participants.

The various tree testimg tools provide a number of metrics and techniques to help with the analysis of tree testing results. These include:

- *Success Rate*: The success rate for a task is the average percentage of participants who correctly answered that task [Whitenton 2017].
- *Directness Rate*: The directness rate for a task is the average percentage of participants who correctly answered a task without backtracking [Whitenton 2017].
- *Task Completion Time*: The task completion time represents how long it took the participant to complete a task.
- *Path Tree*: A path tree is a tree visualization, which shows an aggregation of all the paths taken by participants for a particular task. Thicker branches indicate more users took this path.



↑ / ↓ to navigate up and down

← / → to collapse/expand a node

SPACE to start moving a node

ENTER to start editing a node

**Figure 2.10:** Treejack: An example setting up a tree testing study. [Screenshot made by the author of this thesis.]

- *Destinations Table*: The destinations table shows, for each task, with all the answers, how many users selected which node in the tree as their answer.

## 2.4 Online Tree Testing Tools

Online tree testing tools are web-based platforms specifically designed to facilitate the process of conducting tree testing to evaluate an information hierarchy. They provide a digital environment to create, administer, and analyze tree testing studies.

### 2.4.1 Treejack [2008]

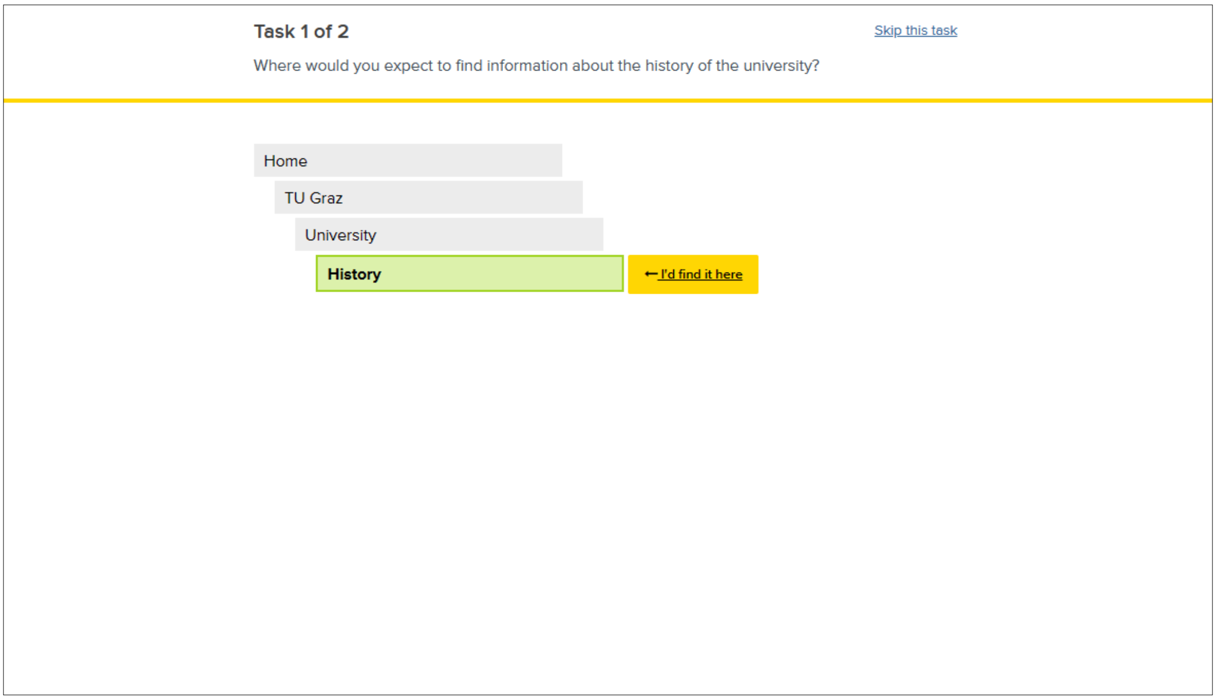
Optimal Workshop provide a suite of online UX tools. Treejack is their tree testing tool [OW 2023c]. It supports the creation and running of online tree testing studies, and provides tools to help analyze the results. Figure 2.10 shows a study owner setting up a tree testing study. Figure 2.11 shows a user navigating through an information hierarchy in the process of answering one of the tasks. Treejack records all the answers from participants and provides analysis of the results for the study owner.

### 2.4.2 PlaybookUX [2018]

PlaybookUX provides a suite of online UX tools as a paid service. Its tree testing tool [PBUX 2023b] has a free plan with a limited number of tests. The platform provides tools to set up the study, create the tree and the tasks, and conduct tests. Figure 2.12 shows the tree and the participant is answering a task. The platform also offers the study owner different methods to analyze the results of the tree tests, as shown in Figure 2.13.

### 2.4.3 UXtweak [2019]

UXtweak provides a suite of online UX tools as a paid service. Its tree testing tool [UXtweak 2023b] has a free plan which offers only one active study at a time with some other limitations on the number of tasks per study and the number of participants. The platform provides tools to set up a study, conduct



Task 1 of 2

Skip this task

Where would you expect to find information about the history of the university?

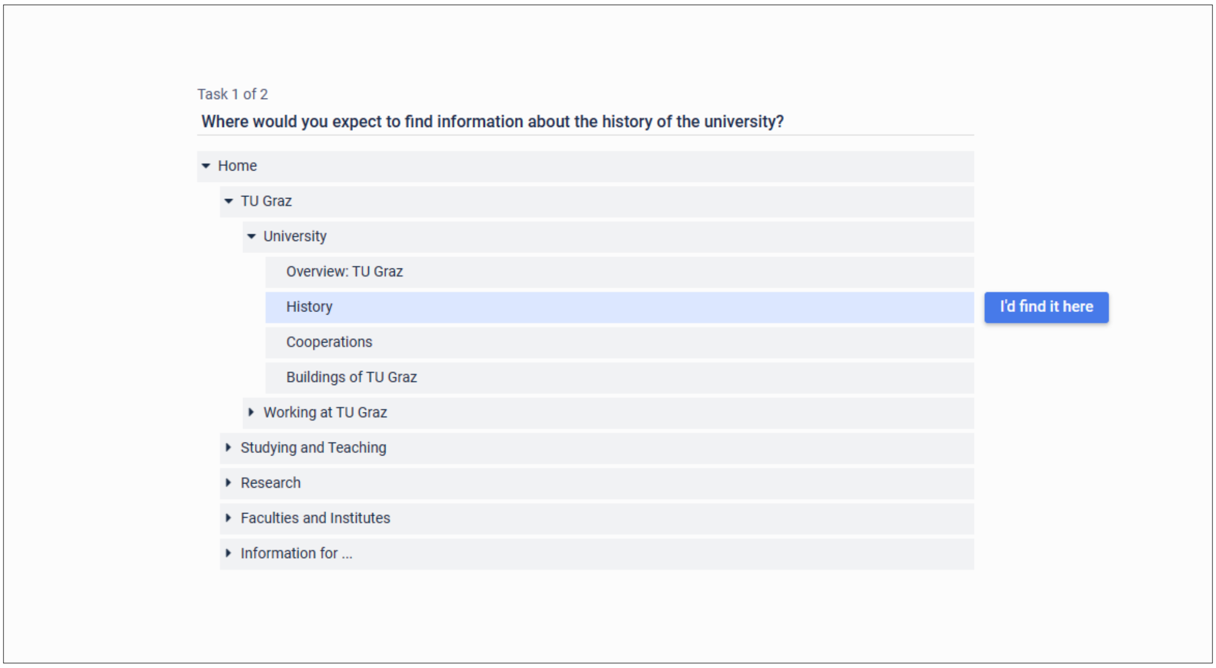
Home

TU Graz

University

History

**Figure 2.11:** Treejack: A user navigating a tree in a tree test study. [Screenshot made by the author of this thesis.]

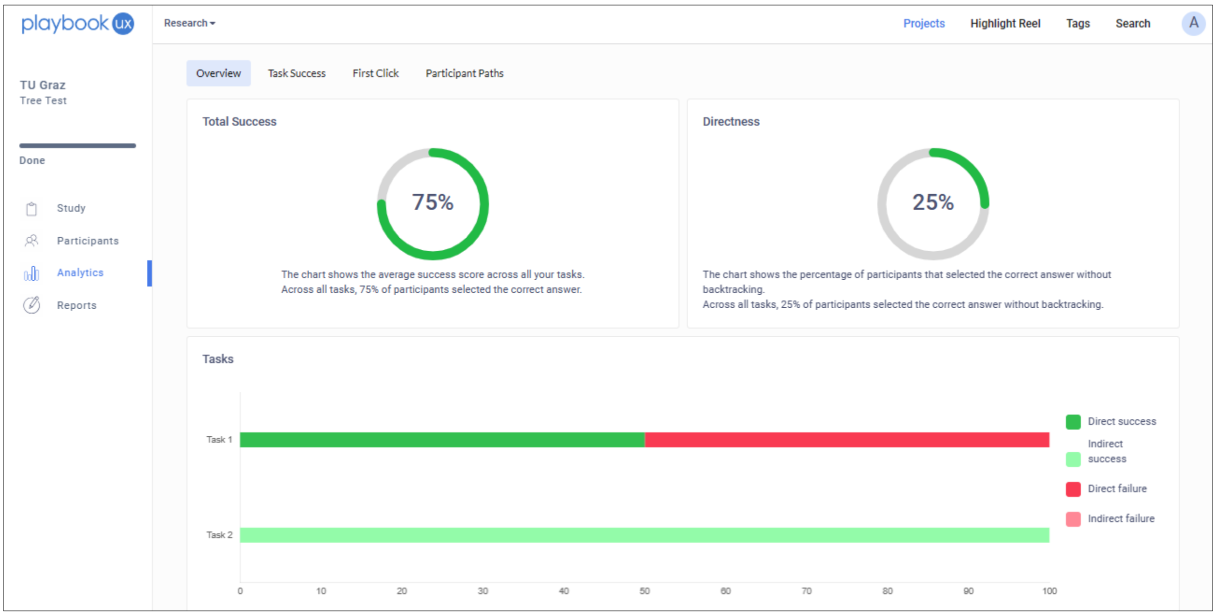


Task 1 of 2

Where would you expect to find information about the history of the university?

- Home
  - TU Graz
    - University
      - Overview: TU Graz
      - History
      - Cooperations
      - Buildings of TU Graz
    - Working at TU Graz
  - Studying and Teaching
  - Research
  - Faculties and Institutes
  - Information for ...

**Figure 2.12:** PlaybookUX: A user navigating a tree in a tree test study. [Screenshot made by the author of this thesis.]

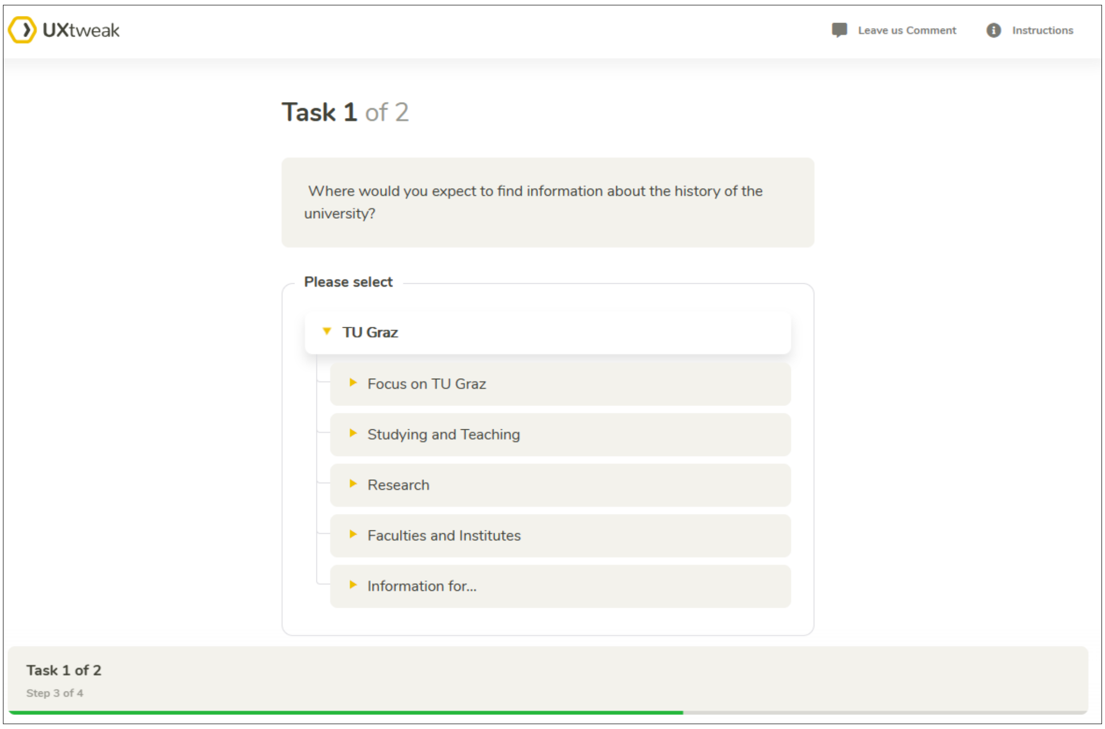


The chart shows the average success score across all your tasks.
Across all tasks, 75% of participants selected the correct answer.

The chart shows the percentage of participants that selected the correct answer without backtracking.
Across all tasks, 25% of participants selected the correct answer without backtracking.

**Figure 2.13:** PlaybookUX: The analytics provided to a tree testing study owner. [Screenshot made by the author of this thesis.]

tests, and analyze the results. The platform also offers the option to recruit participants on behalf of the study owner. Figure 2.14 shows an example a user navigating a tree as part of a tree testing study.



Task 1 of 2

Where would you expect to find information about the history of the university?

**Figure 2.14:** UXtweak: A user navigating a tree in a tree testing study. [Screenshot made by the author of this thesis.]

## Chapter 3

# Modern Web Technology

Modern web technology has come a long way since the early days of the internet, and today's web landscape is more diverse and dynamic than ever. With advances in HTML, CSS, and JavaScript, as well as the introduction of new technologies, web developers now have a vast array of tools and frameworks at their disposal to create engaging, interactive, and responsive web applications. This chapter covers some of the technologies which are used in the implementation of Diamond.

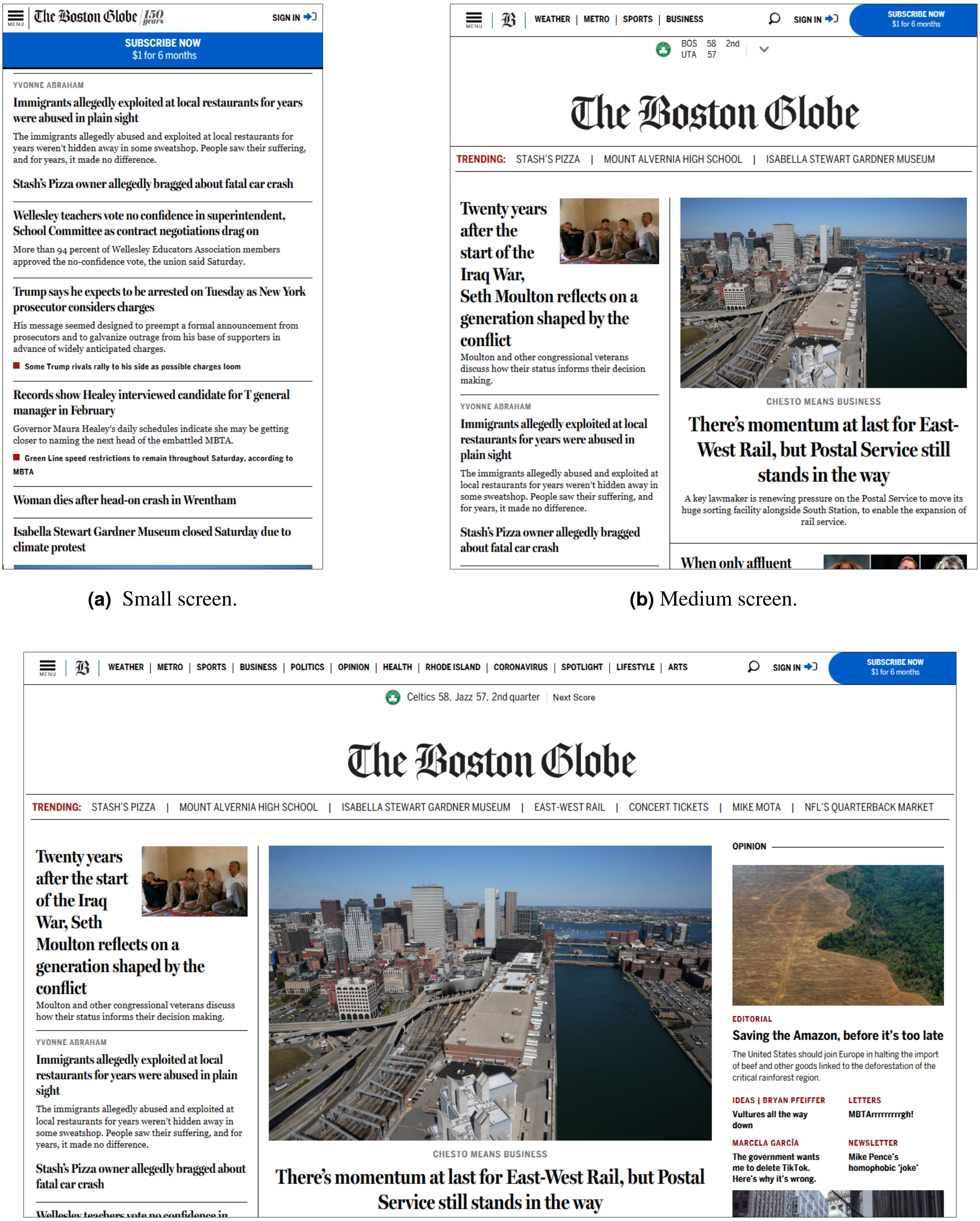
## 3.1 Responsive Web Design

Responsive web design (RWD) was first defined in 2010 by Ethan Marcotte [Marcotte 2010]. It is a web design approach which allows web pages to be displayed appropriately on all kinds of end device while ensuring good usability. An example is shown in Figure 3.1. The Boston Globe was the first ever responsive web site. It adapts itself to best utilize the available display space.

Responsive web designs also respond to different kinds of interactivity. Mouse, keyboard, touch, and any other input mechanisms available on the end device should be supported. This means that web sites can be accessed easily and efficiently on a range of devices, from desktop computers to smartphones and tablets. The approach uses a number of CSS and HTML features and techniques to create a design that adapts to the environment in which it is viewed. Since it allows web designers to provide a seamless and consistent user experience across devices, responsive web design has become the default way of building web sites [MDN 2023a].

## 3.2 Single Page Applications

A Single-Page Application (SPA) is a type of web application that loads only a single web document and dynamically updates the content using JavaScript APIs such as XMLHttpRequest and Fetch to retrieve the content in a JSON [Ecma 2017; JSON 2023] format. This approach allows users to navigate the web site without having to load new pages from the server. An illustration of how an SPA is rendered can be seen in Figure 3.2. Note that JavaScript is required for an SPA to work. The result is a more dynamic and seamless user experience, with potentially significant performance gains. However, there are some tradeoffs to consider, including the need to put in extra effort to maintain state and implement navigation, as well as potential challenges with search engine optimization (SEO) and performance monitoring. Despite these challenges, SPAs continue to be a popular approach for web developers looking to create fast, responsive, and interactive web applications [MDN 2023b].



MENU The Boston Globe 150 years SIGN IN

SUBSCRIBE NOW
$1 for 6 months

YVONNE ABRAHAM

**Immigrants allegedly exploited at local restaurants for years were abused in plain sight**

The immigrants allegedly abused and exploited at local restaurants for years weren't hidden away in some sweatshop. People saw their suffering, and for years, it made no difference.

**Stash's Pizza owner allegedly bragged about fatal car crash**

**Wellesley teachers vote no confidence in superintendent, School Committee as contract negotiations drag on**

More than 94 percent of Wellesley Educators Association members approved the no-confidence vote, the union said Saturday.

**Trump says he expects to be arrested on Tuesday as New York prosecutor considers charges**

His message seemed designed to preempt a formal announcement from prosecutors and to galvanize outrage from his base of supporters in advance of widely anticipated charges.

■ Some Trump rivals rally to his side as possible charges loom

**Records show Healey interviewed candidate for T general manager in February**

Governor Maura Healey's daily schedules indicate she may be getting closer to naming the next head of the embattled MBTA.

■ Green Line speed restrictions to remain throughout Saturday, according to MBTA

**Woman dies after head-on crash in Wrentham**

**Isabella Stewart Gardner Museum closed Saturday due to climate protest**

**(a)** Small screen.

MENU | WEATHER | METRO | SPORTS | BUSINESS SIGN IN SUBSCRIBE NOW $1 for 6 months

BOS 58 2nd
UTA 57

The Boston Globe

TRENDING: STASH'S PIZZA | MOUNT ALVERNIA HIGH SCHOOL | ISABELLA STEWART GARDNER MUSEUM

**Twenty years after the start of the Iraq War, Seth Moulton reflects on a generation shaped by the conflict**

Moulton and other congressional veterans discuss how their status informs their decision making.

YVONNE ABRAHAM

**Immigrants allegedly exploited at local restaurants for years were abused in plain sight**

The immigrants allegedly abused and exploited at local restaurants for years weren't hidden away in some sweatshop. People saw their suffering, and for years, it made no difference.

**Stash's Pizza owner allegedly bragged about fatal car crash**

CHESTO MEANS BUSINESS

**There's momentum at last for East-West Rail, but Postal Service still stands in the way**

A key lawmaker is renewing pressure on the Postal Service to move its huge sorting facility alongside South Station, to enable the expansion of rail service.

**When only affluent**

**(b)** Medium screen.

MENU | WEATHER | METRO | SPORTS | BUSINESS | POLITICS | OPINION | HEALTH | RHODE ISLAND | CORONAVIRUS | SPOTLIGHT | LIFESTYLE | ARTS SIGN IN SUBSCRIBE NOW $1 for 6 months

Celtics 58, Jazz 57, 2nd quarter | Next Score

The Boston Globe

TRENDING: STASH'S PIZZA | MOUNT ALVERNIA HIGH SCHOOL | ISABELLA STEWART GARDNER MUSEUM | EAST-WEST RAIL | CONCERT TICKETS | MIKE MOTA | NFL'S QUARTERBACK MARKET

**Twenty years after the start of the Iraq War, Seth Moulton reflects on a generation shaped by the conflict**

Moulton and other congressional veterans discuss how their status informs their decision making.

YVONNE ABRAHAM

**Immigrants allegedly exploited at local restaurants for years were abused in plain sight**

The immigrants allegedly abused and exploited at local restaurants for years weren't hidden away in some sweatshop. People saw their suffering, and for years, it made no difference.

**Stash's Pizza owner allegedly bragged about fatal car crash**

**Wellesley teachers vote no confidence in**

CHESTO MEANS BUSINESS

**There's momentum at last for East-West Rail, but Postal Service still stands in the way**

OPINION

EDITORIAL

**Saving the Amazon, before it's too late**

The United States should join Europe in halting the import of beef and other goods linked to the deforestation of the critical rainforest region.

IDEAS | BRYAN PFEIFFER

**Vultures all the way down**

LETTERS

**MBTArrrrrrrrrgh!**

MARCELA GARCÍA

**The government wants me to delete TikTok. Here's why it's wrong.**

NEWSLETTER

**Mike Pence's homophobic 'joke'**

**(c)** Large screen.

**Figure 3.1:** A responsive web site adapts itself to the available display space. The Boston Globe was the first web site to implement a responsive design. [Screenshots created by the author of this thesis.]

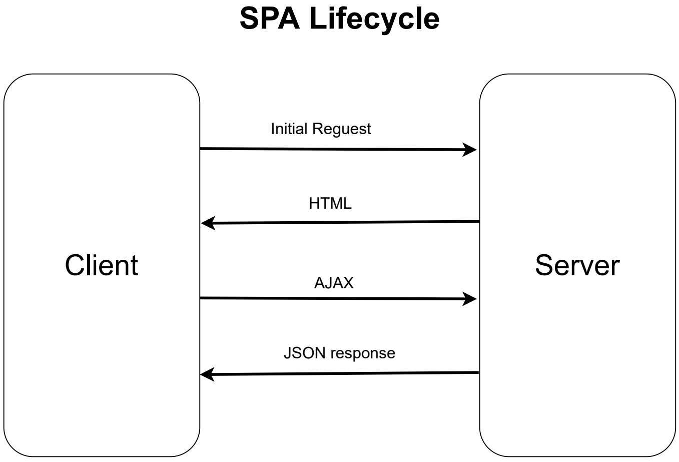


**Figure 3.2:** The lifecycle of a single-page application (SPA). The server sends the initial page as HTML to the client. Further data is sent in JSON format based on AJAX requests sent by the client. JavaScript is required for an SPA to work. [Diagram made by the author of this thesis.]

## 3.3 MEAN Stack

Diamond is a responsive, single-page web application built with the MEAN (MongoDB, Express, Angular, Node) stack:

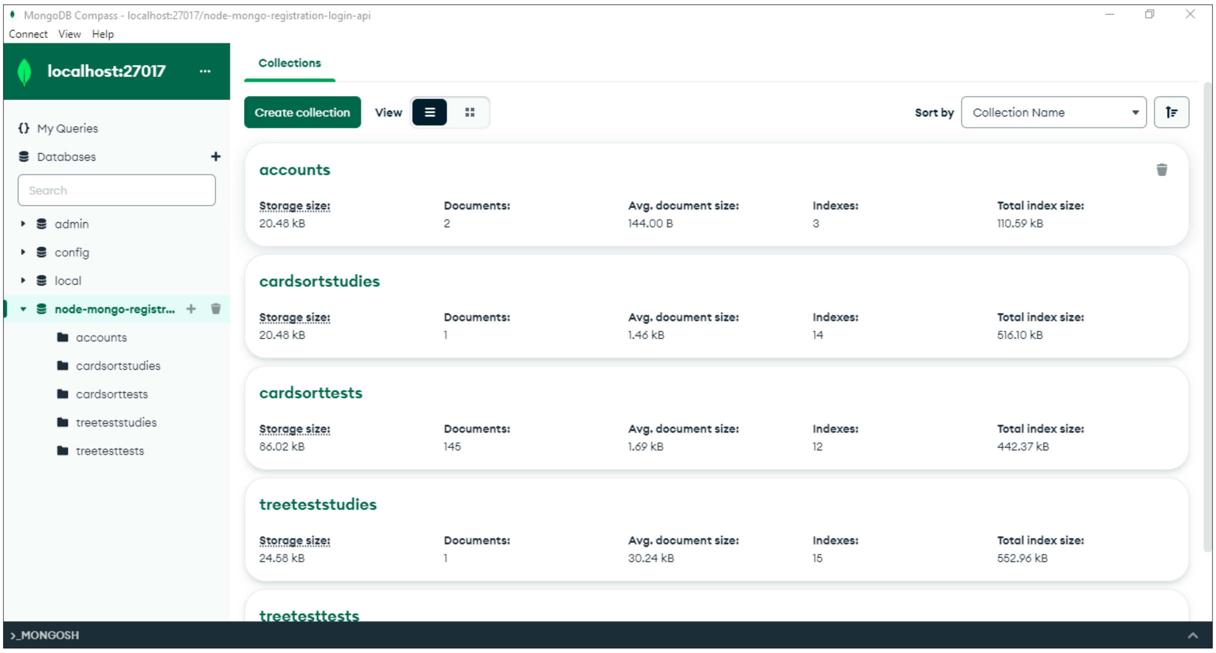
- *Node*: Node.js is an open-source, cross-platform JavaScript runtime environment [Node 2023]. It is built on top of the Google Chrome V8 JavaScript engine, and is mainly used to create web servers. It is used to run the web server of Diamond, which is built with Express.

  npm is the Node Package Manager and is a library and registry for JavaScript software packages. It is used to install different JavaScript packages and manage their dependencies.

- *Express*: Express is a minimal and flexible Node.js web application framework, which provides HTTP utility methods and middleware for building web and mobile applications [Express 2023]. Express is used to develop the API endpoints which Diamond uses to communicate with the MongoDB database.

- *Angular*: Angular is one of the more popular SPA frameworks [MDN 2023b]. Developing web apps in Angular consists of creating components [Angular 2023]. A component is simply a folder containing one HTML, one CSS, and one TypeScript file. At the moment writing, Diamond is being developed with Angular 15.

- *MongoDB*: MongoDB is a document database used to build scalable applications [Mongo 2023]. Diamond uses MongoDB to store all of its data. MongoDB Compass is a utility which can be used to view, analyze, and query the data in a MongoDB database and is very helpful for debugging. Figure 3.3 shows an overview of the database collections which hold data for Diamond.



accounts

| Storage size: | Documents: | Avg. document size: | Indexes: | Total index size: |
|---|---|---|---|---|
| 20.48 kB | 2 | 144.00 B | 3 | 110.59 kB |

cardsortstudies

| Storage size: | Documents: | Avg. document size: | Indexes: | Total index size: |
|---|---|---|---|---|
| 20.48 kB | 1 | 1.46 kB | 14 | 516.10 kB |

cardsorttests

| Storage size: | Documents: | Avg. document size: | Indexes: | Total index size: |
|---|---|---|---|---|
| 86.02 kB | 145 | 1.69 kB | 12 | 442.37 kB |

treeteststudies

| Storage size: | Documents: | Avg. document size: | Indexes: | Total index size: |
|---|---|---|---|---|
| 24.58 kB | 1 | 30.24 kB | 15 | 552.96 kB |

treetesttests

**Figure 3.3:** MongoDB Compass showing some of the collections used by Diamond to hold data. [Screenshot made by the author of this thesis.]

# Chapter 4

# Diamond Version 1.0

The original version of Diamond only supported tree testing and was called TreeTest. It was built by Ajdin Mehic for his Master's thesis [Mehic 2019] using the MEAN stack and Angular 7.

Later, in 2020, a group of students in the course Information Architecture and Web Usability added support for card sorting and the project was renamed to Diamond [Oser et al. 2021]. In 2021 [Brandl et al. 2022] and 2022 [Egger et al. 2023], the project was further extended and improved. This status of the project will be called Diamond Version 1.0.

Back then, Diamond was hosted on using Heroku's free service. Study owners were able to create and run tree testing studies and analyze the results within Diamond. For card sorting, it was only possible to create and run card sorting studies. To analyze the results, it was necessary to download them and use another tool, such as CSA [Kargl et al. 2018].

## 4.1 Limitations

Sometimes, a study owner might want to archive a study, or to move a study and its data to another instance of Diamond. Unfortunately, Diamond had no possibility to export or import studies. The only possible workaround was to export and reimport (parts of) the database using MongoDB tools, which was both tricky and time-consuming.

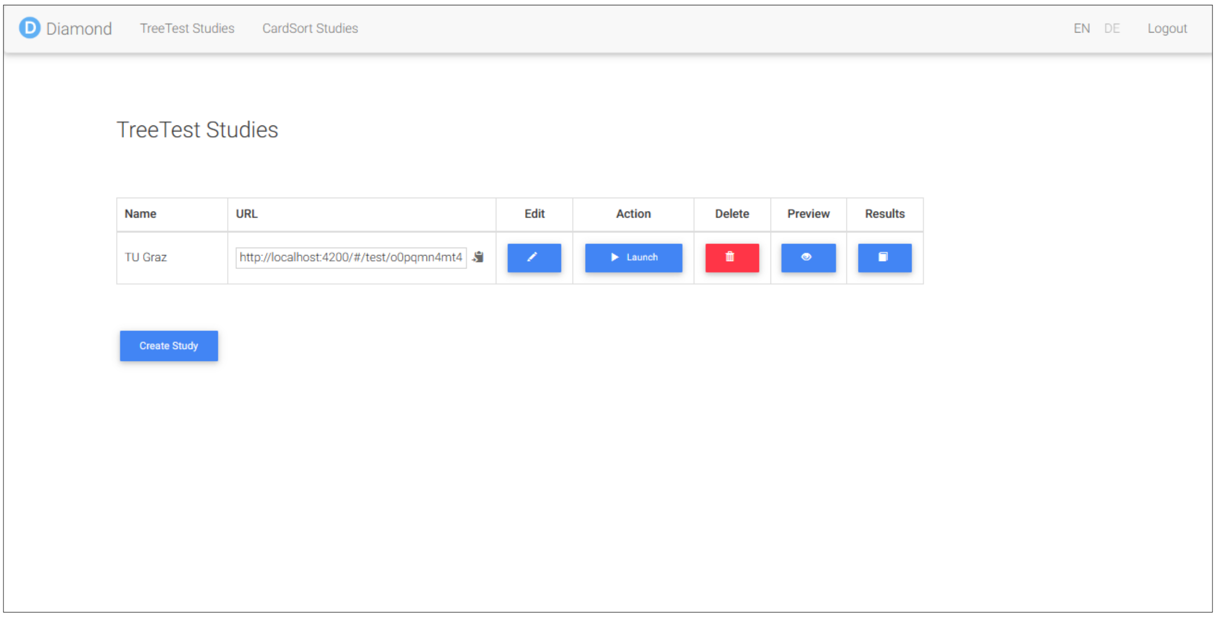
Additionally, study owners desired more meta-information to better distinguish their studies, for example, the last time the study was launched and ended.

Since Diamond was initially built for tree testing and its card sorting functionality was added later, the source code of the app was poorly organized, with inconsistent naming of files, components, and database fields. It needed refactoring to make the whole developer experience easier when adding or maintaining features.

## 4.2 Capabilities

Diamond has three types of users: admin, study owner, and participant. An admin user manages all user accounts. A study owner creates and manages studies. Participants take part in studies. Admin and study owners need an account, but participants do not. A default admin account is created during the initial setup of Diamond.

After logging in to Diamond, the study owner's landing page displays all the studies created by that study owner, as shown in Figure 4.1. Studies are divided into two types: CardSort Studies and TreeTest Studies, reflecting the two major components of Diamond. The study owner can see and access their studies and perform actions such as Preview, Delete, Edit, Launch/End the study, or see its Results.



TreeTest Studies

| Name | URL | Edit | Action | Delete | Preview | Results |
|---|---|---|---|---|---|---|
| TU Graz | http://localhost:4200/#/test/o0pqmn4mt4 |  | ▶ Launch |  |  |  |

**Figure 4.1:** Diamond v1.0: The landing page of a study owner. [Screenshot made by the author of this thesis.]

Creating a tree testing study on Diamond is a straightforward process. The study owner first needs to set a title and an optional password, as shown in Figure 4.2. Secondly, the tree to be tested is created or imported from a tab-separated value (TSV) file, as shown in Figure 4.3. Thirdly, tasks are configured by specifying questions and their correct location in the tree, as shown in Figure 4.4. Finally, it is possible to configure various messages, such as the welcome message, instructions, and thank you message shown to participants.

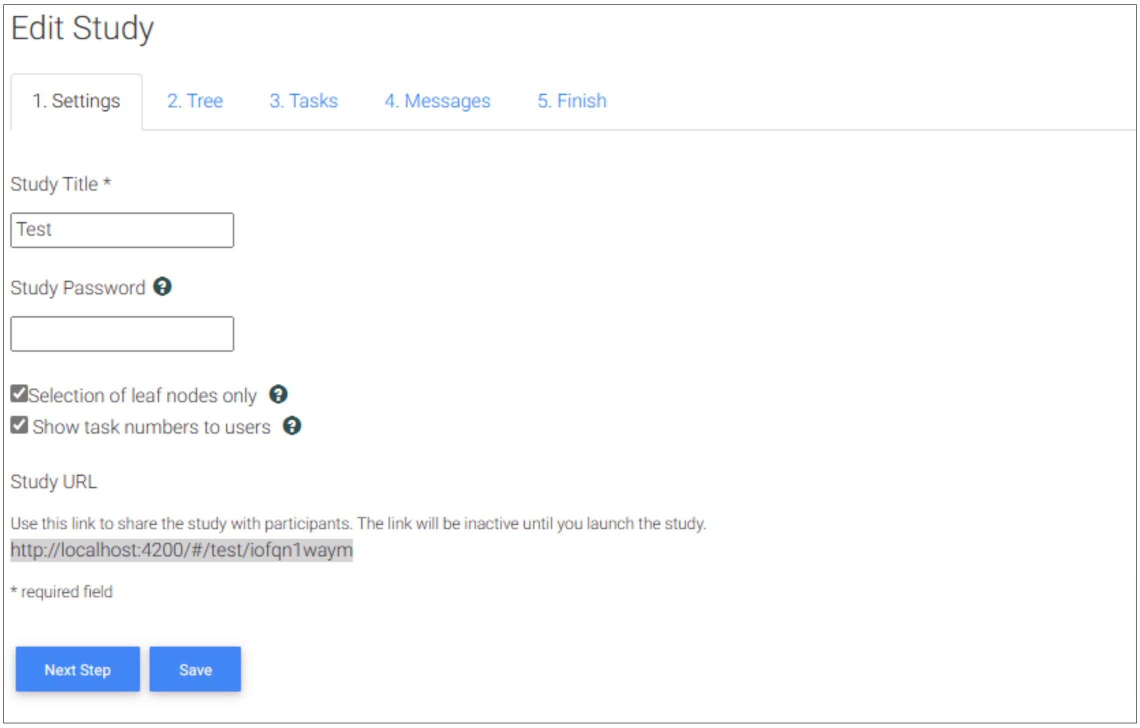
The study creation process is analogous for a card sorting study. Cards are created or imported, and various messages can be configured.

When clicking on the Results of a study in the landing page, the study owner sees the results page, as shown in Figure 4.5. Initially, an overview of various metrics like completion and success rates is shown. Further tabs provide access to the participants, task analysis, and destinations table.

In the Participants tab of the results page of a tree test study, the study owner can see a tabular view of all the participants in the study together with various metadata, such as the date and time of the test, how many tasks were completed, and how many tasks were correctly completed. This is shown in Figure 4.6. The study owner can delete or exclude a participant from consideration.

For card sorting studies, the study owner also sees a tabular overview of participants, as shown in Figure 4.7. Each row shows one participant, the date and time of their sort, the explanation of their sorting strategy, and other feedback. Clicking the View Results button opens the result page for that participant, showing the categories and their cards. This is shown in Figure 4.8. The participants and their sorts can each be exported as a CSV file.

For a card sorting study, no tools for further analysis are provided in Diamond. The results must be exported and anlyzed externally. However, a number of analyis tools are provided for tree testing studies. The study owner can see the success and directness rate for each task, as shown in Figure 4.9. The study owner can also see the aggregated path tree of paths taken by participants while doing the task, shown in Figure 4.10. The green path shows the correct answer, thicker branches indicate more participants took that path. The destinations table, shown in Figure 4.11, indicates how many times a particular node in the tree was chosen as the answer for each task.



Edit Study

Study Title *

Test

Study Password

☑ Selection of leaf nodes only

☑ Show task numbers to users

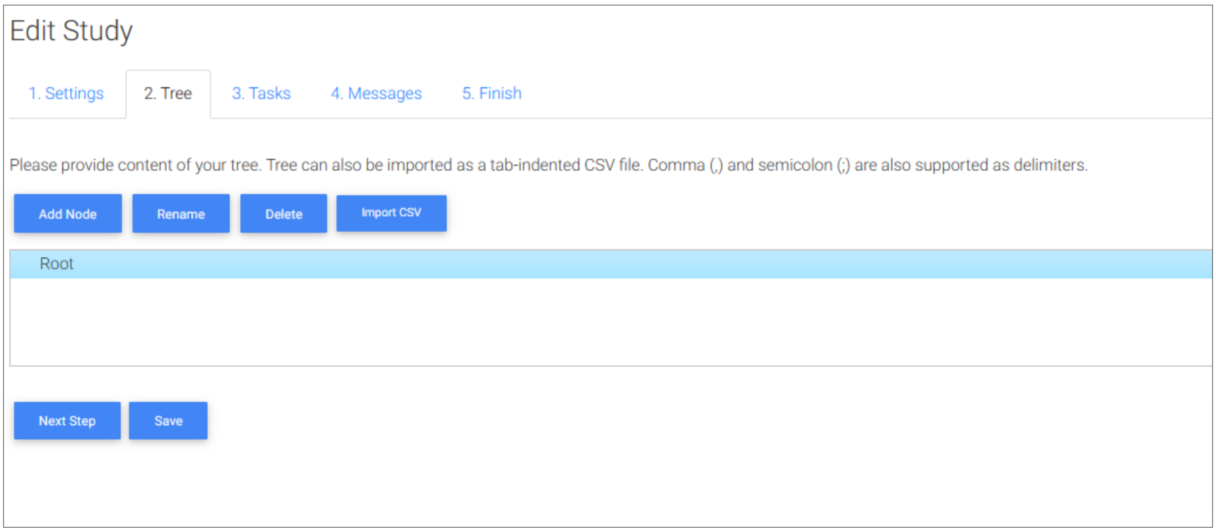
Study URL

Use this link to share the study with participants. The link will be inactive until you launch the study.

http://localhost:4200/#/test/iofqn1waym

* required field

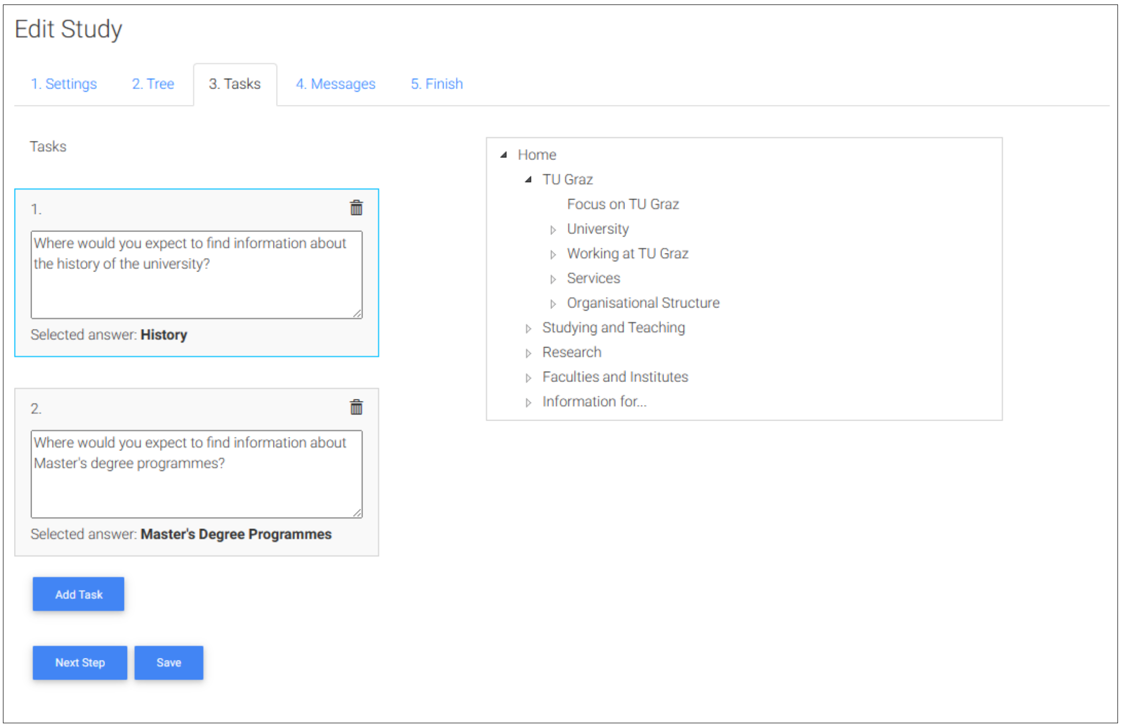
**Figure 4.2:** Diamond v1.0: Overview of the process of creating a tree testing study. [Screenshot made by the author of this thesis.]



Edit Study

Please provide content of your tree. Tree can also be imported as a tab-indented CSV file. Comma (,) and semicolon (;) are also supported as delimiters.

**Figure 4.3:** Diamond v1.0: Creating the tree for a tree testing study, either manually or by importing a tree from a TSV file. [Screenshot made by the author of this thesis.]



1. Where would you expect to find information about the history of the university?

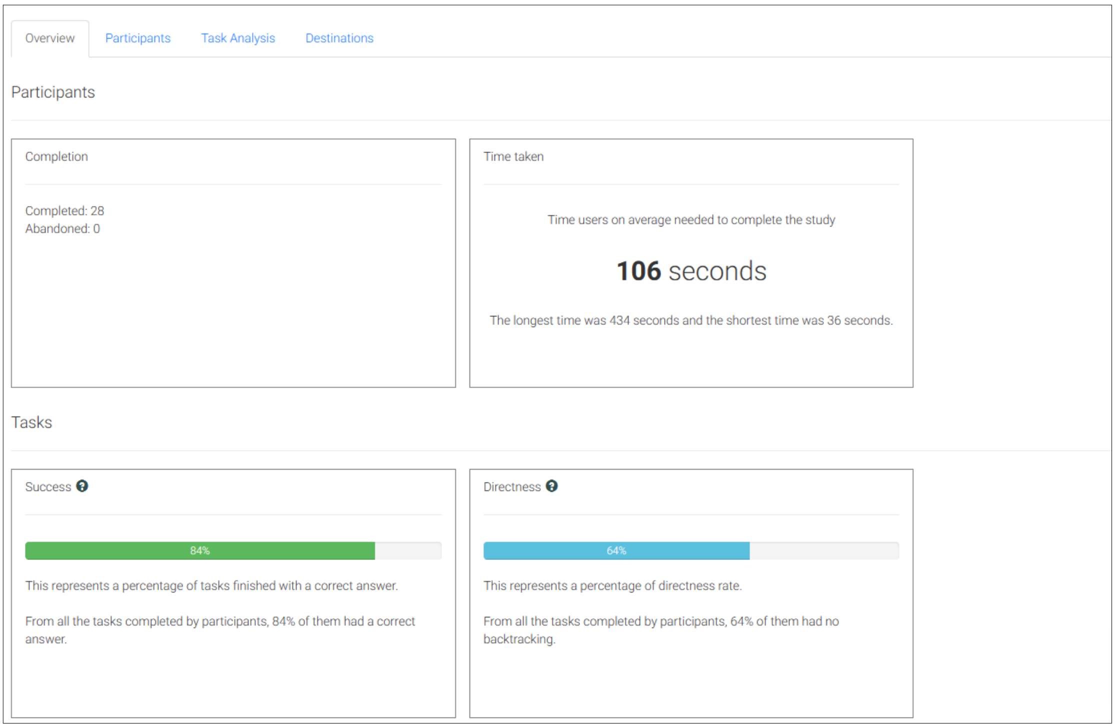
Selected answer: **History**

2. Where would you expect to find information about Master's degree programmes?

Selected answer: **Master's Degree Programmes**

- Home
  - TU Graz
    - Focus on TU Graz
    - University
    - Working at TU Graz
    - Services
    - Organisational Structure
  - Studying and Teaching
  - Research
  - Faculties and Institutes
  - Information for...

**Figure 4.4:** Diamond v1.0: Defining tasks for a tree testing study and specifying the correct answer for each task. [Screenshot made by the author of this thesis.]



Completion

Completed: 28
Abandoned: 0

Time taken

Time users on average needed to complete the study

**106** seconds

The longest time was 434 seconds and the shortest time was 36 seconds.

Success

This represents a percentage of tasks finished with a correct answer.

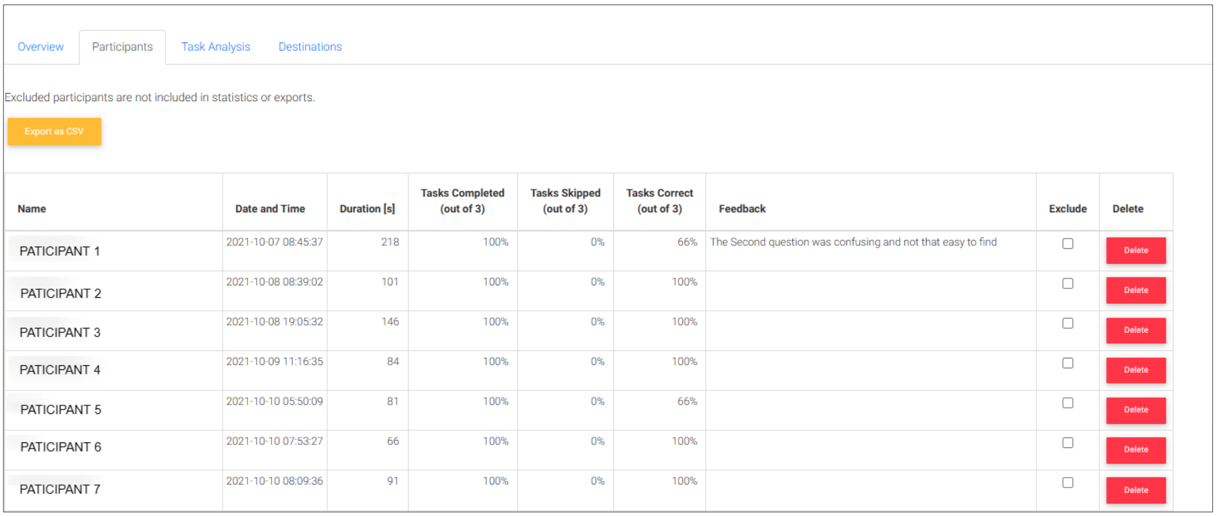
From all the tasks completed by participants, 84% of them had a correct answer.

Directness

This represents a percentage of directness rate.

From all the tasks completed by participants, 64% of them had no backtracking.

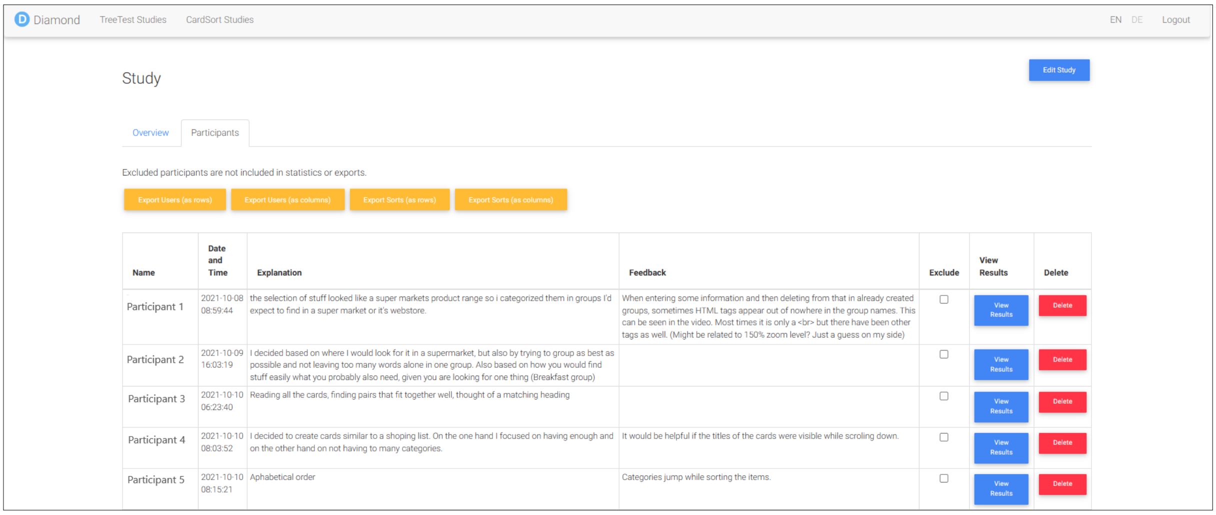
**Figure 4.5:** Diamond v1.0: The results page for a tree testing study shows an overview of various metrics. [Screenshot made by the author of this thesis.]



Excluded participants are not included in statistics or exports.

| Name | Date and Time | Duration [s] | Tasks Completed (out of 3) | Tasks Skipped (out of 3) | Tasks Correct (out of 3) | Feedback | Exclude | Delete |
|---|---|---|---|---|---|---|---|---|
| PATICIPANT 1 | 2021-10-07 08:45:37 | 218 | 100% | 0% | 66% | The Second question was confusing and not that easy to find | □ | Delete |
| PATICIPANT 2 | 2021-10-08 08:39:02 | 101 | 100% | 0% | 100% | | □ | Delete |
| PATICIPANT 3 | 2021-10-08 19:05:32 | 146 | 100% | 0% | 100% | | □ | Delete |
| PATICIPANT 4 | 2021-10-09 11:16:35 | 84 | 100% | 0% | 100% | | □ | Delete |
| PATICIPANT 5 | 2021-10-10 05:50:09 | 81 | 100% | 0% | 66% | | □ | Delete |
| PATICIPANT 6 | 2021-10-10 07:53:27 | 66 | 100% | 0% | 100% | | □ | Delete |
| PATICIPANT 7 | 2021-10-10 08:09:36 | 91 | 100% | 0% | 100% | | □ | Delete |

**Figure 4.6:** Diamond v1.0: Tabular overview of participants in a tree testing study. [Screenshot made by the author of this thesis.]

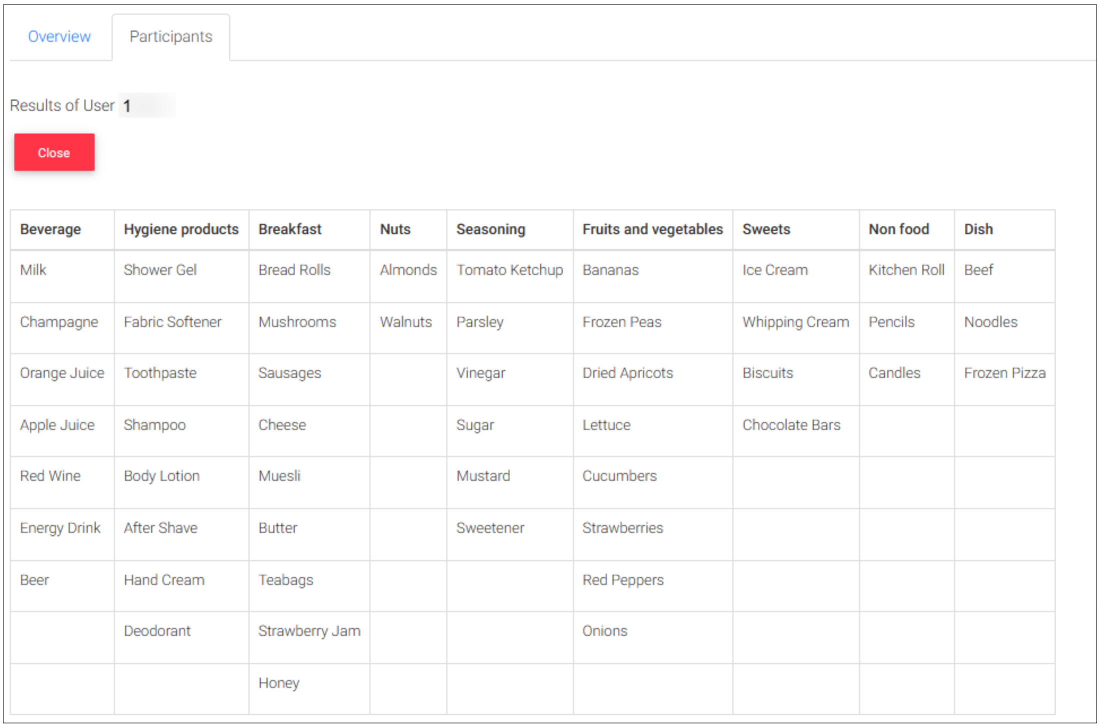


Study

Excluded participants are not included in statistics or exports.

| Name | Date and Time | Explanation | Feedback | Exclude | View Results | Delete |
|---|---|---|---|---|---|---|
| Participant 1 | 2021-10-08 08:59:44 | the selection of stuff looked like a super markets product range so i categorized them in groups I'd expect to find in a super market or it's webstore. | When entering some information and then deleting from that in already created groups, sometimes HTML tags appear out of nowhere in the group names. This can be seen in the video. Most times it is only a <br> but there have been other tags as well. (Might be related to 150% zoom level? Just a guess on my side) | □ | View Results | Delete |
| Participant 2 | 2021-10-09 16:03:19 | I decided based on where I would look for it in a supermarket, but also by trying to group as best as possible and not leaving too many words alone in one group. Also based on how you would find stuff easily what you probably also need, given you are looking for one thing (Breakfast group) | | □ | View Results | Delete |
| Participant 3 | 2021-10-10 06:23:40 | Reading all the cards, finding pairs that fit together well, thought of a matching heading | | □ | View Results | Delete |
| Participant 4 | 2021-10-10 08:03:52 | I decided to create cards similar to a shoping list. On the one hand I focused on having enough and on the other hand on not having to many categories. | It would be helpful if the titles of the cards were visible while scroling down. | □ | View Results | Delete |
| Participant 5 | 2021-10-10 08:15:21 | Aphabetical order | Categories jump while sorting the items. | □ | View Results | Delete |

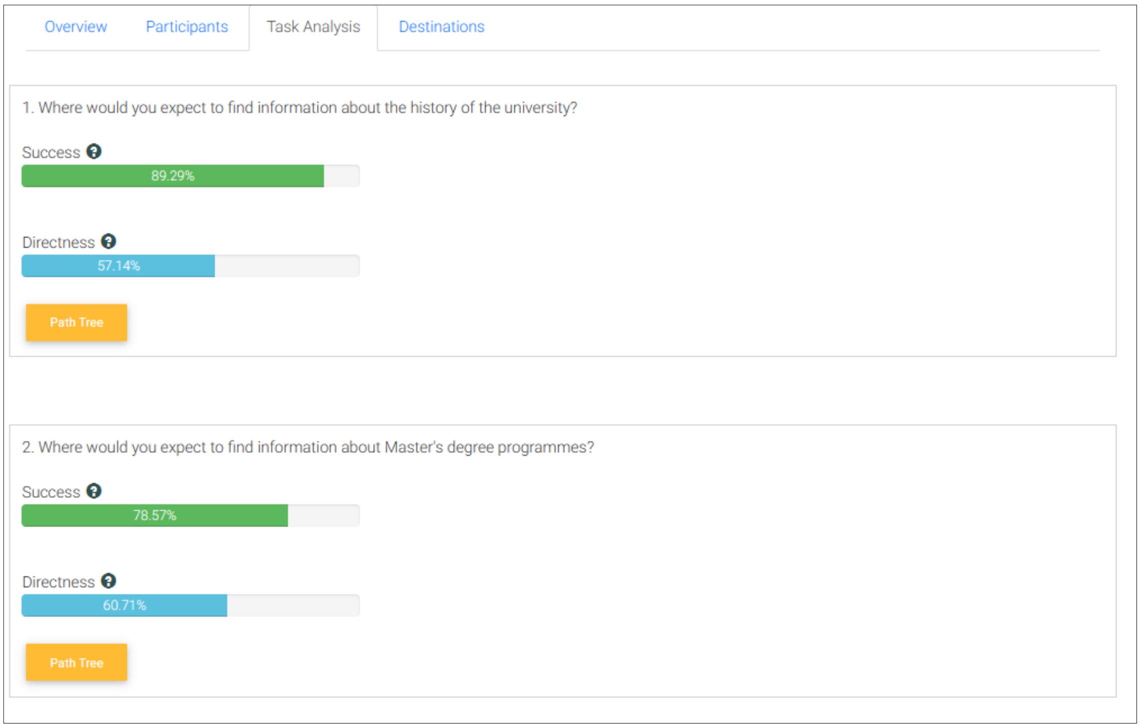
**Figure 4.7:** Diamond v1.0: Tabular overview of participants in a card sorting study. [Screenshot made by the author of this thesis.]



Results of User 1

| Beverage | Hygiene products | Breakfast | Nuts | Seasoning | Fruits and vegetables | Sweets | Non food | Dish |
|---|---|---|---|---|---|---|---|---|
| Milk | Shower Gel | Bread Rolls | Almonds | Tomato Ketchup | Bananas | Ice Cream | Kitchen Roll | Beef |
| Champagne | Fabric Softener | Mushrooms | Walnuts | Parsley | Frozen Peas | Whipping Cream | Pencils | Noodles |
| Orange Juice | Toothpaste | Sausages |  | Vinegar | Dried Apricots | Biscuits | Candles | Frozen Pizza |
| Apple Juice | Shampoo | Cheese |  | Sugar | Lettuce | Chocolate Bars |  |  |
| Red Wine | Body Lotion | Muesli |  | Mustard | Cucumbers |  |  |  |
| Energy Drink | After Shave | Butter |  | Sweetener | Strawberries |  |  |  |
| Beer | Hand Cream | Teabags |  |  | Red Peppers |  |  |  |
|  | Deodorant | Strawberry Jam |  |  | Onions |  |  |  |
|  |  | Honey |  |  |  |  |  |  |

**Figure 4.8:** Diamond v1.0: The results from one participant in a card sorting study, i.e. one sort, showing the categories and contained cards. [Screenshot made by the author of this thesis.]



1. Where would you expect to find information about the history of the university?

Success ?
89.29%

Directness ?
57.14%

2. Where would you expect to find information about Master's degree programmes?

Success ?
78.57%

Directness ?
60.71%

**Figure 4.9:** Diamond v1.0: An overview of success and directness rates for each task in a tree testing study. [Screenshot made by the author of this thesis.]

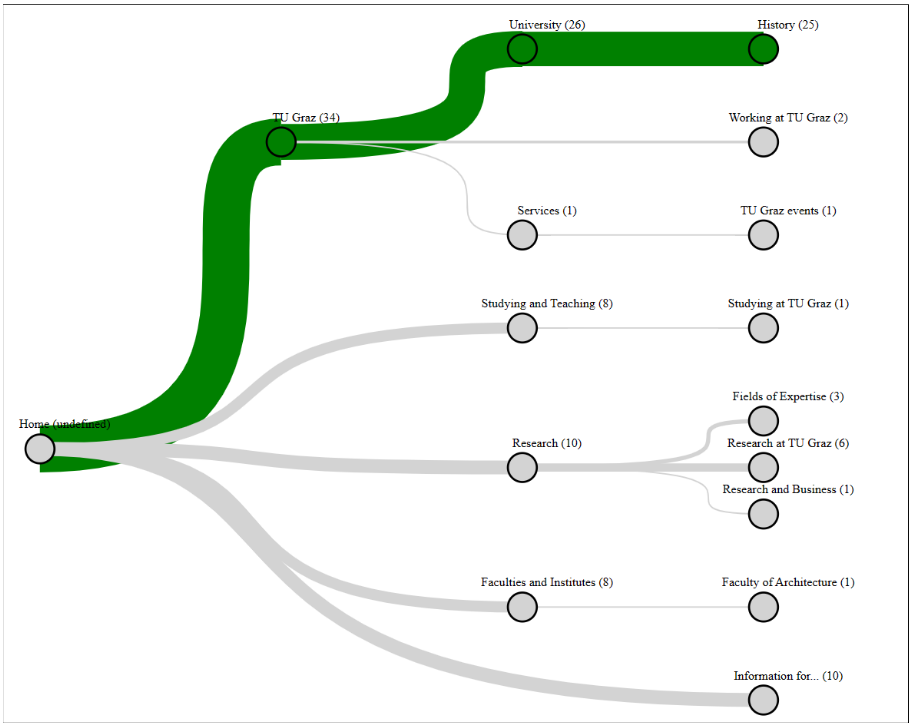
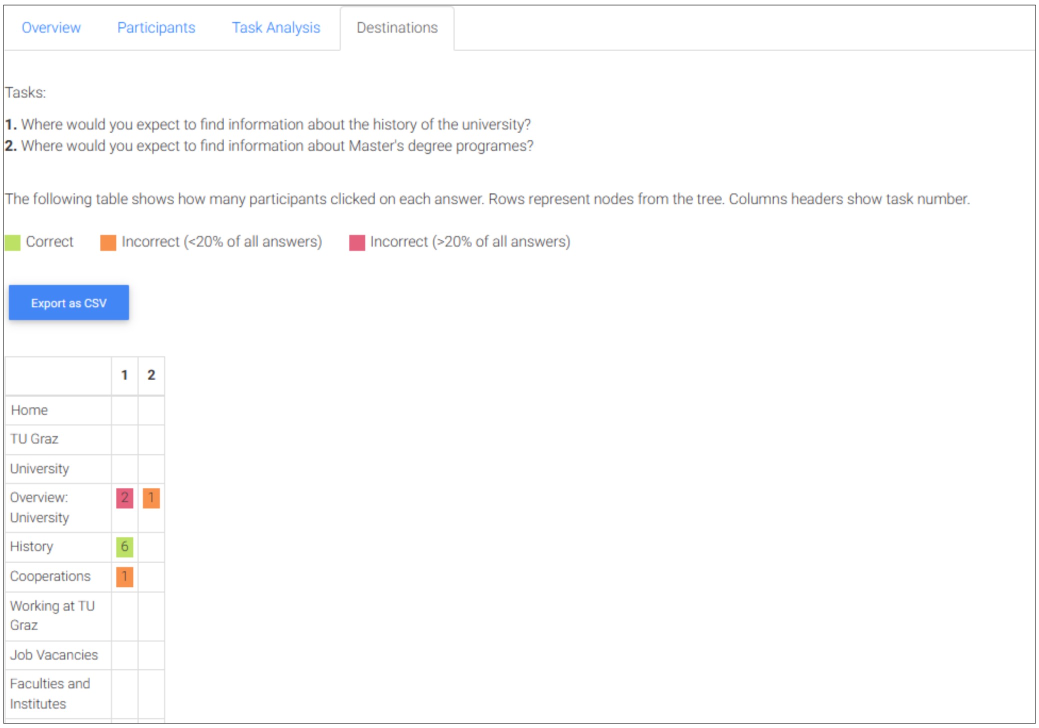


**Figure 4.10:** Diamond v1.0: A path tree aggregates the paths taken by participants in a tree testing task into a visual map. [Screenshot made by the author of this thesis.]



Tasks:

**1.** Where would you expect to find information about the history of the university?
**2.** Where would you expect to find information about Master's degree programes?

The following table shows how many participants clicked on each answer. Rows represent nodes from the tree. Columns headers show task number.

🟩 Correct 🟧 Incorrect (<20% of all answers) 🟥 Incorrect (>20% of all answers)

| | 1 | 2 |
|---|---|---|
| Home | | |
| TU Graz | | |
| University | | |
| Overview: University | 2 | 1 |
| History | 6 | |
| Cooperations | 1 | |
| Working at TU Graz | | |
| Job Vacancies | | |
| Faculties and Institutes | | |

**Figure 4.11:** Diamond v1.0: The destinations table for a tree testing study shows how many participants chose a particular node in the tree as their answer for each task. [Screenshot made by the author of this thesis.]

# Chapter 5

# Extending Diamond

Diamond version 1.0 had a number of issues and scope for improvement. This chapter describes the functionality implemented during this work for Diamond version 1.1. The chapter first discusses global changes which affected the entire app and how they were implemented to improve the overall user experience. It then discusses specific changes made to the card sorting and tree testing components of the app.

## 5.1 Global Changes

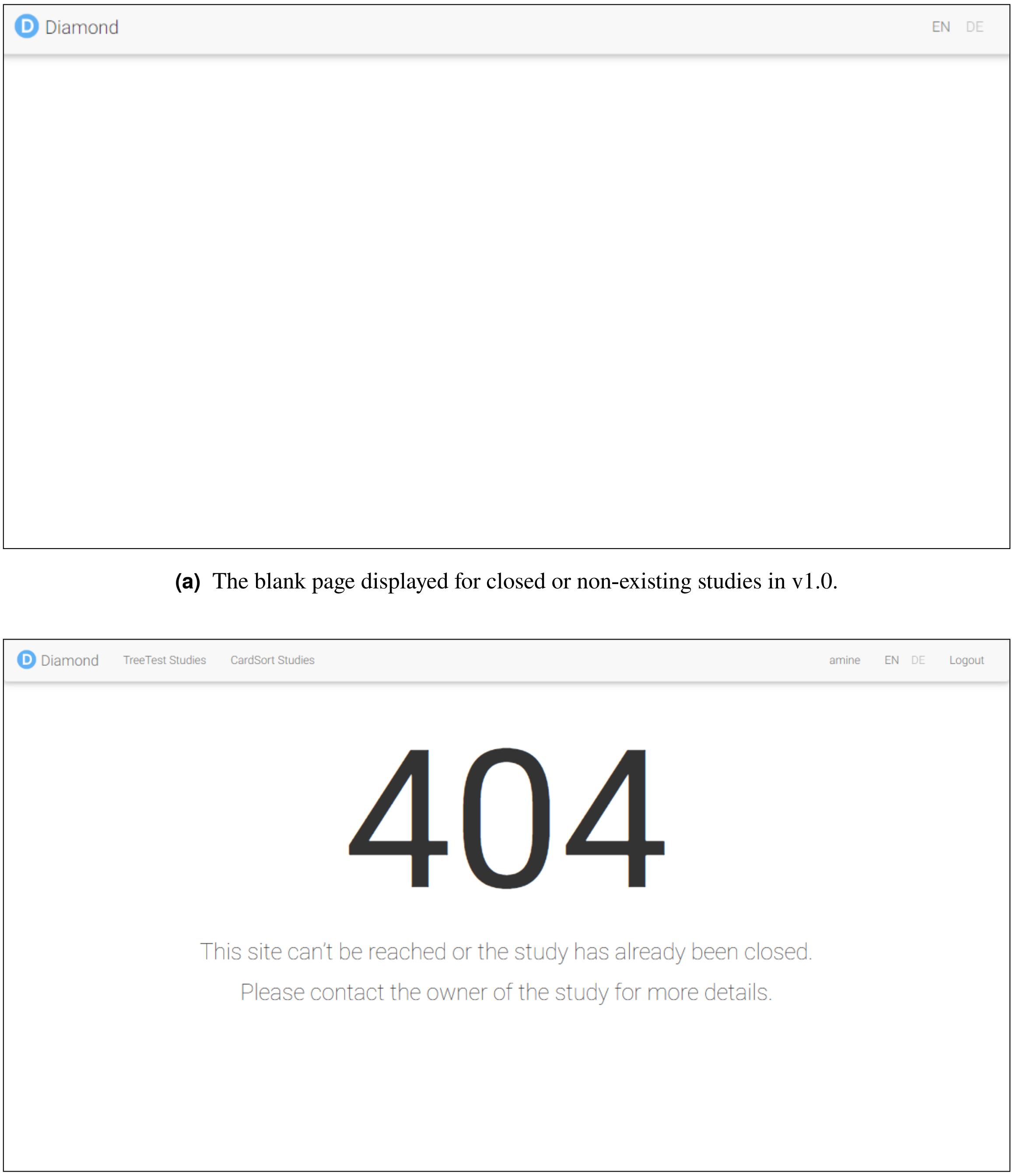
The first improvement was to refactor the code base. In v1.0, a study comprising multiple participants and their results was referred to as a *test*. The performance of an individual test user was referred to as a *result*.

This terminology was changed to refer to a study comprising multiple participants and their results as a *study*, and the results of an individual test user as a *test*. A study is created by a study owner, who then has access to the results of the individual tests. To make the source code of the app consistent, a refactoring of the database schemas, server base code, and the frontend was done to adjust the variable, class, and function names accordingly. Various files were also renamed.

Further, the server source code of v1.0 had all the models, controllers, and services in the same folder, making it hard to maintain. Instead, three folders were created to contain these respective files. The same restructuring was made to the frontend code base, where the components of card sorting and tree testing were split into two folders containing the respective components.

Another issue with Diamond v1.0 was that it displayed a blank page, shown in Figure 5.1a, when a user requested a URL for a non-existing study or a study which had already been closed. A new page was created for closed and non-existing studies to display an appropriate message for the user, as can be seen in Figure 5.1b.

Dealing with dates is not always easy as there are many different formats. Dates in the database of Diamond are stored in ISO 8601 [ISO 2023; Wikipidia 2023] standard format: `yyyy-MM-ddTHH:mm:ssZ`. To make dates easier for humans to read, Diamond displays them in the form `yyyy-MM-dd HH:mm:ss` . Angular offers a DatePipe to transform dates, however, the format `yyyy-MM-dd HH:mm:ss` used for Diamond is not available. For this reason, a customized DatePipe was implemented for Diamond.



404

This site can't be reached or the study has already been closed.

Please contact the owner of the study for more details.

**(a)** The blank page displayed for closed or non-existing studies in v1.0.

**(b)** The new error page displayed for closed or non-existing studies in v1.1.

**Figure 5.1:** The page displayed when a study is not found or is already closed. [Screenshots made by the author of this thesis.]

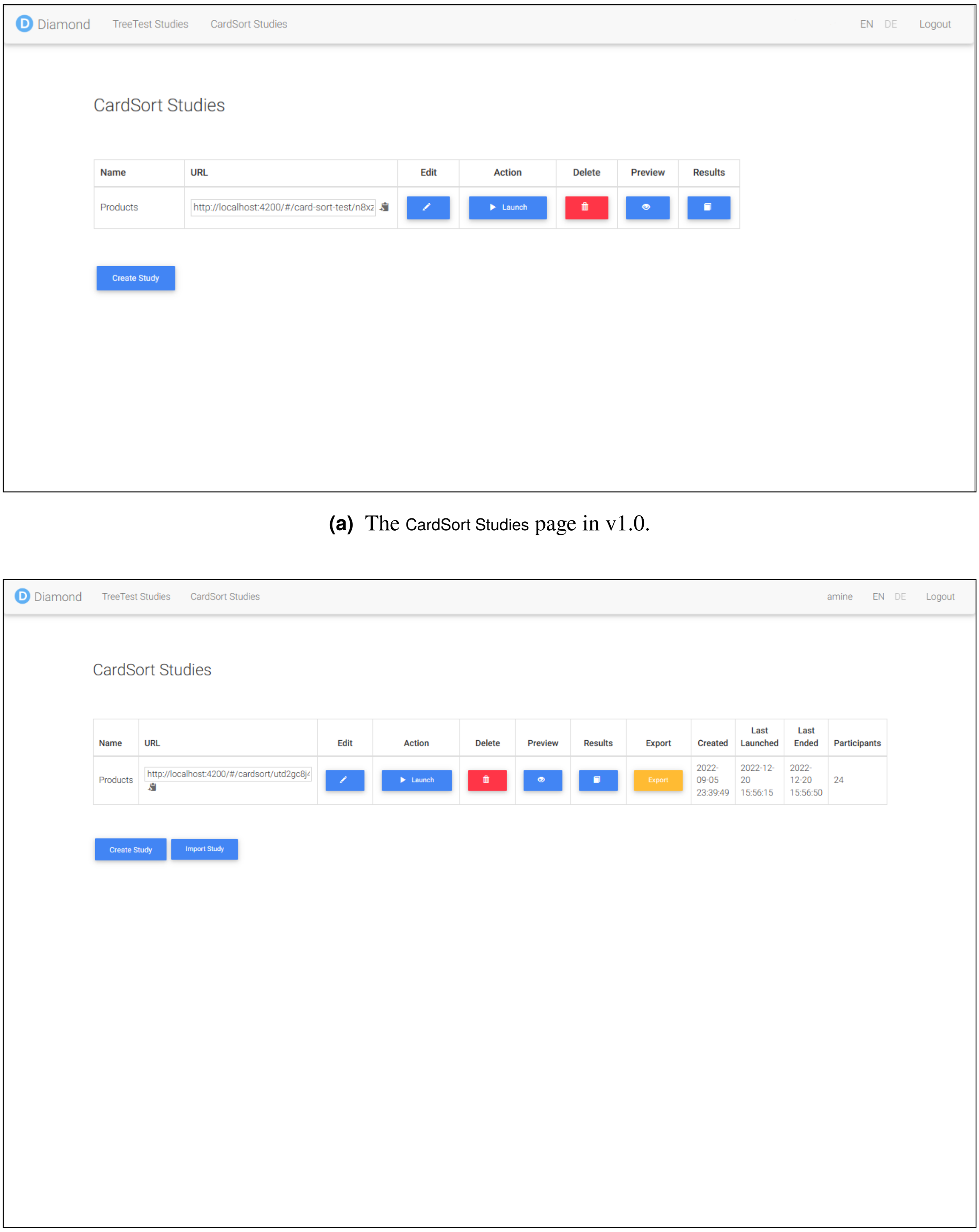
## 5.2 Card Sorting

One of the larger challenges for study owners using Diamond was the inability to import or export a study. This issue was solved by adding a feature of importing or exporting a study, as shown in Figure 5.2. Now, a study owner can simply click on the Export button to create an export of a study as a JSON file. The JSON file contains all the information pertaining to the study, as explained in Appendix A. A study can be imported by clicking the Import Study button.

Further, some important information for a study owner was missing in Diamond v1.0, in particular when a study was created, the last time it was launched, and when it was last ended. These new entries were added to the database schema, and are now displayed in the dashboard, so that the study owner can have more insights into their studies. Figure 5.2b shows the CardSort Studies page after adding the changes, where the extra information about each study are displayed in the table.

Another new feature that was implemented, was to give a study owner the ability to exclude a specific participant from further analysis. This is sometimes necessary, for example when a test goes wrong or a participant does not take the test seriously. Excluded tests are given a gray background in the study overview table, as shown in Figure 5.3. The introductory text was modified to indicate the total number of participants and the number of the excluded ones. The `excluded` flag is stored in database and it is consistent even when exporting the study.

In addition to exporting an entire study, Diamond also allowed specific tables of results, the users and the sorts, to be exported as CSV files. However, in v1.0, this functionality was somewhat buggy with rows messed up when the CSV file was opened. The bug was fixed and is working now without any issues.



CardSort Studies

| Name | URL | Edit | Action | Delete | Preview | Results |
|---|---|---|---|---|---|---|
| Products | http://localhost:4200/#/card-sort-test/n8xz | ✎ | ▶ Launch | 🗑 | 👁 | ▯ |

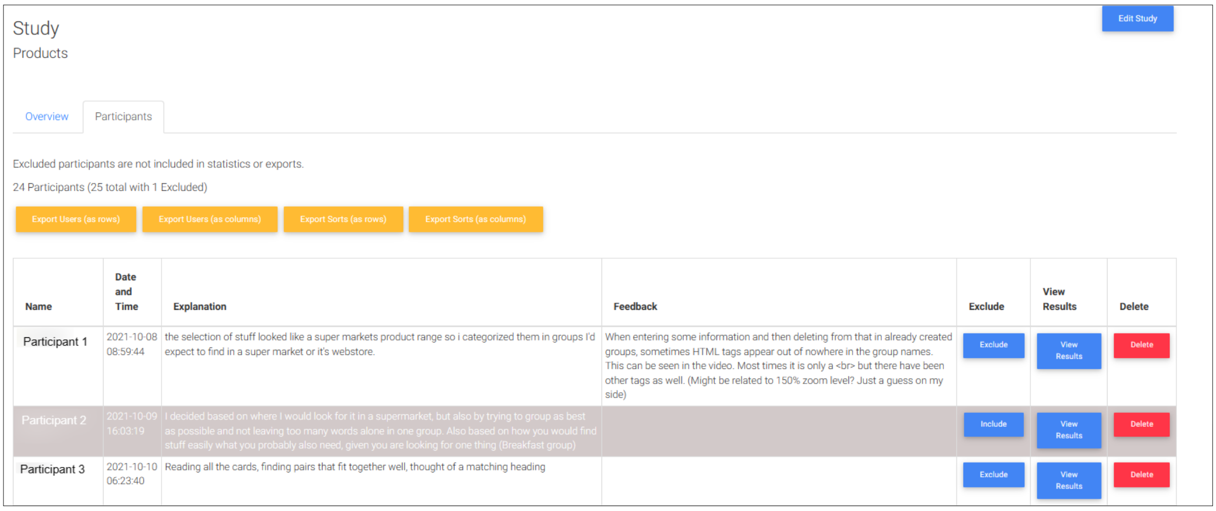
**(a)** The CardSort Studies page in v1.0.

CardSort Studies

| Name | URL | Edit | Action | Delete | Preview | Results | Export | Created | Last Launched | Last Ended | Participants |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Products | http://localhost:4200/#/cardsort/utd2gc8j | ✎ | ▶ Launch | 🗑 | 👁 | ▯ | Export | 2022-09-05 23:39:49 | 2022-12-20 15:56:15 | 2022-12-20 15:56:50 | 24 |

**(b)** The new CardSort Studies page in v1.1.

**Figure 5.2:** The CardSort Studies page before and after implementing the export and import of studies and adding more meta-information for each study. [Screenshots made by the author of this thesis.]



Excluded participants are not included in statistics or exports.

24 Participants (25 total with 1 Excluded)

| Name | Date and Time | Explanation | Feedback | Exclude | View Results | Delete |
|---|---|---|---|---|---|---|
| Participant 1 | 2021-10-08 08:59:44 | the selection of stuff looked like a super markets product range so i categorized them in groups I'd expect to find in a super market or it's webstore. | When entering some information and then deleting from that in already created groups, sometimes HTML tags appear out of nowhere in the group names. This can be seen in the video. Most times it is only a <br> but there have been other tags as well. (Might be related to 150% zoom level? Just a guess on my side) | Exclude | View Results | Delete |
| Participant 2 | 2021-10-09 16:03:19 | I decided based on where I would look for it in a supermarket, but also by trying to group as best as possible and not leaving too many words alone in one group. Also based on how you would find stuff easily what you probably also need, given you are looking for one thing (Breakfast group) | | Include | View Results | Delete |
| Participant 3 | 2021-10-10 06:23:40 | Reading all the cards, finding pairs that fit together well, thought of a matching heading | | Exclude | View Results | Delete |

**Figure 5.3:** CardSort Overview: Participants excluded by the study owner from the analysis are given a gray background. [Screenshot made by the author of this thesis.]

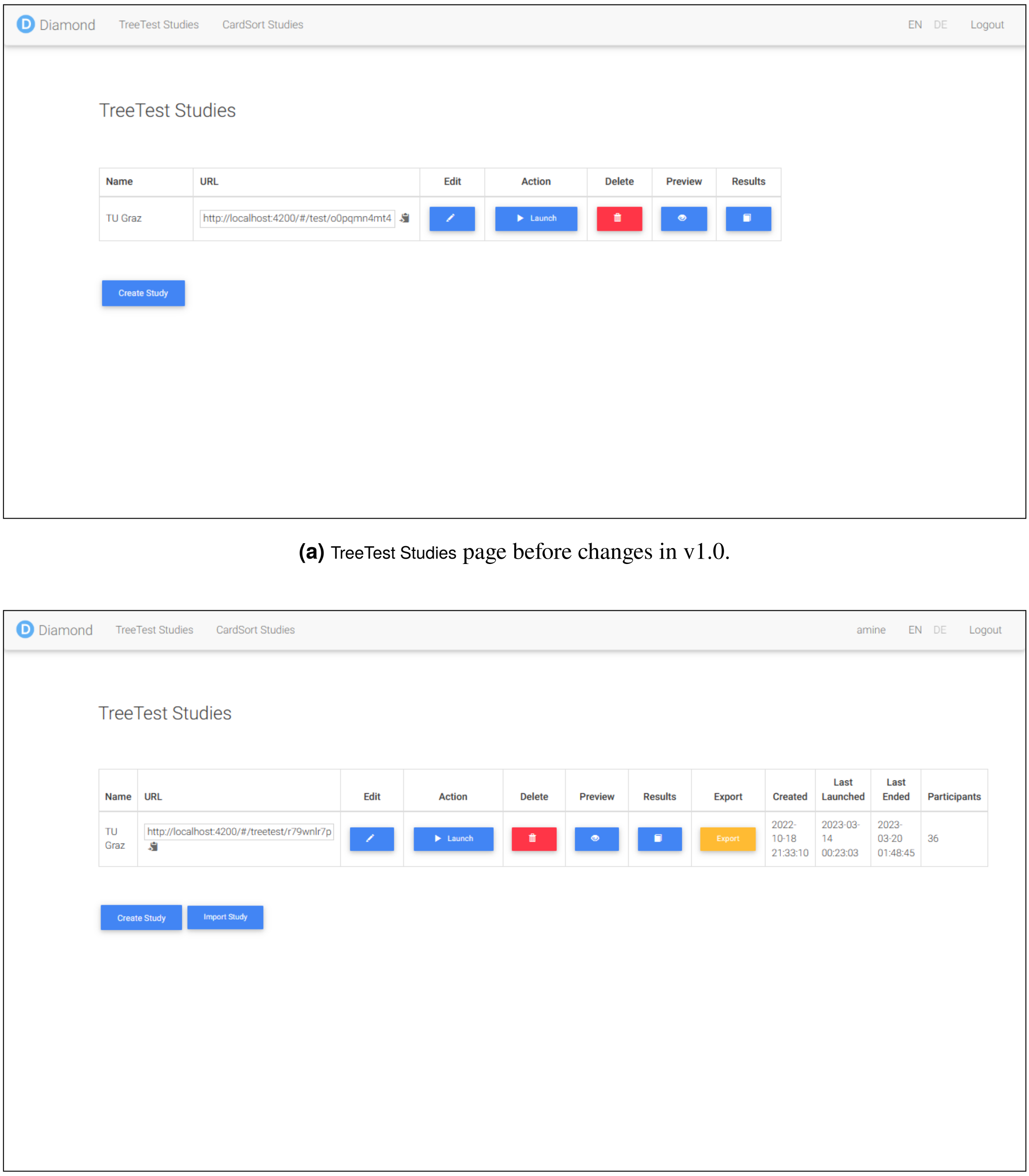
## 5.3 Tree Testing

The same features mentioned above regarding the importing and exporting a study, extra meta-information for a study (when created, last time launched and last time ended), and the ability to exclude a participant from a study were implemented for tree testing too. Figure 5.4 shows TreeTest Studies page in v1.0 and after the changes in v1.1, where a study owner can export a study, import a study, and see the extra meta-information. Furthermore, the URL of study now indicates whether it is a tree testing study or a card sorting study, as can also be seen in Figure 5.4.

Like in card sorting, it is now possible for a study owner to exclude a tree testing participant's test from further consideration. Figure 5.5 shows excluded participants with a gray background. The number of participants no longer includes excluded participants, and the numbers of each are now stated explicitly.

When doing a tree test, the user has the option of skipping a task, but Diamond v1.0 did not display the number of skipped tasks in the Destinations table of the study results. Figure 5.6 shows the new Destinations table in v1.1, now displaying the number of skipped tasks.

Another bug that was fixed is the responsiveness of the Overview and Task Analysis tabs in the results of a tree testing study on narrower screens. Further, the Task Analysis tab now displays the absolute number of successful and directly successful tests, in addition to the percentage rates, as shown in Figure 5.7. The task analysis statistics now also ignore excluded participants, which was not the case in v1.0.



TreeTest Studies

| Name | URL | Edit | Action | Delete | Preview | Results |
|---|---|---|---|---|---|---|
| TU Graz | http://localhost:4200/#/test/o0pqmn4mt4 | ✎ | ▶ Launch | 🗑 | 👁 | ■ |

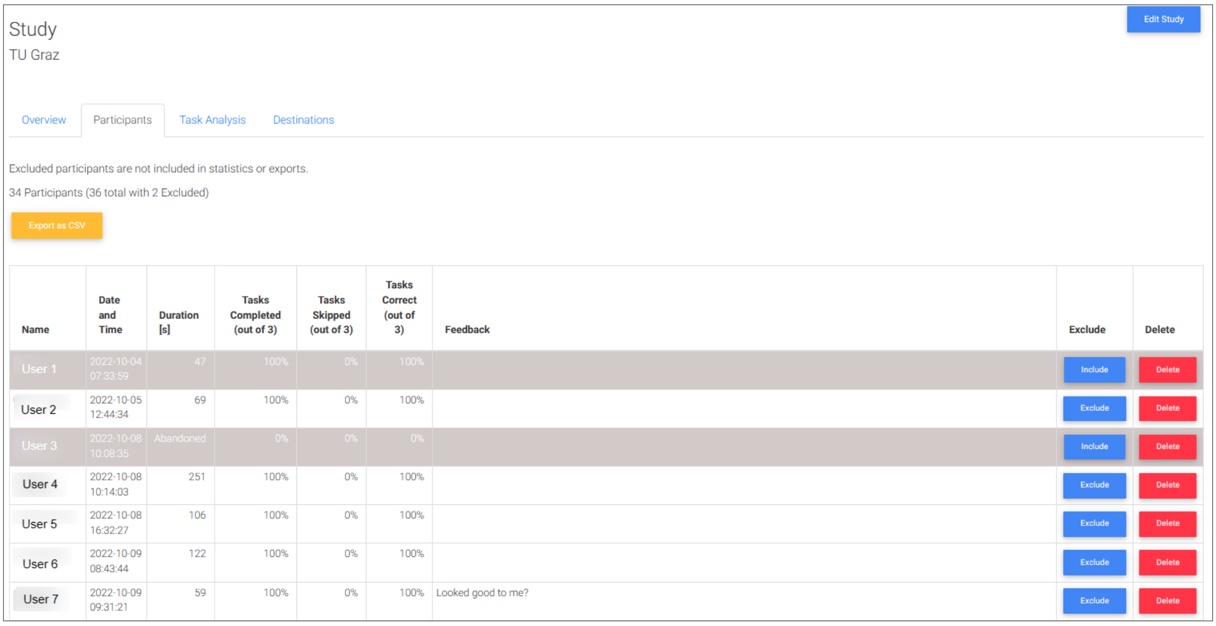
(a) TreeTest Studies page before changes in v1.0.

TreeTest Studies

| Name | URL | Edit | Action | Delete | Preview | Results | Export | Created | Last Launched | Last Ended | Participants |
|---|---|---|---|---|---|---|---|---|---|---|---|
| TU Graz | http://localhost:4200/#/treetest/r79wnlr7p | ✎ | ▶ Launch | 🗑 | 👁 | ■ | Export | 2022-10-18 21:33:10 | 2023-03-14 00:23:03 | 2023-03-20 01:48:45 | 36 |

(b) TreeTest Studies page after changes in v1.1.

**Figure 5.4:** The TreeTest Studies page before and after implementing features like the export and import of a study. [Screenshots made by the author of this thesis.]



Study

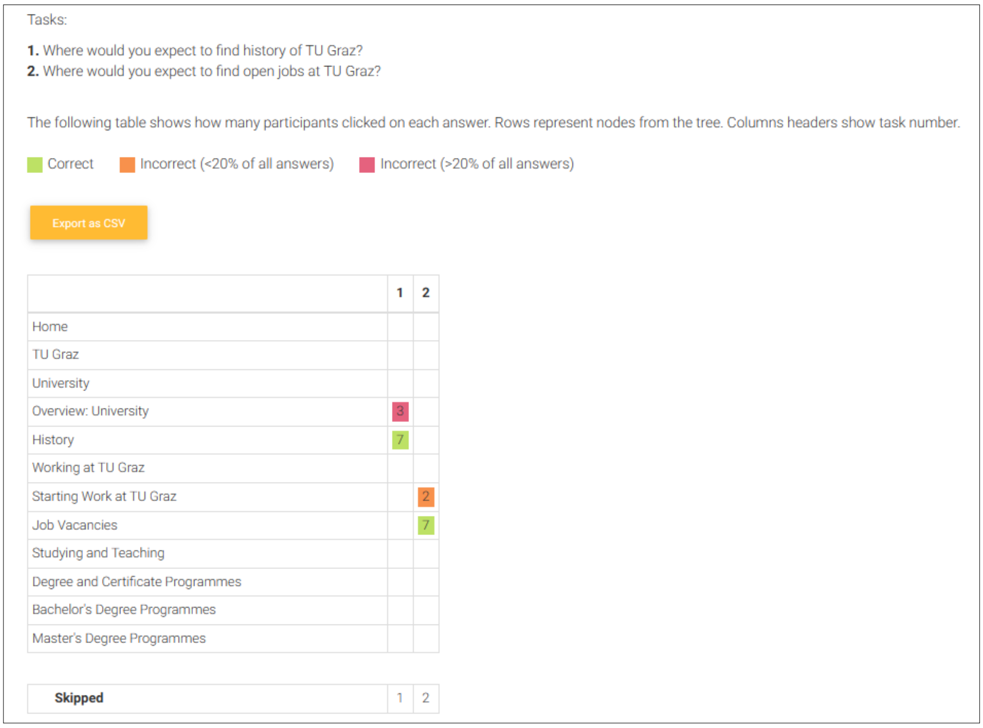
TU Graz

Excluded participants are not included in statistics or exports.

34 Participants (36 total with 2 Excluded)

| Name | Date and Time | Duration [s] | Tasks Completed (out of 3) | Tasks Skipped (out of 3) | Tasks Correct (out of 3) | Feedback | Exclude | Delete |
|---|---|---|---|---|---|---|---|---|
| User 1 | 2022-10-04 07:33:59 | 47 | 100% | 0% | 100% | | Include | Delete |
| User 2 | 2022-10-05 12:44:34 | 69 | 100% | 0% | 100% | | Exclude | Delete |
| User 3 | 2022-10-08 10:08:35 | Abandoned | 0% | 0% | 0% | | Include | Delete |
| User 4 | 2022-10-08 10:14:03 | 251 | 100% | 0% | 100% | | Exclude | Delete |
| User 5 | 2022-10-08 16:32:27 | 106 | 100% | 0% | 100% | | Exclude | Delete |
| User 6 | 2022-10-09 08:43:44 | 122 | 100% | 0% | 100% | | Exclude | Delete |
| User 7 | 2022-10-09 09:31:21 | 59 | 100% | 0% | 100% | Looked good to me? | Exclude | Delete |

**Figure 5.5:** TreeTest Overview: Participants excluded by the study owner from the analysis are given a gray background. [Screenshot made by the author of this thesis.]



Tasks:

**1.** Where would you expect to find history of TU Graz?
**2.** Where would you expect to find open jobs at TU Graz?

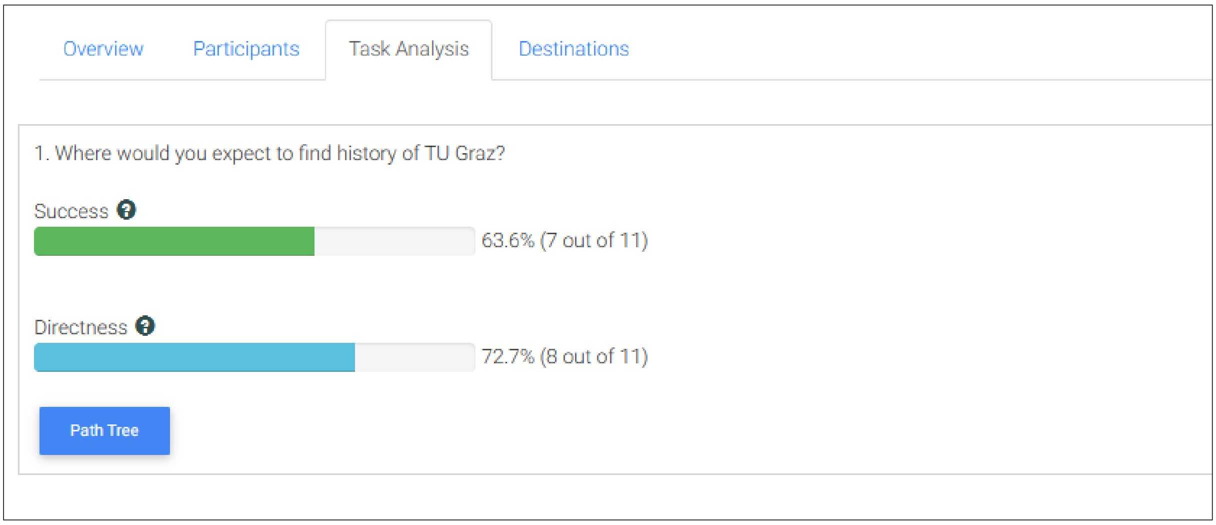
The following table shows how many participants clicked on each answer. Rows represent nodes from the tree. Columns headers show task number.

🟩 Correct 🟧 Incorrect (<20% of all answers) 🟥 Incorrect (>20% of all answers)

| | **1** | **2** |
|---|---|---|
| Home | | |
| TU Graz | | |
| University | | |
| Overview: University | 3 | |
| History | 7 | |
| Working at TU Graz | | |
| Starting Work at TU Graz | | 2 |
| Job Vacancies | | 7 |
| Studying and Teaching | | |
| Degree and Certificate Programmes | | |
| Bachelor's Degree Programmes | | |
| Master's Degree Programmes | | |

| **Skipped** | 1 | 2 |
|---|---|---|

**Figure 5.6:** The number of skipped tasks is included in the Destinations table of a tree testing study. [Screenshot made by the author of this thesis.]



1. Where would you expect to find history of TU Graz?

Success

63.6% (7 out of 11)

Directness

72.7% (8 out of 11)

**Figure 5.7:** The study owner can now see the absolute numbers of tests included in the calculation of success and directness for each task. [Screenshot made by the author of this thesis.]

# Chapter 6

# Future Work

Although the application has many useful features, there are still some areas which need improvement. One such area is the need for a landing page to provide study owners with all the necessary information and a tutorial on how the app works. This would be beneficial for users as the app continues to grow and more features are added.

To enhance the developer and the user experience, Diamond should have a design system to standardize the styles and components used by the app. Following these guidelines would ensure consistency throughout the app, making it easier for users to navigate.

The card sorting interface for participants could also be made more intuitive. In terms of accessibility, users should also be able to perform card sorting tests using only a keyboard.

Furthermore, for card sorting studies, tools are required to help study owners identify and manage mindsets, and create standardized names for group labels. Then, analysis tools like a similarity matrix, PCA, and a dendrogram, could be provided to help study owners explore the results. Owners should also be able to configure studies to run for a specific period. Moreover, the analysis of a card sorting study could be improved by adding the number of categories created by each user as meta-information to the table of results.

# Appendix A

# Card Sorting Study Export Format

This appendix describes the format of an exported card sorting study in Diamond, which is a JSON file. The export file contains all the data necessary to archive and restore a card sorting study: the study itself, the tests completed by users, and the results of each test, including how the cards were grouped and the names which were assigned to each group. An exported study can be re-imported on the same or a different Diamond server.

Listing A.1 shows the schema used to export a card sorting study. It contains an array of unsorted cards and some meta-information like the name, user (study owner), creation date, etc. A second array holds the tests done by participants. Each test has its own metadata (e.g. name of the participant) and the result as an array containing the name of each group (category) and a list of cards in that group. During this work, the `lastEnded` and `lastLaunched` flags were added to both card sorting and tree testing studies.

Listing A.2 shows an example of a card sorting study exported from Diamond. Participants in the study are asked to describe the strategy (mindset) they used to group the cards; this description is saved as a string. The `excluded` flag gives the exclusion status of that test. The default name of the exported file comprises the word `study` and the unique study ID, for example `study-n8xzwuutya8.json`.

```
{
cards: Array,             // all the cards in a study
name: String,
password: String,
launched: Boolean,
id: String,
createdDate:  Date,
user: String,
welcomeMessage: String,
instructions: String,
explanation: String,
thankYouScreen: String,
leaveFeedback: String,
subCategories: Boolean,
lastEnded: Date,
lastLaunched: Date,
tests: Array(             // tests done by users
  {
  id: String,
  createdDate: Date,
  finished: Boolean,
  username: String,
  timestamp: String,
  mindset: String,
  feedback: String,
  excluded: Boolean,
  results: Array({
    group_name: String,   // name of the category
    group_list: Array,    // cards in the category
    })
  }
)
}
```

**Listing A.1:** The schema of an exported card sorting study.

```
{
"cards": [
"Red Peppers",
"Mushrooms",
"Lettuce",
"Butter",
"Milk",
"Cheese"],
"name": "Products",
"launched": true,
"password": "password",
"id": "n8xzwuutya8",
"user": "amine",
"welcomeMessage": "Welcome to this study!",
"instructions": "Please group the provided cards as you see fit and
  name the created groups.",
"explanation": "Please explain how you decided on the way the cards
  should be grouped.",
"thankYouScreen": "Thank you for your participation!",
"leaveFeedback": "Your results are saved. You can give us your feedback (optional)
.",
"subCategories": true,
"lastEnded": "2023-03-20T01:48:45.565Z",
"lastLaunched": "2023-03-14T00:23:03.865Z",
tests: [
{
  "_id": "616008803746ca00169559fb",
  "id": "n8xzwuutya8",
  "finished": true,
  "username": "Mohamed",
  "timestamp": "2021-10-08 08:59:44",
  "feedback": "Easy to categorize",
  "mindset": "The selection of stuff looked like a supermarket's product range
    so I categorized them in groups I'd expect to find in a supermarket or its
    webstore."
  excluded: false,
  results: [{
    "group_name": "Vegetables",
    "group_list": ["Red Peppers", "Mushrooms", "Lettuce"]
  },
  {
    "group_name": "Milk products",
    "group_list": ["Butter", "Milk", "Cheese"]
  }]
}
]
}
```

**Listing A.2:** An example of real data from a card sorting study in Diamond.

# Appendix B

# Tree Testing Study Export Format

This appendix describes the format of an exported tree testing study in Diamond, which is a JSON file. The export file contains all the data necessary to archive and restore a tree testing study: the study, the tests, and the results of each test, including the answers selected for each task during the test. An exported study can be re-imported on the same or a different Diamond server.

Listing B.1 shows the schema used to export a tree testing study. It contains an array which holds the nodes of the tree: each node is an object with a name and an id, and the ids of its child nodes. The study has some meta-information like the name, user (study owner), creation date, etc. Each test has its own metadata (e.g. name of the participant) and the results as an array containing the answer, time taken and an array that holds the clicks done by the participants to get to the answer. The clicks are nodes of the tree and they are used by Diamond to construct the path tree. During this work, the `lastEnded` and `lastLaunched` flags were added to both card sorting and tree testing studies.

Listing B.2 shows an example of a tree testing study exported from Diamond. The results for each participant contain the clicks the user made on the tree while doing a task, the id of the selected answer, and the time spent to complete it. The default name of the exported file comprises the word `study` and the unique study ID, for example `study-r79wnlr7pic.json`.

```
{
  tree: Array,            // the tree for a study
  name: String,
  password: String,
  launched: Boolean,
  id: String,
  tasks: Array,
  user: String,
  welcomeMessage: String,
  instructions: String,
  thankYouScreen: String,
  leaveFeedback: String,
  leafNodes: Boolean,
  orderNumbers: Boolean,
  lastEnded: Date,
  lastLaunched: Date,
  isLaunchable: Boolean
  tests: Array(           // tests done by users
  {
  _id: String,            // the id of the study
  id: String,             // the id of the test
  finished: Boolean,
  username: String,
  timestamp: String,
  feedback: String,
  excluded: Boolean,
  createdDate: Date,
  results: Array({
    clicks: Array,        // path to the answer
    answer: String,       // id of the selected answer
    time: Float           // time spent on task in seconds
    })
  }
)
}
```

**Listing B.1:** The schema of an exported tree testing study.

```
{
  tree: [],
  "name": "TU Graz",
  "password": "password",
  "launched": false,
  "id": "r79wnlr7pic",
  "tasks": [
  {
    "text": "Where would you expect to find information about the history
    of the university?",
    "answer": "History",
    "id": "easv9c"
  }
  ],
  "user": "amine",
  "welcomeMessage": "Welcome to the study. Your answers can help improve our
  information hierarchy.",
  "instructions": "Please read the task, and then find the most appropriate
  location in the hierarchy.",
  "thankYouScreen": "Thank you for your participation!",
  "leaveFeedback": "Your results are saved. You can write us your feedback
  (optional).",
  "leafNodes": true,
  "orderNumbers": true,
  "lastEnded": "2023-03-20T01:48:45.565Z",
  "lastLaunched": "2023-03-14T00:23:03.865Z",
  "isLaunchable": true,
  tests: [
  {
    "excluded": true,
    "_id": "634f1b9776e1be2774995601",
    "id": "r79wnlr7pic",
    "finished": true,
    "username": "Mohamed",
    "timestamp": "2022-10-04 07:33:59",
    "feedback": "Good test",
    "createdDate": "2022-10-18T21:33:11.068Z",
    "results": [
    {
      clicks: [
      {
        "id": "nf4iqs",
        "text": "TU Graz",
        "icon": true,
        "parent": "root",
        "children": ["itiym8", "y2w3f"]
      },
      ...
      ],
      "answer": "easv9c",
      "time": 20.325
    }]
  }]
}
```

**Listing B.2:** An example of real data from a tree testing study in Diamond.

# Bibliography

Albert, Bill and Tom Tullis [2022]. *Measuring the User Experience*. 3$^{rd}$ Edition. Morgan Kaufmann, 07 Mar 2022. ISBN 0128180803 (cited on page 5).

Andrews, Keith [2021]. *Writing a Thesis: Guidelines for Writing a Master's Thesis in Computer Science*. Graz University of Technology, Austria, 03 Dec 2021. https://ftp.isds.tugraz.at/pub/keith/thesis (cited on page ix).

Andrews, Keith [2022]. *Information Architecture and Web Usability*. 18 Oct 2022. https://courses.isds.tugraz.at/iaweb/iaweb.pdf (cited on pages 3, 5, 10).

Angular [2023]. *Angular*. Google, 19 Jul 2023. https://angular.io/ (cited on page 17).

Balachandran, Kailaash [2023]. *KardSort*. 16 Jul 2023. https://kardsort.com/ (cited on page 9).

Brandl, Philipp, Tamara David, and Bernhard Kargl [2022]. *Diamond: Improving the Diamond Card Sorting Web Application*. Slide deck. Graz University of Technology, Austria, 26 Jan 2022. https://courses.isds.tugraz.at/iaweb/projects/ws2021/iaweb-ws2021-g1-project-diamond-slides.pdf (cited on page 19).

Diamond [2023]. *Diamond: Online Tree Testing and Card Sorting*. 30 May 2023. https://github.com/tugraz-isds/diamond (cited on page 1).

Ecma [2017]. *The JSON data interchange syntax*. Ecma International, Dec 2017. https://ecma-international.org/publications-and-standards/standards/ecma-404 (cited on page 15).

Egger, David, Ludwig Reinhardt, Stefan Schnutt, and Sebastian Überreiter [2023]. *Improving Diamond Tree Testing*. Slide deck. Graz University of Technology, Austria, 24 Jan 2023. https://courses.isds.tugraz.at/iaweb/projects/ws2022/iaweb-ws2022-g3-project-diamond-treetest-slides.pdf (cited on page 19).

Express [2023]. *Express JS*. 19 Jul 2023. https://expressjs.com/ (cited on page 17).

ISO [2023]. *ISO 8601 Date and time format*. 27 Sep 2023. https://iso.org/iso-8601-date-and-time-format.html (cited on page 27).

JSON [2023]. *Introducing JSON*. 27 Sep 2023. https://json.org/ (cited on page 15).

Kargl, Michaela, Ajdin Mehic, Zoran Prodanovic, and David Seywald [2018]. *Enhanced Card Sorting Analysis in R*. Graz University of Technology, 27 Jun 2018. https://courses.isds.tugraz.at/ivis/projects/ss2018/ivis-ss2018-g5-project-r-card-sort.pdf (cited on page 19).

Marcotte, Ethan [2010]. *Responsive Web Design*. A List Apart, 25 May 2010. http://alistapart.com/article/responsive-web-design (cited on page 15).

Maze [2023]. *Maze Card Sorting*. Maze, 16 Jul 2023. https://maze.co/guides/card-sorting (cited on page 7).

MDN [2023a]. *Responsive Design*. Mozilla Developer Network, 10 Feb 2023. https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Responsive_Design (cited on page 15).

MDN [2023b]. *SPA (Single-page application)*. Mozilla Developer Network, 10 Feb 2023. https://developer.mozilla.org/en-US/docs/Glossary/SPA (cited on pages 15, 17).

Mehic, Ajdin [2019]. *TreeTest: Online Tree Testing for Information Hierarchies*. Master's Thesis. Graz University of Technology, Austria, 14 Oct 2019. 96 pages. https://ftp.isds.tugraz.at/pub/theses/amehic-2019-msc.pdf (cited on page 19).

Mongo [2023]. *MongoDB*. 19 Jul 2023. https://mongodb.com (cited on page 17).

Node [2023]. *Nodejs*. 19 Jul 2023. https://nodejs.org/ (cited on page 17).

Oser, Christopher, Markus Ruplitsch, and Markus Stradner [2021]. *Diamond: An Online Tool for Card Sorting and Tree Testing*. Graz University of Technology, Austria, 01 Feb 2021. https://courses.isds.tugraz.at/iaweb/projects/ws2020/iaweb-ws2020-g3-project-diamond.pdf (cited on page 19).

OW [2023a]. *OptimalSort*. Optimal Workshop, 16 Jul 2023. https://optimalworkshop.com/optimalsort (cited on page 5).

OW [2023b]. *Similarity Matrix*. Optimal Workshop, 04 Aug 2023. https://optimalworkshop.com/learn/card-sorting-101-similarity-matrix (cited on page 5).

OW [2023c]. *Treejack*. Optimal Workshop, 12 Jul 2023. https://optimalworkshop.com/treejack (cited on page 11).

PBU [2023]. *ProvenByUsers Card Sorting*. Proven By Users, 16 Jul 2023. https://provenbyusers.com/provenbyusers-cardsort.php (cited on page 7).

PBUX [2023a]. *PlaybookUX Card Sorting*. 16 Jul 2023. https://playbookux.com/card-sorting (cited on page 7).

PBUX [2023b]. *PlaybookUX Tree Testing*. 16 Jul 2023. https://playbookux.com/tree-testing (cited on page 11).

Resmini, Andrea and Luca Rosati [2011]. *A Brief History of Information Architecture*. Journal of Information Architecture 3.2 (2011), pages 33–46. https://journalofia.org/volume3/issue2/03-resmini/jofia-0302-03-resmini.pdf (cited on page 3).

Schilb, Steffen [2006]. *Design, Implementation, and Evaluation of a Tool for Testing Hierarchy-Based Web Navigation Systems*. Master's Thesis. University of Bremen, 03 Sep 2006. 148 pages. http://steffenschilb.com/Master_thesis_Steffen_Schilb.pdf (cited on pages ix, 4).

Sherwin, Katie [2018]. *Card Sorting: Uncover Users' Mental Models for Better Information Architecture*. Nielsen Norman Group, 18 Mar 2018. https://nngroup.com/articles/card-sorting-definition (cited on page 3).

Spencer, Donna [2009]. *Card Sorting: Designing Usable Categories*. Rosenfeld Media, Apr 2009. ISBN 1933820020. https://rosenfeldmedia.com/books/card-sorting (cited on page 5).

Spencer, Donna [2014]. *A Practical Guide to Information Architecture*. 2nd Edition. UX Mastery, 2014. 443 pages. ISBN 0992538025. https://maadmob.com.au/speaking/books/practical-ia (cited on pages ix, 4).

UXtweak [2023a]. *UXtweak Card Sorting*. 16 Jul 2023. https://uxtweak.com/card-sort-tool (cited on page 7).

UXtweak [2023b]. *UXtweak Tree Testing*. 16 Jul 2023. https://uxtweak.com/tree-testing-tool (cited on page 11).

Whitenton, Kathryn [2017]. *Tree Testing Part 2: Interpreting the Results*. Nielsen Norman Group, 09 Jul 2017. `https://nngroup.com/articles/interpreting-tree-test-results` (cited on page 10).

Wikipidia [2023]. *ISO 8601*. 27 Sep 2023. `https://wikipedia.org/wiki/ISO_8601` (cited on page 27).