University of Duisburg-Essen
Faculty of Business Administration and Economics
Human-Computer Interaction Group

Bachelorprojekt

# IndiLearn: Individualized Exercise Selection in a Web-based Learning Platform using an IRT-based Recommender System

Vorgelegt der Fakultät Wirtschaftswissenschaften der Universität Duisburg-Essen (Campus Essen) von

**Jia Lu, Zhaoyu Wu, Kun Yang**

Hauptstraße 45, 45127 Essen

| | |
|---|---|
| **Course of Study:** | Angewandte Informatik |
| **Matriculation Number:** | 3030063, 3061066, 3082107 |
| **First Examiner:** | Prof. Dr. Stefan Schneegass |
| **Supervisor:** | Nick Wittig, M.Sc.<br>Noro Schloke, M.Sc. |

| | |
|---|---|
| **Date:** | 01. March 2023 |

## Abstract

The COVID-19 pandemic has prompted a global shift towards online teaching, disrupting the traditional mode of student learning in schools. As a result, schools around the world have started online teaching. The shift from traditional offline face-to-face teaching to the current online teaching has posed a significant challenge not only for teachers but also for students who are adapting to this new mode of learning. However, each student is unique and requires a customized approach to learning. Although modern educational software can provide student performance data, teachers often rely on personal experience to analyze each student's performance, leading to increased workload and decreased quality of analysis. In this context, our bachelor project "IndiLearn: Individualized Exercise Selection in a Web-based Learning Platform using an IRT-based Recommender System" aims to develop a learning system "IndiLearn" that uses a recommendation system as a tool to provide customized exercises in different subjects based on each student's performance. The system evaluates the difficulty of the practice questions and the user's performance, and automatically recommends more challenging tasks to reinforce their performance in a particular subject. The goal is to improve the quality of teaching and learning, reduce the teacher's workload, and increase students' interest in learning.



## Kurzfassung

Die COVID-19-Pandemie hat zu einer weltweiten Verlagerung auf den Online-Unterricht geführt und damit die traditionelle Art des Lernens in den Schulen durchbrochen. Infolgedessen haben Schulen auf der ganzen Welt mit dem Online-Unterricht begonnen. Die Umstellung vom traditionellen Offline-Face-to-Face-Unterricht auf den aktuellen Online-Unterricht stellt nicht nur für die Lehrkräfte eine große Herausforderung dar, sondern auch für die Schüler, die sich an diese neue Art des Lernens gewöhnen müssen. Jeder Schüler ist jedoch einzigartig und erfordert einen individuellen Lernansatz. Obwohl moderne Lernsoftware Daten über die Leistungen der Schüler liefern kann, sind die Lehrkräfte bei der Analyse der Leistungen jedes einzelnen Schülers oft auf ihre persönliche Erfahrung angewiesen, was zu einer erhöhten Arbeitsbelastung und einer geringeren Qualität der Analyse führt. Vor diesem Hintergrund zielt unser Bachelorprojekt "IndiLearn: Individualisierte Übungsauswahl in einer webbasierten Lernplattform mit einem IRT-basierten Recommender System" darauf ab, ein Lernsystem names "IndiLearn" zu entwickeln, das ein Empfehlungssystem als Werkzeug nutzt, um maßgeschneiderte Übungen in verschiedenen Fächern auf der Grundlage der Leistungen jedes einzelnen Schülers bereitzustellen. Das System bewertet den Schwierigkeitsgrad der Übungsfragen und die Leistung des Benutzers und empfiehlt automatisch anspruchsvollere Aufgaben, um die Leistung in einem bestimmten Fach zu verbessern. Ziel ist es, die Qualität des Lehrens und Lernens zu verbessern, die Arbeitsbelastung des Lehrers zu verringern und das Interesse der Schüler am Lernen zu steigern.

# Contents

# List of Figures

# List of Tables

# List of Listings

# 1 Introduction

With the increase in younger mobile device users and the push of the epidemic in previous years, online learning is becoming more and more popular [21]. Many students also use the online learning system to consolidate their knowledge after class. Students improve themselves by doing topics that match their abilities.

The learning system was expected to recommend topics that match the student's ability level. And this learning system can be operated easily and quickly. Students' data do not interfere with each other.

First, the system gets a sample of a certain number of students doing a certain number of questions in the background, and the sample marks the correctness or incorrectness of each student doing each question. The analysis of this sample needs to be done before the student enables the learning system. Because the system describes the relationship between a student's ability value and the probability of being able to answer a question correctly according to the Item Response Theory(IRT). Therefore, the system estimates the IRT model parameters for each question by Maximum Likelihood Estimation(MLE) [16]. Using MLE, the system describes the most likely characteristics of each topic, including the difficulty and discrimination of each question. After that, these questions are then stored in a database along with their parameters. Students choose the type of questions they prefer to answer, and their records are stored in a database for analysis, or they can view their data on their pages. The system uses the students' questioning behavior with the IRT model to estimate their ability and finally recommends them questions that match their ability value range.

To achieve these goals, the structure of the entire system must first be determined. On the one hand, there are not many main functions like login, select course and answer questions, that need to be implemented in the foreground of this system, and it focuses on the use of algorithms in the background. On the other hand, the database of this system is not complicated, and the Students from the front end do not need access to the questions from the database directly. In summary, using a three-tier(State-Logic-Display) structure [10] is most suitable for this system. The recommendation system completes front-end interaction with the Students through the web app. The back-end system is a collection of methods like filtering questions from the database and assuming students' abilities, which need to handle data requests from the front end. The database is responsible for storing students' personal information, question data, and records of students' work, etc.

After determining the three-tier system architecture, one also needs to determine what programming language to use to build the front-end and back-end of the system, and which database to use. Because we identified the students of a web app for front-end functionality. JavaScript can achieve complex functions on the web page, the web page shows you no longer simple static information, but real-time content updates. Students need to interact with the recommendation system in real-time, JavaScript is, therefore, a perfect programming language, which suits this requirement. And React.js is a very popular and suitable JavaScript library for simple web page building like this system. For back-end building, python is a common programming language, and for the database side, PostgreSQL is a convenient database to link with the Python back-end.

Once all this is done, one only needs to focus on how to implement the various functions. In the end, the system needs to be tested by several students to check whether it can accurately assess the student's ability value.

# 2 Background and Related Work

## 2.1 Recommender system

A recommender system or a recommendation system aims to provide users with personalized online product or service recommendations to deal with the increasing online information overload problem and improve customer relationship management [14]. It uses specific information filtering(IF) techniques to recommend information items that may be of interest to the user. These information items can be blogs, commercial products, movies, music, news, pictures, etc . [6].

One example system that uses this technique is the "Recommended for You" feature on Amazon. This feature suggests products to customers based on their past purchase history, and items they have viewed or added to their wishlist. The system analyzes the customer's behavior and preferences to generate personalized recommendations that are specific to them. These recommendations are tailored to each user and can vary based on their browsing and purchase history.

The First recommender system was created by Goldberg, Nichols, and Oki&Terry in 1992 [3]. They propose the term "collaborative filtering" for a commercial recommender system, called Tapestry, designed to recommend files from newsgroups to a group of users. Collaborative filtering, which analyzes usage data across users to find well-matched user-item pairs, has since been juxtaposed against the older methodology of content filtering, which had its original roots in information retrieval. In content filtering, recommendations are not "collaborative" in the sense that suggestions made to a user do not explicitly utilize information across the entire user base [24].

Currently, recommendation techniques can be classified into four categories [25]: content-based filtering(CBF), collaborative filtering(CF), Demographic Filtering(DF), and hybrid-based recommender systems.

**Content-based filtering** considers the user's preferences and interests, which are inferred from their interactions with the system and the content metadata, to generate personalized suggestions [26]. Content-Based filtering recommendations depend on the user's former choices. These algorithms aim to suggest items or product which are alike to the items that user enjoyed in the past or is looking at in the present day.

**Collaborative filtering** approaches build the system by considering the user's past behavior (rating is given to those items, previously parched or chosen items) and an additional similar decision made by different users, then use the system to calculate the item or else rating that the user may be interested in [3]. The basic idea behind collaborative filtering is that if two users have similar preferences for some items, they are likely to have similar preferences for other items as well. Collaborative filtering can be divided into user-based approaches and item-based approaches [28].

**Demographic Filtering** is a technique used in recommendation systems to generate personalized recommendations based on demographic information. This technique of recommendation system used user profile information like age, gender, demographic area, education, interests, and their opinion about rating items and find the common users who have similar rating items and interests“ [15].

**Hybrid recommender systems** combine two or more recommendation techniques to improve performance and address the limitations of individual techniques. Collaborative filtering is often combined with other techniques to avoid the ramp-up problem. Some methods used to combine techniques include: Switching, Mixed, Feature combination, Cascade, and Feature augmentation Meta-level [2].

## 2.2 Item response theory

Item response theory (IRT) is a psychometric theory widely used in education to calibrate and evaluate items in tests, questionnaires, and other instruments, and to rate subjects' abilities, attitudes, or other underlying characteristics. Today, all major educational tests, such as the Scholastic Aptitude Test (SAT) and the Graduate Record Examination (GRE), are developed through the use of item response theory because this approach can greatly increase the accuracy and reliability of measurement while significantly reducing assessment time and effort, particularly through computer adaptive testing [1]. Lord and Novick's (1968) classic book introduced the basic concepts of IRT, including the idea of a latent trait, item response functions, and the three-parameter logistic model [13]. Samejima (1969) developed the graded response model, an extension of the two-parameter logistic model that allows for multiple response categories [23]. Hambleton and Swaminathan's (1985) comprehensive treatment of IRT models introduced the Rasch model, the two-parameter logistic model, and the three-parameter logistic model [4]. Let $\theta_i$ denote the latent ability of student $i$, $a_j$ denote the discrimination parameter of item $j$, $b_j$ denote the difficulty parameter of item $j$, $c_j$ denote the guessing parameter of item $j$, and $X_{ij}$ is the binary event that indicates whether student $i$ correctly answered item $j$. The probability of student $i$ answering item $j$ correctly, given that the item parameters are $a_j$, $b_j$, $c_j$, and the latent ability of student $i$ is $\theta_i$ can be expressed as:

$$P(X_{i,j} = 1|\theta_i; a_j, b_j, c_j) = c_j + (1 - c_j)\frac{e^{a_j(\theta_i - b_j)}}{1 + e^{a_j(\theta_i - b_j)}}$$

The above equation is called the three-parameter model. When $c = 0$ and all values are equal (i.e., only considering item difficulty), the model is referred to as the one-parameter model (Rasch model). Similarly, when $c = 0$ (i.e., only considering item difficulty and discrimination), the model is referred to as the two-parameter model:

$$P(X_{ij} = 1|\theta_i; a_j, b_j) = \frac{e^{a_j(\theta_i - b_j)}}{1 + e^{a_j(\theta_i - b_i)}}$$

IRT has a long history in educational and psychological testing, but its utilization in recommendation systems is relatively new. One of the first works to apply IRT to recommendation systems was the study by Wang and Blei [27], who used IRT to model the likelihood of a user clicking on an item in a recommendation system. They showed that their IRT-based model outperformed other popular recommendation system models.

### 2.2.1 IRT Model and its application

IRT consists of two fundamental models, namely, the normal ogive model and the logistic regression model. The initial IRT model takes the form of the normal ogive model, but it employs an integral function that renders it impractical for parameter estimation and utilization purposes. Consequently, the logistic model has become the primary model used in practice. Recent developments in IRT-based recommendation systems include the use of adaptive IRT models, which imply that the employment of machine learning in combination with IRT could indeed alleviate the effect of the cold start problem in an adaptive learning environment [19], and the use of multidimensional IRT models, which mainly focus on constructing a multilevel multidimensional model to fit the hierarchical dataset about a large-scale English achievement test. Particular attention is given to assessing the correlation between multiple latent abilities and covariates [30]. Another trend is the use of Bayesian IRT models, which allow for uncertainty estimation in the model parameters [29]. Recently, IRT-based recommendation systems have been applied in various domains, such as virtual learning environments (VLE), social media, and clinical trials. For example, W L. Leite et al. evaluated their recommendation system in a large field experiment [8], both before and after school closures due to the COVID-19 pandemic. The results show evidence of the effectiveness of the video recommendation algorithm during the period of normal school operations, but the effect disappears after school closures. Implications for teacher orchestration of technology for normal classroom use and periods of school closure are discussed. IRT-based models have also contributed to the selection of a psychometric approach, Jennifer Petrillo et al. provided comparisons and a worked example of the item- and scale-level evaluations based on three psychometric methods used in patient-reported outcome development—classical test theory (CTT), item response theory (IRT), and Rasch measurement theory (RMT)—in an analysis of the National Eye Institute Visual Functioning Questionnaire (VFQ-25) [18].

### 2.2.2 Item Characteristic Curve

Item Characteristic Curve (ICC) is a graphical representation of the relationship between the probability of a correct response to an item and the level of the trait being measured. The trait being measured is usually a latent construct, such as ability or proficiency, that is not directly observable. The ICC is a key concept in item response theory [12]. The ICC shows the probability of a correct response to an item as a function of the level of the latent trait. It is based on the idea that items have different levels of difficulty and that individuals have different levels of the latent trait being measured. The ICC for an item is typically represented as an S-shaped curve, where the x-axis represents the level of the trait being measured, and the y-axis represents the probability of a correct response to the item. The IRT model uses the ICC to estimate the level of the latent trait for each individual based on their response to a set of test items. The model assumes that the probability of a correct response to an item depends on both the level of the latent trait and the item's characteristics, such as its difficulty level or discrimination power. The IRT model can be used to estimate the difficulty and discrimination parameters of each item, as well as the level of the latent trait for each individual.

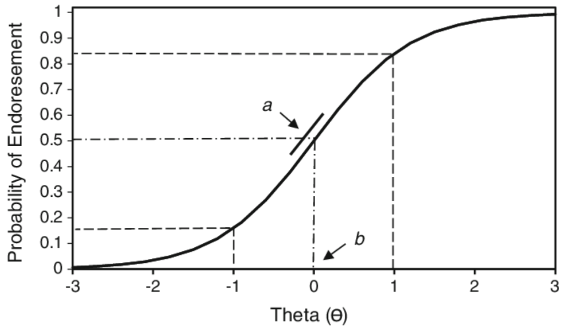



**Figure 2.1:** $a, b$ indicate the discrimination and difficulty of the item accordingly. Source: [17]

# 3 Concept

## 3.1 Content-based filtering

The two basic entities which appear in our Recommender System are the user (students who want to improve their study efficiency and academic performance) and the item (Exercises with different difficulty levels). A user is a person who utilizes the Recommender System providing his opinion about various items and receiving recommendations about new items from the system.

IndiLearn uses the three-phase process, this theory was proposed by Isinkaye et al [7]. The first phase is information collection, in this phase, relevant information of users is collected to generate a user profile or model for the prediction tasks. In IndiLearn, this information includes the user's name, ability value, and the subjects the user wishes to study. By analyzing this model, user interests, preferences, and learning styles are captured which can be used to provide personalized recommendations. The second phase is the learning phase. In this phase, we used context-based filtering based on Item Response Theory (IRT) theory to evaluate the difficulty of the practice questions and the user's performance. The third and final phase is the prediction/recommendation phase. It recommends or predicts what kind of items the user may prefer. This can be made either directly based on the dataset collected in the information collection phase which could be memory-based or model-based or through the system's observed activities of the user. This is the most important phase of a recommendation system. It ensures that the user will be offered exercises of appropriate difficulty.

IndiLearn uses content-based filtering as a recommendation technique. Content-based filtering is a suitable recommendation system technique for IndiLearn for several reasons:

**Personalized recommendations:** IndiLearn is designed to provide personalized exercises to each student based on their proficiency level and learning needs. Content-based filtering uses the proficiency level of the student and the difficulty level of the exercise to recommend the most suitable items for each student. This approach ensures that each student receives personalized exercises that match their level of proficiency.

**Item Response Theory (IRT):** IndiLearn uses IRT to determine the student's proficiency level. IRT is a statistical model that is widely used in educational testing to measure the ability of the test-taker. Content-based filtering uses the proficiency level obtained from IRT to recommend exercises that match the student's ability level. This approach ensures that the recommended exercises are not too easy or too difficult for the student.

**Accuracy:** Content-based filtering is an accurate recommendation system technique for IndiLearn. It analyzes the content of the exercises and the proficiency level of the student to identify the best-matching exercises to be recommended. This approach ensures that the recommended exercises are relevant to the student's learning needs and interests.

In IndiLearn, each user is assigned an ability value, while each exercise is assigned a difficulty value. The system uses Item Response Theory (IRT) to determine the student's proficiency level by analyzing their correctness on exercises of different difficulty levels. This proficiency level is then used to recommend more challenging exercises to the student. The content-based filtering algorithm compares the difficulty value of the exercises with the proficiency level of the student to identify the best-matching items to be recommended. This approach allows the system to provide students with individualized exercises that are based on their level of proficiency and learning needs.

## 3.2 Assessing Latent Traits in Education using Item Response Theory

The most frequently used models in applications are the Rasch and two-parameter model [1]. Our Indilearn uses a two-parameter model, each exercise is assigned a difficulty parameter value and a discrimination parameter. In contrast to the other two models, the two-parameter model can account for differences in item discrimination across items, which is a limitation of the Rasch model that assumes equal discrimination across all items. This is particularly useful when some items are better at discriminating between individuals with different levels of latent traits than others [5]. In such cases, using a Rasch model may lead to biased estimates of individuals' latent trait levels, as the model assumes that all items have the same ability to discriminate. Second, the two-parameter model is simpler and more parsimonious than the three-parameter model, which includes a guessing parameter to account for random guessing. If there is no reason to suspect that guessing is a significant factor in the data, then using a three-parameter model may lead to overparameterization and decreased model fit. Third, the two-parameter model can provide more information about the psychometric properties of the items, such as the item discrimination parameter. This information can be useful for item selection and development, as it can help identify items that are particularly effective at discriminating between individuals with different levels of latent traits.

In IndiLearn, ICC is a fundamental tool used to identify the most suitable exercises for each student based on their level of proficiency. Specifically, the ICC predicts the probability of a user's correct response to a given question difficulty level, which in turn helps determine the difficulty of the exercises to be recommended. The ICC is generated using item parameters, including item difficulty and discrimination, which are estimated from the responses of a large sample of individuals to the item. When a student responds to a test item, their response can be evaluated in terms of the ICC for that item. If the response is correct, the probability of a correct response on the ICC is used to estimate the student's level on the latent trait:

$$P(X_{i,j} = 1|\theta_i; a_j, b_j) = \frac{e^{a_j(\theta_i - b_j)}}{1 + e^{a_j(\theta_i - b_j)}}$$

If the response is incorrect, the probability of an incorrect response displayed on the reverse ICC (i.e., 1 minus the probability of correct response) is used instead:

$$P(X_{i,j} = 0|\theta_i; a_j, b_j) = 1 - \frac{e^{a_j(\theta_i - b_j)}}{1 + e^{a_j(\theta_i - b_j)}}$$

the ICC provides information about the item's ability to distinguish between individuals with different levels of the trait being measured. When a student answers all items correctly or incorrectly, their estimated level on the latent trait is the value on the x-axis at the point on the ICC where the correct rate reaches 50 percent. If a student answers both correctly and incorrectly, their estimated level of the latent trait is the value on the x-axis of the vertex of the resulting curve generated by multiplying the curves for correct and incorrect responses.

Furthermore, By analyzing the ICC for each exercise, IndiLearn can determine the level of difficulty at which a student has a 40-55 percent chance of answering correctly. This information is then used to recommend exercises that are appropriately challenging for each student's level of proficiency. As a result, the system can provide personalized exercise recommendations that are both challenging and achievable, leading to improved learning outcomes.

## 3.3 Maximum Likelihood Estimation

Maximum Likelihood Estimation(MLE), in layman's terms, is the use of known information about sample outcomes to invert the values of model parameters that are most likely (with maximum probability) to lead to those sample outcomes. MLE provides a way to evaluate the model parameters given the observed data, i.e., the model is fixed, and the parameters are unknown. In the context are the difficulty parameter and the discrimination parameter from IRT. Let's first look at the likelihood function:

$$p(x|\theta)$$

which has two factors, x and $\theta$. x denotes a specific piece of data and $\theta$ denotes the parameters of the model(IRT). If $\theta$ is known for sure and x is a variable, this function is called the probability function, which describes what is the probability of occurrence for different sample points x. If x is known with certainty and $\theta$ is a variable, this function is called the likelihood function, which describes the probability of occurrence of the sample point x for different model parameters.

# 4 Implementation

## 4.1 System Design

### 4.1.1 Requirement

In order to develop a successful recommendation system for students to identify their abilities and recommend relevant tasks based on their skill level, we have prioritized the system's features using the "must have", "should have" and "could have" framework. Our critical "must have" requirements include the ability to provide individual training in different subjects, student and teacher accounts, competency level analysis, and the ability for students to work on exercises. Our "should have" requirements include describing student preferences, the ability for teachers to divide tasks and collect answers through the system, the option for students to select tasks themselves, and data storage capabilities. Finally, our "could have" features include creating reports for students, providing relevant knowledge points for wrong tasks, allowing students to download their own reports, and providing a feedback system for students to give feedback to teachers. By prioritizing these critical requirements first, we can ensure the success of the project while making the most efficient use of resources.

### 4.1.2 Milestone

During the M0 period, we mainly prepare the program, including the theory to be applied and its rationale, determine the system structure and determine the programming language to be used.

During M1, we needed to set up the system and implement basic functions such as basic login, switching between web pages and interaction between students and the system. Various database reading and writing methods also needed to be implemented.

In the M2 phase, we need to apply IRT algorithm for deriving students' ability values, and then the system recommends suitable topics based on students' ability values. We also write scripts to generate questions for students to answer.

In the M3 phase, we need to apply the MLE to analyze the sample of questions done by students, and then derive the IRT parameters for each question. At the same time, we also need to complete the report.

| Milestone | ID | Description | Start-Time | Done-Time |
|---|---|---|---|---|
| M0 | 0.1 | Preparation | 03.11.2022 | 30.11.2022 |
| | 0.1.1 | Problem Statement and Research Question | 07.11.2022 | 17.11.2022 |
| | 0.1.2 | Reference | 10.11.2022 | 21.11.2022 |
| | 0.1.3 | Architecture | 10.11.2022 | 17.11.2022 |
| | 0.1.4 | Proposal | 10.11.2022 | 30.11.2022 |
| | 0.2 | Framework | 17.11.2022 | 30.11.2022 |
| | 0.2.1 | Component Diagram | 17.11.2022 | 30.11.2022 |
| | 0.2.2 | Class Diagram | 26.11.2022 | 30.11.2022 |
| | 0.2.3 | Paper Prototype | 26.11.2022 | 23.11.2022 |
| M1 | 1.1 | Database | 01.12.2022 | 21.12.2022 |
| | 1.2 | Login Page | 01.12.2022 | 06.01.2023 |
| | 1.2.1 | Register Page | 01.12.2022 | 06.01.2023 |
| | 1.3 | Navigation Menu | 01.12.2022 | 14.12.2022 |
| | 1.4 | Answer Page | 01.12.2022 | 06.01.2023 |
| | 1.5 | Back-end | 20.12.2022 | 01.01.2023 |
| M2 | 2.1 | Script for Data Generation | 12.01.2023 | 19.01.2023 |
| | 2.2 | Item-Response-Theory | 19.01.2023 | 09.02.2023 |
| M3 | 3.1 | Report | 01.03.2023 | 23.03.2023 |
| | 3.2 | Maximum Likelihood Estimation | 01.03.2023 | 16.03.2023 |
| | 3.3 | Test | 16.03.2023 | 23.03.2023 |

**Table 4.1:** Milestone Graphic

### 4.1.3 Paper Prototype

In Figure 4.1a, the user inputs his username and password in the text file.

If the username does not exist in Database, the user should register in Figure 4.1b first with username and password.

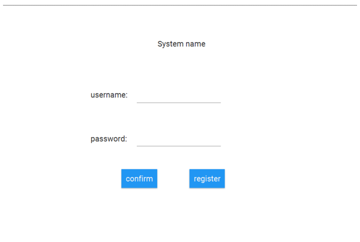
The homepage consists of "welcome", "my course", "join course" and "my profile" sub-pages. User can switch between the sub-pages using the menu.

In Figure 4.1c, the users can find themselves the menu above.

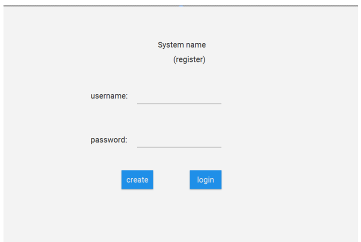
User can start to answer questions in Figure 4.1d, choosing the saved question category from Figure 4.1e, which provides the available category to the user. A user can view his score in Figure 4.1f, after he has done some questions.

In Figure 4.1g, user answers the question with the input bar. The user is also able to switch between questions, so that he can change his answer.

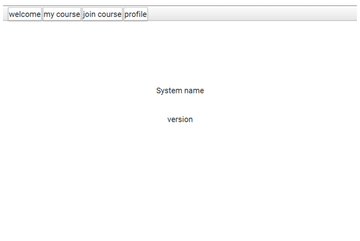
After answer a set of questions, user can review his answer in Figure 4.1h.



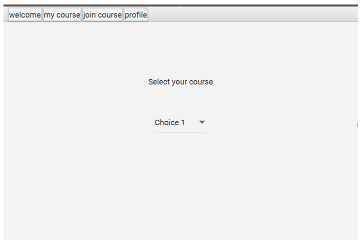

(a) login



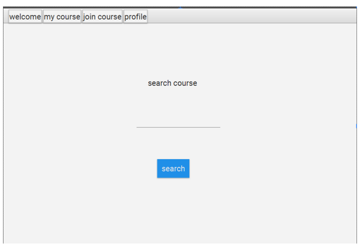

(b) register



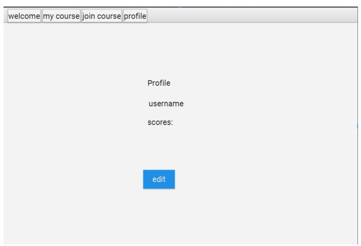

(c) welcome



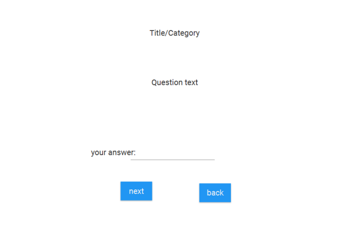

(d) my course



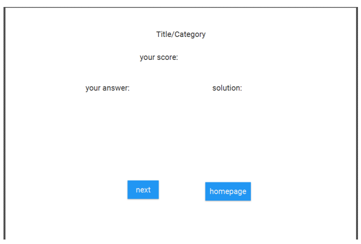

(e) join course




(f) profile




(g) answer




(h) score

**Figure 4.1:** Papier prototype

### 4.1.4 IndiLearn architecture

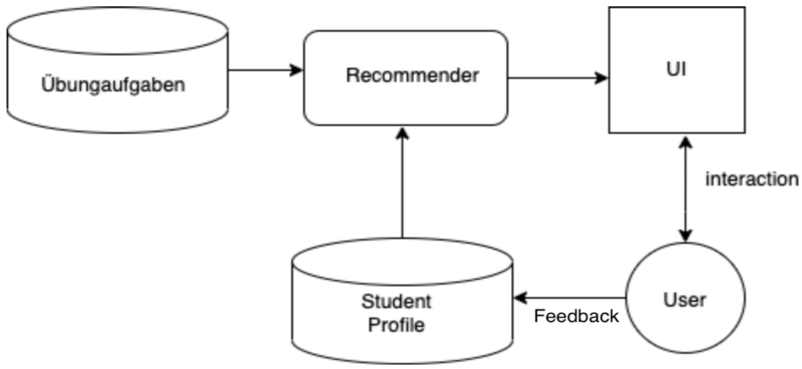



**Figure 4.2:** The Indilearn architecture

Figure 4.2 shows the Indilearn architecture. A student profile is learned from feedback provided by the user. The recommender system compares the student profile with the quiz questions. The quiz questions are ranked based on the probability that the student answered them correctly. After ranking the exercise questions based on the probability of the student's correct response, the system selects and displays questions with certain probabilities in the Answer page.

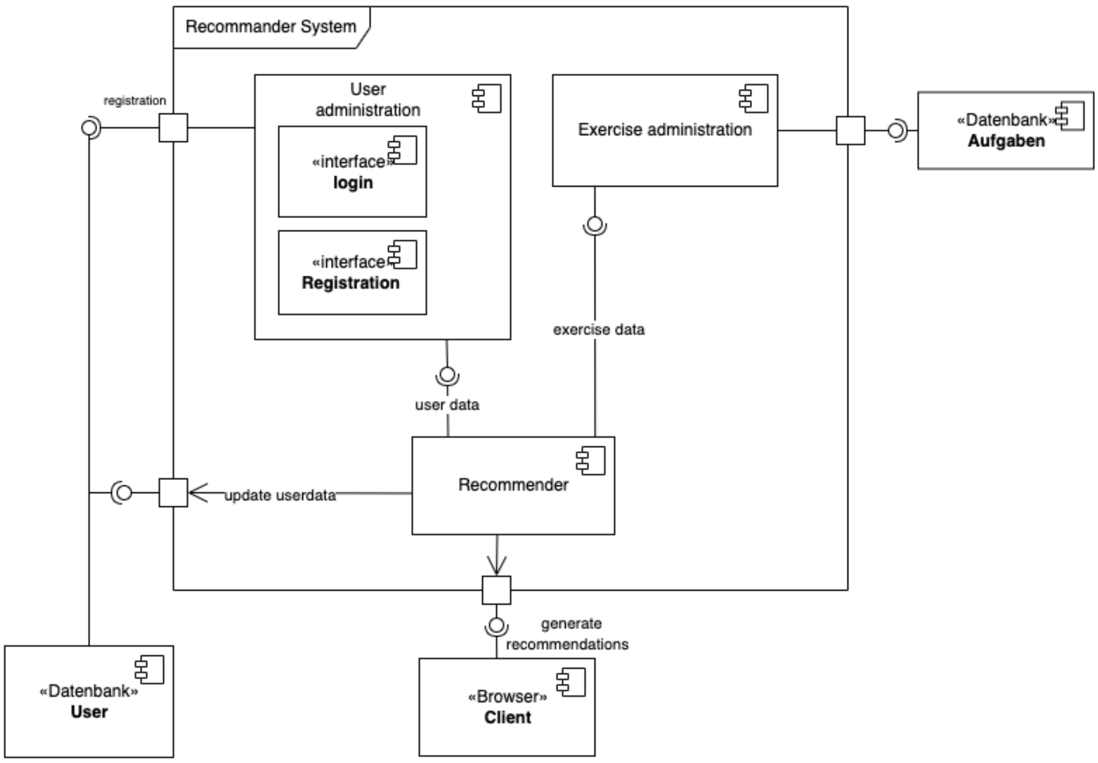



**Figure 4.3:** The RS components

Figure 4.3 is the Component Diagram of IndiLearn. The recommender system (RS) is a key component in providing personalized recommendations to users. It is composed of three subcomponents: user administration, exercise administration, and the recommender itself. The user administration and exercise administration subcomponents act as interfaces to the recommender. They retrieve information from the corresponding databases and provide it to the recommender for processing. The user administration subcomponent handles user registration and login services, ensuring that users are authenticated before accessing the RS. The recommender subcomponent generates new recommendations based on the information it receives and displays them in the browser, providing users individualized exercises.

### 4.1.5 Database

Selecting PostgreSQL as our database management system provides scalability, high performance, reliability, and compatibility with other technologies for storing and managing educational task datasets, student profiles, and past performance data.

### 4.1.6 IRT model

In the System Design phase, we have selected the Two-Parameter Logistic Model (2PLM) as the IRT model that is appropriate for educational task datasets. The 2PLM allows for the estimation of two item parameters: the difficulty parameter, which indicates the level of skill required to answer an item correctly, and the discrimination parameter, which measures the item's ability to distinguish between high and low ability students.

### 4.1.7 Front-end and Back-end frameworks

For our system design, we have selected React as our front-end Framework due to its suitability for creating responsive and interactive web interfaces. React's component-based architecture allows for modularity, making it easier to maintain and scale our application. Additionally, React's virtual DOM enables efficient rendering and updates of components, improving overall performance.

For our back-end framework, we have chosen Flask due to its ability to handle large amounts of data and provide robust API endpoints for data processing. Flask is a lightweight and flexible framework that allows for easy integration with various databases, making it suitable for our needs. Its microservices approach also allows us to add or remove components as required, making it scalable for future growth [11].

## 4.2 Front-end and UI

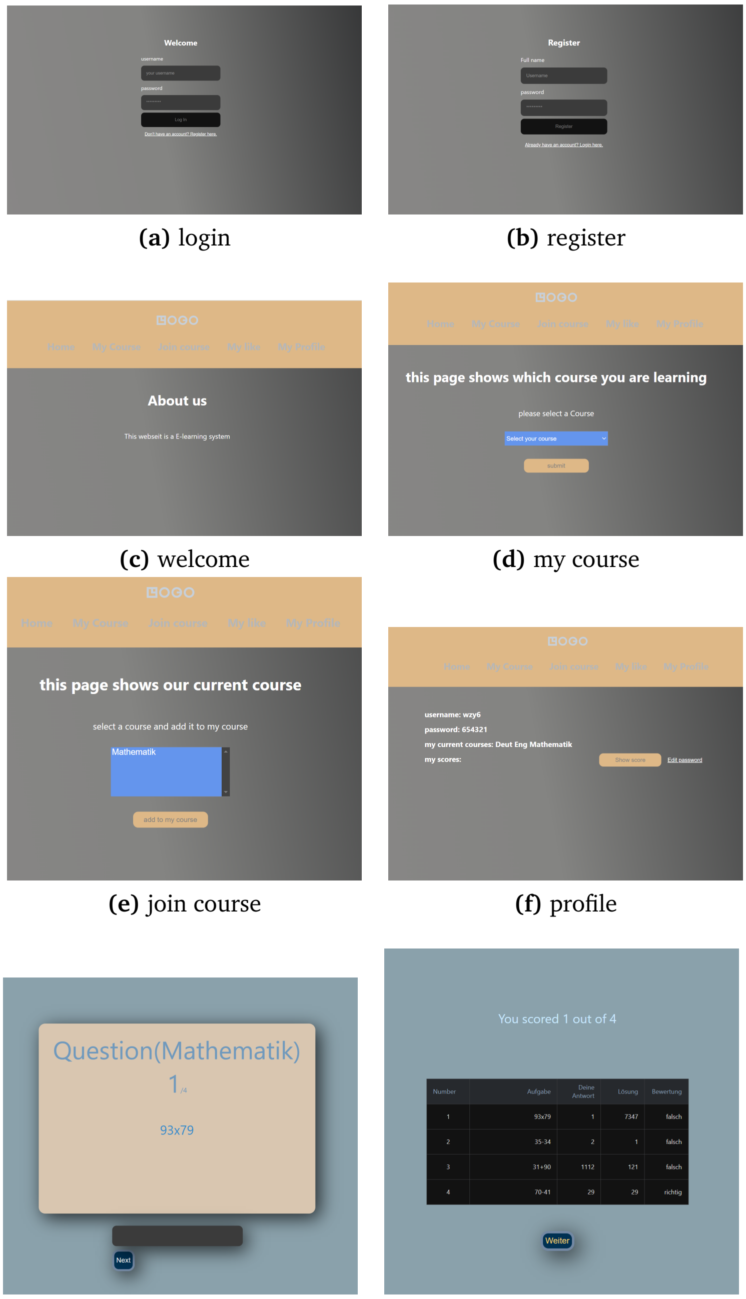
In Figure 4.4a there are 2 input bars, one for the user to input username and the other for the user to input the password. If the user data does not exist in the database, the user should click the "register" button to create an account in the database. If the user data exists, the user clicks the "Login" button and the system will check if the information is correct. If yes, the user will get into the homepage, if not, an error message will be shown. Fetch method is applied in this page to connect the POST method for users in back-end.

In Figure 4.4b, user input their username and password to create a user account in the database. Fetch method is applied in this page to connect the POST method for users in back-end.

There is a menu above Figure 4.4c. User can switch between the following sub-pages with it. There are some labels in the welcome page to show information about the system.

Figure 4.4d provides an option bar, so that the user can select which category of the question that he wants to answer. User click "Confirm" button to forward to the answer page. Fetch method is applied here to connect the GET method for categories.

In Figure 4.4e, the system will read all the categories that exist in the database and then provide them to the user in this page. The user can select the category in the option bar, and then click "add" button to save the category in "my course" page. Or he can also click "remove" button to remove the selected and saved category from "my course". Fetch method is applied in this page to connect the GET method for categories in back-end.




**(a)** login **(b)** register **(c)** welcome **(d)** my course **(e)** join course **(f)** profile **(g)** answer **(h)** score

**Figure 4.4:** front-end UI

In Figure 4.4f, user can read his username directly and his score after clicking "show scores" button in this page. He can click "edit" to change his information. Fetch methods are applied in this page to connect the GET method for scores and users, and to connect the PUT method for users in back-end.

Figure 4.4g includes a title, which mentions the category of the questions and the question number. Then the text of the question is shown in the middle of the page. There is an input, where students answer the question. At the bottom are the next or back buttons, which are used to switch between questions. If all questions of a question set are answered and the student click the button to hand on the answer.

In Figure 4.4h, the result of the answers would be given in a table form. The student can compare their answer with the solution. The GET method of axios is applied here to access the GET method for questions in back-end. The fetch method is applied here to connect the POST method for scores and user abilities.

## 4.3 Back-end

Flask is used as IndiLearn's backend server, it connects to a PostgreSQL database to perform CRUD operations (Create, Read, Update, Delete) on various models. The web application uses several Flask extensions to handle different functionalities, such as Flask-RESTful, Flask-CORS, Flask-SQLAlchemy, Flask-Migrate, and Flask-Marshmallow.

The models used in the IndiLearn include Users, Subjects, Exercises, Leistung, and Level. The Users model represents a user of the application and contains fields for their username and password. The Subjects model represents the subject a user selects and contains fields for the subject name and the user's username. The Exercises model contains fields for the exercise problem, solution, category, difficulty level, and discrimination value. The Leistung model contains fields for the user's performance on a particular exercise, including their score, the exercise question, category, and difficulty level. Finally, the Level model contains the user's skill level for a particular category. Models are used to define the structure of the data stored in the PostgreSQL database. Each model represents a table in the database and defines the fields or columns that are stored in that table.

The following table shows the names of all the databases and their attributes.

| Table name | Attributes |
|---|---|
| aufgaben | id, aufgabenstellung, musterloesung, kategorie, schwerigkeit, discrimination |
| leistung | id, username, aufgabestellung, score, kategorie, schwerigkeit, zeitpunkt |
| level | id, username, faehigkeit, kategorie, create_time |
| mysubject | id, fachname, username |
| studentUsers | id, username, password |

**Table 4.2:** Database Tables

In our web application, we have implemented several Flask routes that serve as HTTP endpoints for various functionalities. These endpoints include routes for adding, retrieving, and updating users, getting exercises based on the user's level, and retrieving categories for exercises. The following is a list of the endpoints and their HTTP methods:

**GET:**

- /get: This endpoint retrieves a list of all users registered in the application.
- /get/<username>: This endpoint retrieves a specific user by their username.
- /getaufgabe/<username>/<kategorie>: This endpoint checks if the given user is new and if so, it forces them to take a placement test. It then returns a set of exercises based on the user's level and the requested category.
- /getleistung/<username>: This endpoint retrieves the performance records for a given user.
- /getkategories: This endpoint retrieves a list of all subject names available in the application.
- /getkategories/<username>: This endpoint retrieves a list of all subjects selected by the given user.

**POST:**

- /api/login: This endpoint is used to authenticate a user with their username and password.
- /api/register: This endpoint creates a new user account in the application's database.
- /addsubject: This endpoint adds a new subject to a user's list of selected subjects.
- /addleistung: This endpoint adds a new performance record to the application's database. It records whether a user answered a question correctly or not.
- /addlevel: This endpoint adds an ability level to a user.

**PUT:**

- /update/<username>: This endpoint updates a specific user's password by their username.

These endpoints provide the necessary functionality for the IndiLearn's core features and can be accessed using the appropriate HTTP methods.

The web application uses Flask-SQLAlchemy to interact with the PostgreSQL database and Flask-Migrate to handle database migrations. Flask-Marshmallow is used to serialize and deserialize data between Python objects and JSON. Finally, the application uses JWT for user authentication and authorization.

## 4.4 Algorithm

IndiLearn's recommendation system relies on IRT to provide personalized learning recommendations. To support this system, the difficulty and discrimination parameters of 50 math problems from the exercise book DEMAT 9 for 9th graders [22] were estimated using a dataset that included responses (correct or incorrect) from 1378 students who completed these problems.

By analyzing the dataset, the difficulty and discrimination parameters were derived based on the responses of the students. This estimation process allowed for a comprehensive understanding of each math problem's level of difficulty and its ability to differentiate between students with varying proficiency levels.

To estimate these parameters, advanced statistical techniques, such as the logistic function as the item response function in the IRT likelihood function, were employed. The maximum likelihood estimates of the parameters were obtained using the minimize function from the scipy.optimize module. This rigorous analysis provided accurate estimations of the difficulty and discrimination parameters for each math problem.

The resulting estimated difficulty and discrimination parameters were then incorporated into the database utilized by IndiLearn's recommendation system. This integration enables the system to generate personalized and effective learning recommendations based on each student's ability level and performance on specific math problems. By considering the estimated parameters, the system can recommend math problems that are appropriately challenging for each learner, fostering their growth and enhancing their overall learning experience.

The fundamental concept behind IndiLearn: is to utilize IRT to forecast a user's accuracy on an exercise with a given difficulty level. Initially, new users will be presented with 5 questions of random difficulty. Using the difficulty levels of the questions and the assigned weights, the system will estimate an initial ability value for the user. The exercises of different difficulty levels and their weights are shown in the following table:

| difficulty level | <-1 | (-1,0) | (0,1) | (1,2) | (2,3) | >3 |
|---|---|---|---|---|---|---|
| weight | 0.45 | 0.25 | 0.15 | 0.1 | 0.05 | 0.02 |

**Table 4.3:** difficulty levels and their weights

This initial ability value serves as the baseline for subsequent recommendations. Following this, the system will use IRT to push four questions at a time of suitable difficulty for the user. As the user interacts with the system, their ability value is updated based on their performance, and the recommendation engine becomes more accurate in predicting their proficiency level. Ultimately, the user's ability value will converge to their true level as they complete more exercises and receive more accurate recommendations. The IndiLearn system utilizes a two-parameter model. The parameters for each question in the IndiLearn system are generated through maximum likelihood estimation (MLE) training based on a training set. This training process involves using the responses of numerous users to each question to estimate the probability that a user with a certain ability level will answer the question correctly. The difficulty parameter is related to the probability of correctly answering the question for users with low ability levels, while the discrimination parameter is related to the extent to which the question can distinguish between users with different ability levels.

The ICC curve of the correctly answered question and the reverse ICC curve of the incorrectly answered question are combined to form a new ICC curve, which is used to estimate the user's ability level. The reverse ICC curve is obtained by subtracting the original ICC curve from a straight line of unit slope and intercept 1. The resulting curve is then multiplied by the ICC curve of the correctly answered question to obtain the new ICC curve. The horizontal coordinate of the vertex of the new ICC curve is taken as the latest ability value of the user.

To ensure that the system can handle situations where a user gets all answers wrong or all answers right, as well as to maintain a positive feedback loop for the user, we have implemented limits on how much their ability value can increase or decrease. Specifically, the maximum increase is capped at 0.5, and the maximum decrease is capped at 0.3. This helps to prevent the ability value from fluctuating too drastically and provides a more stable estimation of the user's true proficiency level over time.

# 5 Evaluation

A focus group consisting of four informatics students majoring in college was conducted to evaluate the personalized recommendation system of the IndiLearn e-learning website [20]. The evaluation aimed to gather feedback on the system's effectiveness and to identify any issues or suggestions for improvement.

## 5.1 Methodology

The focus group consisted of six participants, comprising two hosts and four testers. All testers were college students majoring in informatics, aged between 20 and 24 years. To collect feedback from the participants, the researchers employed the System Usability Scale (SUS) questionnaire, which is widely recognized as a standardized tool for assessing perceived usability [9]. The questionnaire was adapted to align with the specific context of the study, incorporating five additional questions. The participants were asked a series of questions related to their experience of using the system, including its complexity, ease of use, and whether it helped improve their learning. For comprehensive details regarding the agenda and questionnaire, please consult the accompanying attachment.

## 5.2 Focus Group Result

The responses to the System Usability Scale questionnaire are listed in the table.

Based on the average results of each question, here are some noteworthy points and why they are important: Testers generally found the system appealing, with an average score of 4.5 out of 7 on the question "I thought I would like to use this system frequently." This indicates that the system has the potential for adoption and continued use.

According to user feedback, the system is generally considered user-friendly. Users rated the complexity of the system with an average score of 3 out of 7 for the question "I find the system unnecessarily complex." Similarly, for the question "I thought there were too many design inconsistencies in this system," users gave an average score of 2.25 out of 7, indicating that they did not perceive the system as overly complex or having significant design flaws. The average response to the question "I needed technical support for using the system" was 3, suggesting that users did not require substantial technical assistance. Overall, the average score of 5.5 out of 7 for the question "I find the system easy to use" confirms the positive user experience and satisfaction.

| | Tester1 | Tester2 | Tester3 | Tester4 | **average** |
|---|---|---|---|---|---|
| 1 | 7 | 7 | 4 | 2 | **4.5** |
| 2 | 3 | 1 | 3 | 5 | **3** |
| 3 | 7 | 7 | 5 | 3 | **5.5** |
| 4 | 7 | 7 | 6 | 7 | **6.75** |
| 5 | 1 | 1 | 4 | 3 | **2.25** |
| 6 | 6 | 7 | 6 | 5 | **6** |
| 7 | 1 | 1 | 6 | 4 | **3** |
| 8 | 7 | 3 | 5 | 1 | **4** |
| 9 | 5 | 7 | 5 | 2 | **4.75** |
| 10 | 2 | 6 | 4 | 5 | **4.25** |
| 11 | 5 | 5 | 4 | 4 | **4.5** |
| 12 | 5 | 5 | 5 | 1 | **4** |
| 13 | 4 | 1 | 3 | 5 | **3.25** |
| 14 | 5 | 3 | 5 | 5 | **4.5** |
| 15 | 3 | 5 | 2 | 5 | **3.75** |

**Table 5.1:** Questionnaire results

It is important to note that the question "I thought that the questions were too easy for me," which received an average response of 5.5, suggests that the testers generally found the questions to be too easy. However, it is crucial to consider that the testers in this study were exclusively college students who may not have been an ideal match for the test questions. Consequently, this result may not hold significant reference value. Furthermore, through interviews with the testers, we discovered that this factor also influenced the average score of 4 for question 8, "I could realize the difference in complexity between questions," to some extent.

And for the additional questions: The majority of the participants liked the style of the webpage (Question 11), as evidenced by the high average answer of 4 out of 5. This is a positive indication that the design of the website is appealing to the users.

The questionnaire results also indicated that most participants preferred to use the system on their cell phones rather than on their computers (Question 14). This may be due to the convenience and portability of mobile devices, which would allow users to access the system from anywhere at any time.

On questions related to the need for interaction with the teacher, the average answer of 3.25 suggests that the participants did not feel a strong need for a board to interact with their teacher within the system. However, it's worth noting that this may depend on the specific use case and context of the system, and further investigation may be needed to fully understand this response.

In addition, participants responded positively to the idea of adding a time limit to the questions, with an average answer of 4. This suggests that users felt that having a time limit would enhance their learning experience and encourage them to answer the questions promptly.

The testers also provided actionable suggestions on how to improve our learning system. Based on the feedback from the testers, the quiz page needs to provide clear guidance on answer formats, including whether to use decimals or fractions, and whether to use commas or periods. Additionally, the system should prompt users for mistakes in punctuation when entering answers and specify the requirements for the answer format.

In terms of UI design, the testers suggested that for an elementary school audience, the quiz page should use bright colors and incorporate cartoon elements to increase user engagement and make it more enjoyable to use. Overall, incorporating these suggestions can lead to an improved user experience and greater success for the IndiLearn system.

Overall, the results of the focus group test suggest that the IndiLearn has both strengths and weaknesses, and further improvements are needed to ensure a positive user experience.

# 6 Conclusion

In this study, we aimed to build a web app that implements basic login and registration functions, allows users to select and answer tasks, and uses IRT to infer individual ability and recommend tasks based on difficulty. We achieved these goals and implemented the MLE method to infer the two parameters of the IRT of the tasks, i.e., difficulty and discrimination. The two-parameter IRT model allowed us to accurately estimate both the difficulty of the tasks and the ability of the users, which enabled us to recommend tasks that matched the difficulty level of the user's ability. In addition, we invited several testers to evaluate the performance of the IRT algorithm used in our web app. During the test, participants were presented with five randomly selected tasks and given a temporary ability value based on their responses. Subsequently, the system updated the user's ability value based on their performance in each round of four tasks, and continued to recommend tasks corresponding to their updated ability level.

| task | discrimination(a) | difficulty(b) |
|---|---|---|
| 1 | 0.861 | -1.623 |
| 2 | 1.173 | -0.304 |
| 3 | 1.410 | -1.004 |
| 4 | -0.151 | 3.426 |
| 5 | -0.268 | 4.968 |
| 6 | -0.193 | 4.169 |
| 7 | 1.492 | -0.919 |
| 8 | 1.818 | -0.133 |
| 9 | 1.877 | -1.328 |
| 10 | -0.286 | 4.883 |
| 11 | -0.216 | 4.260 |
| 12 | -0.140 | 3.239 |
| 13 | 2.115 | -1.262 |
| 14 | -0.118 | 2.033 |
| 15 | 1.958 | 0.165 |

**Table 6.1:** difficulty and discrimination of the first 15 tasks estimated using MLE

The results of our test demonstrated that the IRT algorithm used in our web app accurately inferred the user's ability level and recommended tasks that matched their difficulty. Specifically, our system achieved a high level of accuracy in inferring user ability, as evidenced by the close alignment between the difficulty level of the recommended tasks and the ability level of the users as estimated by the IRT model. For instance, when a tester was estimated to have an ability value of 1.5 (on a scale of -2 to 5), the system typically recommended tasks with difficulty levels ranging from 1.2 to 1.8, and the probability of the tester correctly solving these tasks was estimated to be approximately 50 percent based on the task discrimination parameter of the IRT model. However, we acknowledge that the sample of tasks was relatively small at the focus group phase, which limited our ability to evaluate the accuracy of IRT's estimation of user ability values at the application level. This is a common limitation of IRT-based recommendation systems, which rely on user feedback and require sufficient item response data to accurately estimate user ability values. Despite this limitation, our study provides a promising proof-of-concept for the use of IRT-based approaches in recommendation systems, and highlights the potential of these methods to personalize and optimize learning experiences for users.

## 6.1 Future Work

Looking ahead, there are many functions we can add to this web app to further improve its functionality, security, interactivity, and user-friendliness. For example, we can implement a countdown function, prompt function, and comment function to enhance the task-answering experience. We can also add Two-Factor Authentication and allow users to choose their results' visibility to increase the app's security. Furthermore, including a discussion board and a reflection board would increase the app's interactivity and provide users with opportunities to discuss and reflect on their learning experiences. Lastly, we can improve the app's layout to make the interface more visually appealing and user-friendly.

In addition, future research could explore ways to overcome the limitation of small sample sizes and insufficient item response data, such as using larger and more diverse samples of tasks and users, or leveraging other sources of data to supplement item response data.

# 7 Attachment

In this appendix, we present additional information related to the evaluation of the personalized recommendation system on the IndiLearn e-learning website. This includes the detailed results of the System Usability Scale questionnaire for the focus group participants.

## 7.1 Focus Group Agenda

The following agenda was followed for the focus group test:

1. **Introduction (5 minutes)**
    a) Introduction of participants and purpose of the event
2. **Brief Introduction to the E-Learning Website (5 minutes)**
    a) Presentation of the main features and characteristics of the website
    b) Explanation of the test and questionnaire
3. **Conducting the Test (15 minutes)**
    a) Each tester is provided with a laptop and asked to use the website
    b) Observations are made of testers as they use the website
    c) Notes are taken of observations and feedback during the test
4. **Completion of the Questionnaire (10 minutes)**
    a) Distribution of the System Usability Scale questionnaire
    b) Each tester fills out the questionnaire
    c) Collection of completed questionnaires
5. **Group Discussion (15 minutes)**
    a) Each tester shares their opinions and experiences with the website
    b) Discussion of positive and negative aspects of the website
    c) Suggestions for improvement of the website
6. **Conclusion (5 minutes)**
    a) Summary of the results

b) Thank you to the participants

## 7.2 Focus Group Questionnaire and Result

The following questionnaire was used to collect feedback from the participants:

1. I thought I would like to use this system frequently. (1-7 scale)
2. I found the system unnecessarily complex. (1-7 scale)
3. I thought the system was easy to use. (1-7 scale)
4. I thought that the questions were too easy for me. (1-7 scale)
5. I thought there were too many design inconsistencies in this system. (1-7 scale)
6. I thought the system is helpful for self-improvement. (1-7 scale)
7. I needed technical support for using the system. (1-7 scale)
8. I could realize the difference in complexity between questions. (1-7 scale)
9. I thought the quantity of questions in one set was sensible. (1-7 scale)
10. I need more detail about my scores. (1-7 scale)
11. How much do you like the style of this webpage? (1-5 scale)
12. Do you think we should add a time limit to questions? (1-5 scale)
13. As a student, do you need a board in this system to interact with your teacher? (1-5 scale)
14. Would you prefer to use this system on your phone or computer? (1-5 scale)
15. Are you willing to allow your records to be accessed by your teachers? (1-5 scale)

Table 1

| | HYC | HYC | TYC | ZYS | |
|---|---|---|---|---|---|
| **1** | 5 | 7 | 4 | 2 | |
| **2** | 3 | 1 | 3 | 5 | |
| **3** | 7 | 7 | 5 | 3 | |
| **4** | 7 | 7 | 6 | 7 | |
| **5** | 1 | 1 | 4 | 3 | |
| **6** | 6 | 7 | 6 | 5 | |
| **7** | 1 | 1 | 6 | 4 | |
| **8** | 7 | 3 | 5 | 1 | |
| **9** | 5 | 7 | 5 | 2 | |
| **10** | 2 | 6 | 4 | 5 | |
| **11** | 5 | 5 | 4 | 4 | |
| **12** | 5 | 5 | 5 | 1 | |
| **13** | 4 | 1 | 3 | 5 | |
| **14** | 5 | 3 | 5 | 5 | |
| **15** | 3 | 5 | 2 | 5 | |
| **Additional suggestions. (Chinese and English below):** | 1.注册 注册的时候可以输2遍密码进行确认，避免打字错误，或者密码栏应可选隐藏或可见密码 1. Registration: When registering, it should be possible to enter the password twice for confirmation to avoid typing errors. Alternatively, the password field should have the option to hide or show the password. | 1. 要提示符号输入是否正确<br>1. There should be a prompt to indicate whether the mathematical symbol input is correct or not. | 输入答案时标点符号的错误需要提示，对答案格式的要求需要说明。zb. 1/5 和0.2; 0.2 和 0,2.<br>Mistakes in punctuation when entering the answer should be prompted, and the requirements for the answer format should be specified. For example, whether to use a decimal point or a decimal comma, and whether to input fractions or decimals. For instance, the system should accept both "1/5" and "0.2" as correct answers, as well as both "0.2" and "0,2". | 1. 登录系统不会自动跳转，需要手动点击， 有些麻烦。 The system does not automatically redirect upon login and requires manual clicking, which can be somewhat inconvenient. | |
| | 2. Profil页面 我的密码可以选择是否可见2. Profile Page: I would like to have the option to choose whether my password is visible or hidden. | 2. 标明用小数还是分数输入<br>2. It should be indicated whether to input the answer as a decimal or a fraction. | 选择course的框颜色有点格格不入.<br>The color of the course selection box looks a bit out of place. | 2. 题目数量太少，经常随机到相同题目。There are too few questions, and I often encounter the same questions randomly. | |
| | 3. 提示框 位置过于靠上<br>3. Information prompt box: The position of the prompt box is too high up. | 3. 碰到一样的题.<br>3. sometimes you may be pushed to do a question that you have done before. | | 3. 可以增加与老师互动版块，可以对错题进行提问。 It would be great to add an interactive section with the teacher where we can ask questions about the incorrect answers. | |
| | 4. 做题 应在做题页面提示答题格式，如在小数中应该使用逗号还是句点 4. Quiz: The quiz page should provide guidance on the answer format, such as whether to use commas or periods in decimal numbers. | | | | |

1

**Figure 7.1:** results overview

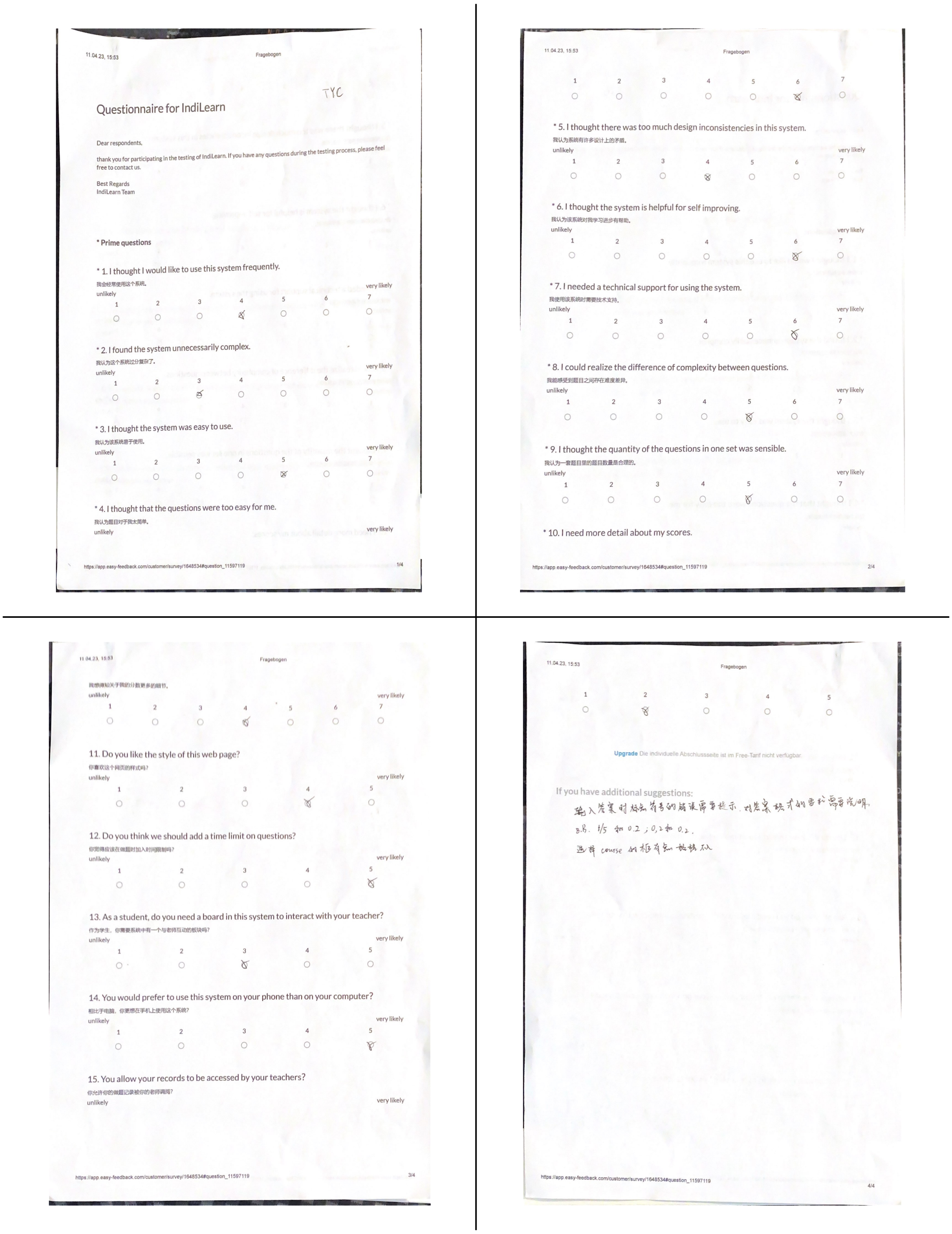
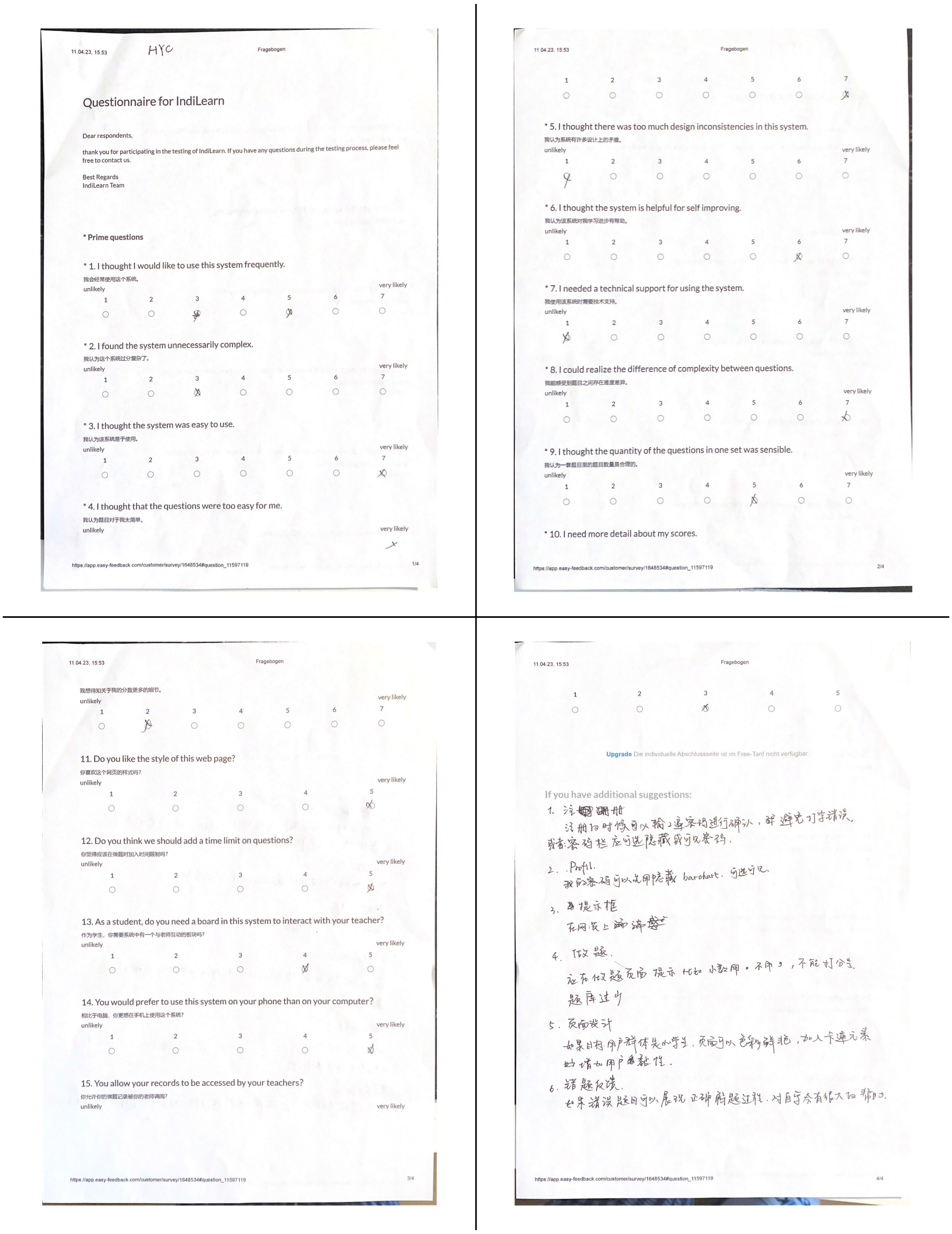


**Figure 7.2:** Tester1 results


11.04.23, 15:53 HYC Fragebogen

Questionnaire for IndiLearn

Dear respondents,

thank you for participating in the testing of IndiLearn. If you have any questions during the testing process, please feel free to contact us.

Best Regards
IndiLearn Team

* Prime questions

* 1. I thought I would like to use this system frequently.
我会经常使用这个系统。
unlikely 1 2 3 4 5 6 7 very likely

* 2. I found the system unnecessarily complex.
我认为这个系统过分复杂了。
unlikely 1 2 3 4 5 6 7 very likely

* 3. I thought the system was easy to use.
我认为该系统易于使用。
unlikely 1 2 3 4 5 6 7 very likely

* 4. I thought that the questions were too easy for me.
我认为题目对于我太简单。
unlikely very likely

https://app.easy-feedback.com/customer/survey/1648534#question_11597119 1/4

11.04.23, 15:53 Fragebogen

1 2 3 4 5 6 7

* 5. I thought there was too much design inconsistencies in this system.
我认为系统有许多设计上的矛盾。
unlikely 1 2 3 4 5 6 7 very likely

* 6. I thought the system is helpful for self improving.
我认为该系统对我学习进步有帮助。
unlikely 1 2 3 4 5 6 7 very likely

* 7. I needed a technical support for using the system.
我使用该系统时需要技术支持。
unlikely 1 2 3 4 5 6 7 very likely

* 8. I could realize the difference of complexity between questions.
我能感受到题目之间存在难度差异。
unlikely 1 2 3 4 5 6 7 very likely

* 9. I thought the quantity of the questions in one set was sensible.
我认为一套题目里的题目数量是合理的。
unlikely 1 2 3 4 5 6 7 very likely

* 10. I need more detail about my scores.

https://app.easy-feedback.com/customer/survey/1648534#question_11597119 2/4

11.04.23, 15:53 Fragebogen

我想得知关于我的分数更多的细节。
unlikely 1 2 3 4 5 6 7 very likely

11. Do you like the style of this web page?
你喜欢这个网页的样式吗?
unlikely 1 2 3 4 5 very likely

12. Do you think we should add a time limit on questions?
你觉得应该在做题时加入时间限制吗?
unlikely 1 2 3 4 5 very likely

13. As a student, do you need a board in this system to interact with your teacher?
作为学生，你需要系统中有一个与老师互动的板块吗?
unlikely 1 2 3 4 5 very likely

14. You would prefer to use this system on your phone than on your computer?
相比于电脑，你更想在手机上使用这个系统?
unlikely 1 2 3 4 5 very likely

15. You allow your records to be accessed by your teachers?
你允许你的做题记录被你的老师调阅?
unlikely very likely

https://app.easy-feedback.com/customer/survey/1648534#question_11597119 3/4

11.04.23, 15:53 Fragebogen

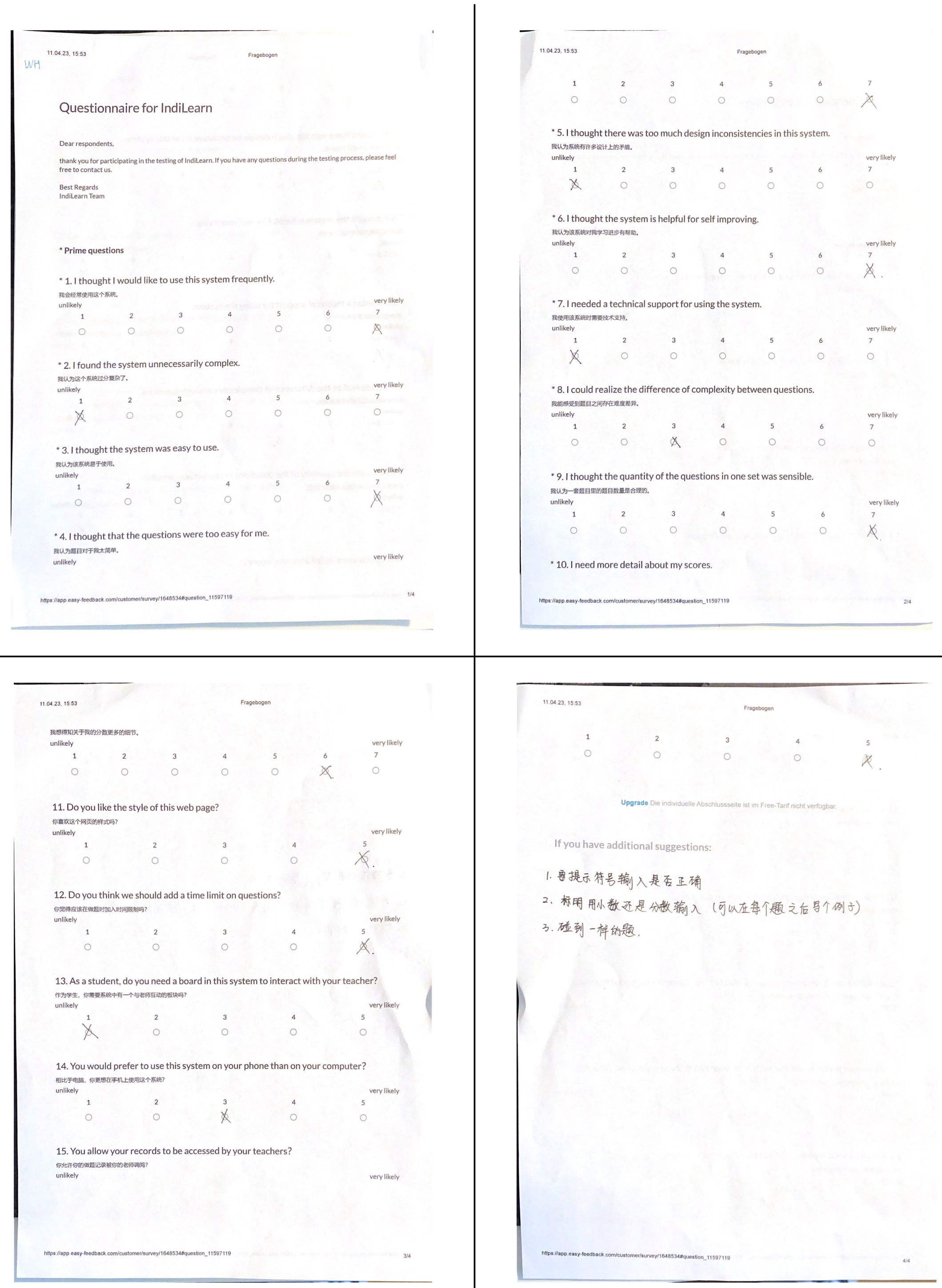
1 2 3 4 5

Upgrade Die individuelle Abschlussseite ist im Free-Tarif nicht verfügbar.

If you have additional suggestions:

https://app.easy-feedback.com/customer/survey/1648534#question_11597119 4/4

**Figure 7.3:** Tester2 results


11.04.23, 15:53 Fragebogen

WH

Questionnaire for IndiLearn

Dear respondents,

thank you for participating in the testing of IndiLearn. If you have any questions during the testing process, please feel free to contact us.

Best Regards
IndiLearn Team

* Prime questions

* 1. I thought I would like to use this system frequently.
我会经常使用这个系统。
unlikely 1 2 3 4 5 6 7 very likely

* 2. I found the system unnecessarily complex.
我认为这个系统过分复杂了。
unlikely 1 2 3 4 5 6 7 very likely

* 3. I thought the system was easy to use.
我认为该系统易于使用。
unlikely 1 2 3 4 5 6 7 very likely

* 4. I thought that the questions were too easy for me.
我认为题目对于我太简单。
unlikely very likely

https://app.easy-feedback.com/customer/survey/1648534#question_11597119 1/4

11.04.23, 15:53 Fragebogen

1 2 3 4 5 6 7

* 5. I thought there was too much design inconsistencies in this system.
我认为系统有许多设计上的矛盾。
unlikely 1 2 3 4 5 6 7 very likely

* 6. I thought the system is helpful for self improving.
我认为该系统对我学习进步有帮助。
unlikely 1 2 3 4 5 6 7 very likely

* 7. I needed a technical support for using the system.
我使用该系统时需要技术支持。
unlikely 1 2 3 4 5 6 7 very likely

* 8. I could realize the difference of complexity between questions.
我能感受到题目之间存在难度差异。
unlikely 1 2 3 4 5 6 7 very likely

* 9. I thought the quantity of the questions in one set was sensible.
我认为一套题目里的题目数量是合理的。
unlikely 1 2 3 4 5 6 7 very likely

* 10. I need more detail about my scores.

https://app.easy-feedback.com/customer/survey/1648534#question_11597119 2/4

11.04.23, 15:53 Fragebogen

我想得知关于我的分数更多的细节。
unlikely 1 2 3 4 5 6 7 very likely

11. Do you like the style of this web page?
你喜欢这个网页的样式吗?
unlikely 1 2 3 4 5 very likely

12. Do you think we should add a time limit on questions?
你觉得应该在做题时加入时间限制吗?
unlikely 1 2 3 4 5 very likely

13. As a student, do you need a board in this system to interact with your teacher?
作为学生，你需要系统中有一个与老师互动的板块吗?
unlikely 1 2 3 4 5 very likely

14. You would prefer to use this system on your phone than on your computer?
相比于电脑，你更想在手机上使用这个系统吗?
unlikely 1 2 3 4 5 very likely

15. You allow your records to be accessed by your teachers?
你允许你的做题记录被你的老师调阅吗?
unlikely very likely

https://app.easy-feedback.com/customer/survey/1648534#question_11597119 3/4

11.04.23, 15:53 Fragebogen

1 2 3 4 5

Upgrade Die individuelle Abschlussseite ist im Free-Tarif nicht verfügbar.

If you have additional suggestions:

1. 要提示符号输入是否正确
2、标明用小数还是分数输入（可以在每个题之后写个例子）
3. 碰到一样的题，

https://app.easy-feedback.com/customer/survey/1648534#question_11597119 4/4

**Figure 7.4:** Tester3 results

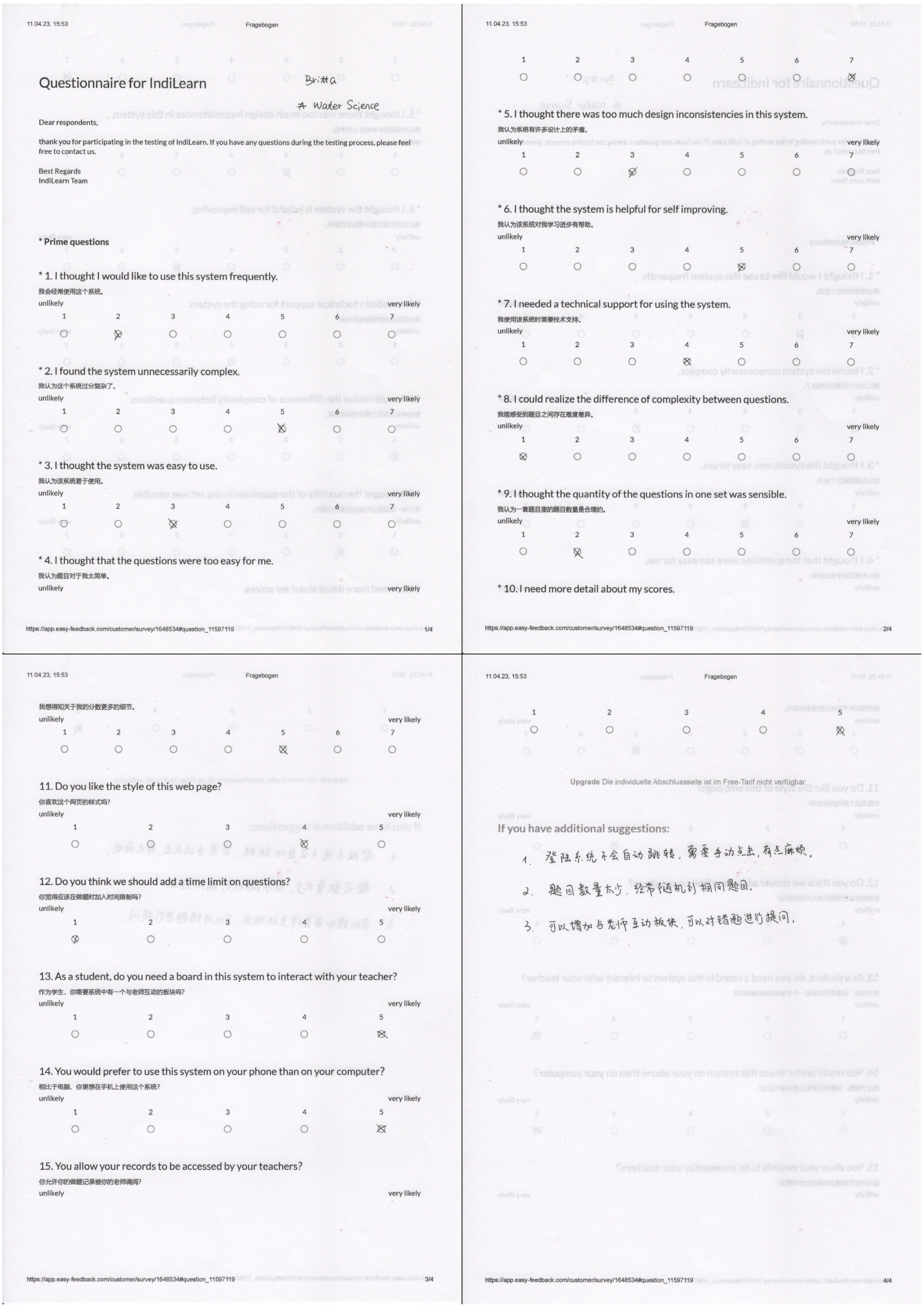


**Figure 7.5:** Tester4 results

# Bibliography


[1] Xinming An and Yiu-Fai Yung. Item response theory: What it is and how you can use the irt procedure to apply it. *SAS Institute Inc. SAS364-2014*, 10(4), 2014.

[2] Robin Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12:331–370, 2002.

[3] Debashis Das, Laxman Sahoo, and Sujoy Datta. A survey on recommendation system. *International Journal of Computer Applications*, 160(7), 2017.

[4] JOANNA S Gorin, SUSAN E Embretson, and D McKay. Item response theory and rasch models. *Handbook of research methods in abnormal and clinical psychology*, pages 271–292, 2008.

[5] Deborah Harris. Comparison of 1-, 2-, and 3-parameter irt models. *Educational Measurement: Issues and Practice*, 8(1):35–41, 1989.

[6] Cheng-Lung Huang and Cheng-Wei Lin. Collaborative and content-based recommender system for social bookmarking website. *International Journal of Computer and Information Engineering*, 4(8):1310–1315, 2010.

[7] Folasade Olubusola Isinkaye, Yetunde O Folajimi, and Bolande Adefowoke Ojokoh. Recommendation systems: Principles, methods and evaluation. *Egyptian informatics journal*, 16(3):261–273, 2015.

[8] Walter L. Leite, Samrat Roy, Nilanjana Chakraborty, George Michailidis, A Corinne Huggins-Manley, Sidney D'Mello, Mohamad Kazem Shirani Faradonbeh, Emily Jensen, Huan Kuang, and Zeyuan Jing. A novel video recommendation system for algebra: An effectiveness evaluation study. In *LAK22: 12th International Learning Analytics and Knowledge Conference*, pages 294–303, 2022.

[9] James R Lewis. The system usability scale: past, present, and future. *International Journal of Human–Computer Interaction*, 34(7):577–590, 2018.

[10] Wei Li, Igor Santos, Flavia C Delicato, Paulo F Pires, Luci Pirmez, Wei Wei, Houbing Song, Albert Zomaya, and Samee Khan. System modelling and performance evaluation of a three-tier cloud of things. *Future Generation Computer Systems*, 70:104–125, 2017.

[11] PS Lokhande, Fankar Aslam, Nabeel Hawa, Jumal Munir, and Murade Gulamgaus. Efficient way of web development using python and flask. 2015.

[12] Frederic M Lord. Practical applications of item characteristic curve theory. *Journal of Educational Measurement*, pages 117–138, 1977.

[13] Frederic M Lord and Melvin R Novick. *Statistical theories of mental test scores*. IAP, 2008.

[14] Jie Lu, Dianshuang Wu, Mingsong Mao, Wei Wang, and Guangquan Zhang. Recommender system application developments: a survey. *Decision Support Systems*, 74:12–32, 2015.

[15] Marwa Hussien Mohamed, Mohamed Helmy Khafagy, and Mohamed Hasan Ibrahim. Recommender systems challenges and solutions survey. In *2019 international conference on innovative trends in computer engineering (ITCE)*, pages 149–155. IEEE, 2019.

[16] In Jae Myung. Tutorial on maximum likelihood estimation. *Journal of mathematical Psychology*, 47(1):90–100, 2003.

[17] Tam H Nguyen, Hae-Ra Han, Miyong T Kim, and Kitty S Chan. An introduction to item response theory for patient-reported outcome measurement. *The Patient-Patient-Centered Outcomes Research*, 7:23–35, 2014.

[18] Jennifer Petrillo, Stefan J Cano, Lori D McLeod, and Cheryl D Coon. Using classical test theory, item response theory, and rasch measurement theory to evaluate patient-reported outcome measures: a comparison of worked examples. *Value in Health*, 18(1):25–34, 2015.

[19] Konstantinos Pliakos, Seang-Hwane Joo, Jung Yeon Park, Frederik Cornillie, Celine Vens, and Wim Van den Noortgate. Integrating machine learning into item response theory for addressing the cold start problem in adaptive learning systems. *Computers & Education*, 137:91–103, 2019.

[20] Richard A Powell and Helen M Single. Focus groups. *International journal for quality in health care*, 8(5):499–504, 1996.

[21] Rajapandian Radha, K Mahalakshmi, V Sathish Kumar, and AR Saravanakumar. E-learning during lockdown of covid-19 pandemic: A global perspective. *International journal of control and automation*, 13(4):1088–1099, 2020.

[22] Kristin Krajewski Sabrina Schmidt, Marco Ennemoser. *DEMAT 9 Deutscher Mathematiktest für neunte Klassen*. HOGREFE, 2012.

[23] Fumiko Samejima. Graded response model. *Handbook of modern item response theory*, pages 85–100, 1997.

[24] Claude Sammut and Geoffrey I Webb. *Encyclopedia of machine learning*. Springer Science & Business Media, 2011.

[25] Gábor Takács, István Pilászy, Bottyán Németh, and Domonkos Tikk. Scalable collaborative filtering approaches for large recommender systems. *The Journal of Machine Learning Research*, 10:623–656, 2009.

[26] Poonam B Thorat, Rajeshwari M Goudar, and Sunita Barve. Survey on collaborative filtering, content-based filtering and hybrid recommendation system. *International Journal of Computer Applications*, 110(4):31–36, 2015.

[27] Chong Wang and David M Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 448–456, 2011.

[28] Jun Wang, Arjen P De Vries, and Marcel JT Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 501–508, 2006.

[29] Mike Wu, Richard L Davis, Benjamin W Domingue, Chris Piech, and Noah Goodman. Variational item response theory: Fast, accurate, and expressive. *arXiv preprint arXiv:2002.00276*, 2020.

[30] Jiwei Zhang, Jing Lu, Feng Chen, and Jian Tao. Exploring the correlation between multiple latent variables and covariates in hierarchical data based on the multilevel multidimensional irt model. *Frontiers in Psychology*, 10:2387, 2019.

All links were last followed on March 24, 2023.

**Declaration**

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

Essen, 04.06.2023 Jialu, Wu Zhaoyu
Kun Yang

place, date, signature