3. Methodology

The proposed system initiates with a dialogue between the individual and an AI agent to capture the individual's career preferences. Based on these insights, a recommendation system proposes potential careers. Following the initial dialogue and career suggestions, the necessary skills for the chosen careers are extracted and distinct paths for each job are formulated. These paths include required certifications and key activities. The final paths are then presented to the user in a visually engaging format.

As depicted in Fig. 1, the proposed model comprises four distinct sub-systems. This chapter will elaborate on these sub-systems in terms of their architecture, benchmark, input data, implementation steps, assumptions, and limitations.

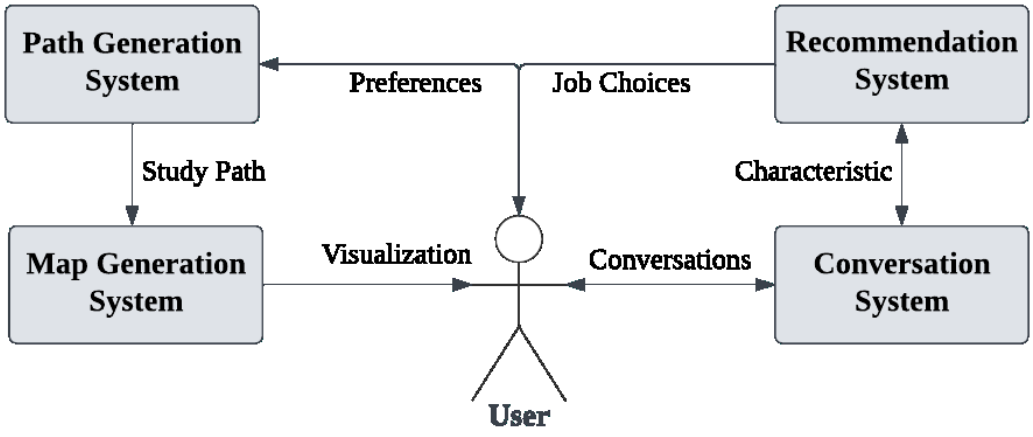

Fig. 1. Overall structure of the Proposed System

3.1. Conversation system

The conversation system serves as the primary data collection module of the model. It facilitates communication with users based on designed question banks, allowing the system to progressively understand and collect user characteristics. The collected and formatted data is then passed on to the subsequent module.

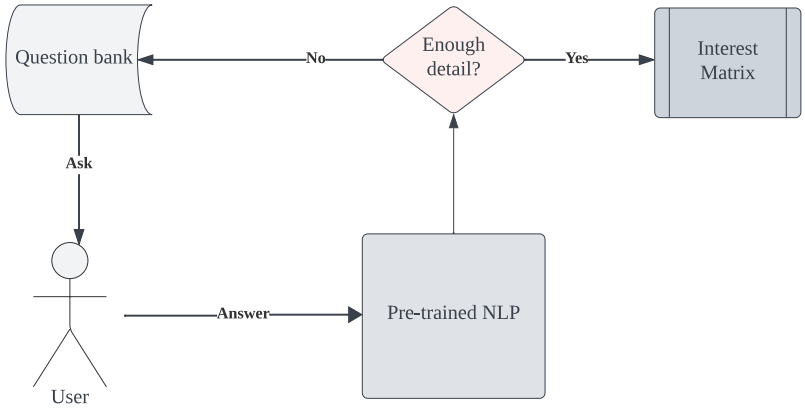

Fig. 2. Architecture of Conversation System

3.1.1. Architecture

The system comprises a question bank and a Natural Language Processing (NLP) model, see figure 2. The question bank contains meticulously crafted questions like MBTI [2]. They are categorized questions based on the nature of relevant jobs and some unbiased questions. A hierarchical structure exists among questions within a category, enabling the system to pose subsequent questions that delve deeper into understanding the user. Unbiased questions help the system define the area of the user's interest; for example, what do you think about yourself?

As for the NLP model, it will be trained before use. It processes the user's responses as input and generates a matrix representing their interests. This provisional output aids the system in continuing with the questions from the bank until the matrix is completed.

3.1.2. Algorithm

The core of the NLP model is an interest matrix:

$$\begin{bmatrix} c_0q_1 & c_0q_2 & \dots & c_0q_n \\ c_1q_1 & c_1q_2 & \dots & c_1q_n \\ \vdots & \vdots & \ddots & \vdots \\ c_mq_1 & c_mq_2 & \dots & c_mq_n \end{bmatrix}$$

Where c_i represents cluster i, and q_j represents question j within cluster i (except for c_0, where c_0 represents those unbiased questions). The c_iq_j is a number between 0 and 1, indicating the user's affinity towards the question. A well-formulated matrix should reflect the user's preferences, with high consistency in certain rows. The matrix is like a composed sentimental analysis.

The algorithm is as follows:

Algorithm 1 NLP for filling interest matrix

$InterestMatrix \leftarrow [0]_{(m+1)\times n}$
$c, q \leftarrow 0, 1$ //$NextQuestion \leftarrow Cluster0\_Question1$
while There is a column of interest matrix is all-zero, i.e., $q < n$ **do**
    $InterestMatrix[:, q+1] \leftarrow NLP(response\,to\,QuestionBank[c, q])$
    $q \leftarrow q + 1$
    $c \leftarrow \underset{c}{\text{argmax}}\, CumulativeInterest(InterestMatrix[:, q])$
end while

The cumulative interest of one row c_i in column $x+1$ is defined as:

$$CI(x+1;c_i) = \sum_{j=1}^{x}(\alpha_{ij} + \beta_{ij}\gamma^j I_{c_i q_j}) \tag{1}$$

where α_{ij} and β_{ij} are constants to be trained, and γ is a discount factor in reducing the influence of further questions, which is common in reinforcement learning [3].

3.1.3. Benchmark

Cronbach's alpha (see equation 2) can be computed for each row c_i (representing the user's responses to a set of questions q_i). A high Cronbach's alpha (close to 1) would indicate that the questions within a cluster are eliciting consistent responses, suggesting that the system understands the user's preferences well [4].

$$\alpha_{c_i} = \frac{m}{m-1}\left(1 - \frac{\sum_{j=1}^{n} \text{Variance}_{q_j}}{\text{Total Variance}}\right) \tag{2}$$

3.1.4. Implementation Steps

3.1.4.a. Build the question bank

1) Collect questions like MBTI [2].
2) Manually categorize the questions and sort them in order according to relevance.

3.1.4.b. Train/Run

1) User Interaction: Begin a conversation with the user, asking questions from the question bank.
2) Response Collection: Collect and store the user's responses.
3) NLP Processing.
4) Propose questions of the most interested area.
5) Iteration: Continue the conversation with the user, updating the matrix as more data is collected.

3.1.5. Assumptions

The primary assumptions for this system are:

- The user responses are truthful and accurately reflect their interests and preferences.
- The NLP model can accurately interpret and quantify the user's responses.
- The questions in the question bank are well-designed and can effectively elicit user preferences related to different job clusters.
- The user will not end the conversation too early.

3.1.6. Limitations

The main limitations of the system are:

- Dependence on User Input: The accuracy of the matrix relies heavily on the user's willingness to provide truthful and comprehensive responses.
- NLP Interpretation: While NLP has made significant strides, it's not perfect. There may be nuances or sentiments in the user's responses that the NLP model fails to capture.
- Question Bank Coverage: The question bank may not cover all possible job clusters or may not include questions that can accurately determine a user's interest in specific areas. It's also possible that the user's interests do not align neatly with the predefined clusters.
- Cronbach's Alpha Limitations: While Cronbach's alpha is a useful measure of internal consistency, it assumes that all variables are equally reliable, which may not always be the case. Also, a high Cronbach's alpha doesn't guarantee that the matrix accurately reflects the user's

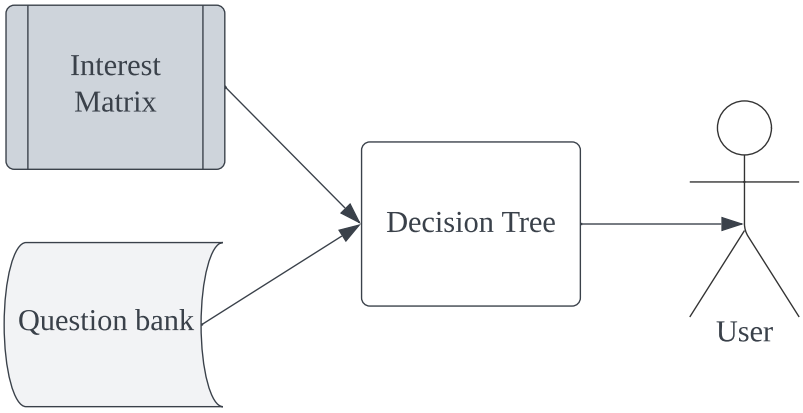

Fig. 3. Architecture of Recommendation System

preferences; it only indicates that the responses to the questions within each cluster are consistent [5].

3.2. Recommendation System

The recommendation system follows the conversation system, using the interest matrix and prior knowledge of the question bank to recommend jobs to users.

3.2.1. Architecture

The system is composed of an input layer and a decision tree. The input layer passes the user's interest on different questions of the bank to the decision tree. The decision tree will generate a list of jobs for the user.

The decision tree model is used because it is easy to understand and interpret. The decision tree mimics the human decision-making process and can handle both categorical and numerical data. The tree is constructed based on the question bank categories and job types, with each question acting as a decision node, leading to different job recommendations based on the user's interests.

3.2.2. Benchmark

This system will be evaluated on the satisfactory rate of users given recommended jobs. In addition to the satisfaction rate of users, the accuracy of the system can also be evaluated by how many users accept the job recommendations. A high acceptance rate would indicate that the system is accurately capturing user preferences and providing relevant job suggestions. User feedback can be collected to further evaluate the performance of the system and refine it.

3.2.3. Implementation Steps

The input is the interest matrix, while the question bank is used to construct the decision tree. After the decision tree, the system will output a list of jobs.

3.2.3.a. Construct the Decision tree

1) Construct the tree based on the categorized question banks manually.
2) Add the job types as the tree leaves.

3.2.3.b. Train/Run

1) Set/Update the threshold of each question's interest.
2) Pass in the interest matrix.

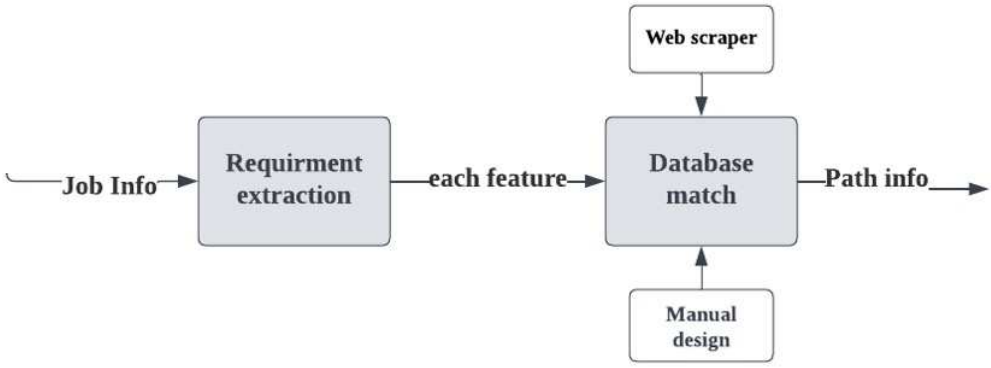

Fig. 4. Architecture of Path Generation System

3.2.3.c. Propose the job list to users

3.2.4. Assumptions

The recommendation system makes a few assumptions:

- The user's responses to questions accurately reflect their interests.
- The interest matrix accurately captures the user's preferences.
- The job types associated with the questions in the question bank are an accurate representation of the job market.

3.2.5. Limitations

The recommendation system also has a few limitations:

- The system assumes that user interests are static and do not change over time. However, in reality, user interests can evolve, and the system may need to adapt accordingly.
- The decision tree model assumes that the questions in the question bank and their associated job types cover all possibilities. However, there may be job types that are not captured in the system, leading to incomplete recommendations.
- Finally, the system's performance largely depends on the quality of the question bank and the accurate categorization of these questions.

3.3. Path Generation System

The Path Generation System aims to provide users with a feasible development plan for their desired job positions. It uses NLP techniques to extract job requirements such as skills and education from the job descriptions. The system then links these requirements to our database to determine the necessary steps for the user to meet them. This could include obtaining certifications, completing relevant coursework, or gaining specific experience. The system's goal is to help users create a practical plan to qualify for their target jobs.

3.3.1. Architecture

The architecture of the Path Generation System is illustrated in the Fig. 3.3 above and consists of several key components:

- Extraction Component: This component uses Natural Language Processing (NLP) techniques to extract job requirements from the job descriptions. Tools like spaCy and MITIE can be used for this purpose [6], [7]
- Database Component: The extracted job requirements are matched with our existing database

of job skills and qualifications.
- Path Information Input Component: This component allows us to manually input path information into our system, such as details about specific courses, certifications, or experiences required for a particular job skill.
- Web Scraping Component: This component uses web scraping techniques to automatically gather information related to the job requirements. Tools like Scrapy or Octoparse can be used to gather information from online resources [8], [9]. Methods such as manual verification will be adopted to select information for entry into the database.

3.3.2. Benchmark

The benchmarking process for the Path Generation System evaluates its accuracy, efficiency, and usability. Accuracy is assessed by comparing the system's output with paths suggested by professionals in the relevant fields. Also, the Skill2Vec dataset which provides a large collection of standardized job descriptions with required skills can be used as a benchmark for evaluating the performance of our system [10]. Efficiency is gauged by the time taken to generate a development path. Usability is determined through user surveys, focusing on aspects such as ease of use and overall user satisfaction.

3.3.3. Implementation Steps

The system takes job descriptions from recruitment websites as input and generates a to-do list for the user with suggested timeline and importance rates.
The implementation process of the Path Generation System involves:

3.3.3.a. Framework build-up

1) Setting up the necessary tools and frameworks, including NLP tools for data extraction and a database for storing job skills and qualifications.

3.3.3.b. Train/Run

1) Feature Extraction: Using NLP to extract job requirements from the job descriptions.
2) Data Matching: Matching the extracted requirements with our database.
3) Path Generation: Creating a development path for each requirement.

3.3.3.c. Update data in database

3.3.4. Assumptions

The Path Generation System operates under several assumptions:

- The job descriptions provided by the user are accurate and comprehensive.
- The system's database contains up-to-date and relevant information about job skills and qualifications.
- The run-time of the system for each job should be controlled within a few seconds to tens of seconds.

3.3.5. Limitations

The Path Generation System has certain limitations:

- Validity of Online Information: The system relies on online information, which may not always be reliable or accurate.
- Ranking Importance: The system does not rank the importance of different job requirements.
- Dynamic Nature of Job Market: The system may not always reflect the most current trends or requirements in the job market.

These will be factors to be considered in future implementation.

3.4. Map Generation System

The Map Generation System, a sub system under path generation system, visualizes the career path suggestions from the Path Generation System in a tree map. It features a main path for general skills and branches for specialized roles, diverging at specific points. The map can be exported in various formats like JPEG, PDF, or HTML for user convenience.

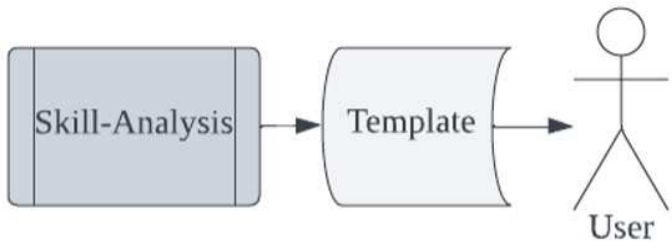

Fig. 5. Architecture of Map Generation System

3.4.1. Architecture

The architecture of the Map Generation System comprises two main parts: Skills Analysis and the Template System.

- Skills Analysis identifies the main skills required for the chosen career path and any specialized skills needed for specific roles within that path.
- The Template System uses this information to generate a visual map, creating a main path for general skills and branches for specialized skills, with divergence points indicating when to acquire each skill.

3.4.2. Benchmark

The benchmark of the Map Generation System is evaluated mainly through user feedback, assessing how effectively the system visualizes their career path.

3.4.3. Implementations Steps

The input data are suggested events generated by the Path Generation System, and the output is a visualized data map of these events.
The implementation involves:

3.4.3.a. Train/Run

1) Input Processing
2) Main Path Generation: The events are sorted using a formula like:

$$\text{Score} = \mathbf{w_1} \times \text{Frequency} + \mathbf{w_2} \times \text{Importance Rate} \tag{3}$$

3.4.3.b. Create a template for visualization

3.4.3.c. Deliver the result to the user

3.4.4. Assumptions

The system assumes that:

- It can accurately distinguish between mainline and branch information.
- It can provide a visual result that satisfies the user's needs.

3.4.5. Limitations

The system does present a couple of limitations:

- The system may not have enough information to accurately define the mainline content, particularly in less common or emerging fields. This is largely due to the need for extensive manually marked importance data.
- For users with a wide range of interests, the single mainline approach may not fully cater to their needs as it could oversimplify their diverse career paths. More sophisticated methods are required to address this issue.