MASTER OF TECHNOLOGY

(MACHINE REASONING

&

REASONING SYSTEM)

PROJECT REPORT

Traveling Recommendation and Schedule System

GROUP 13 TEAM MEMBER

XU KAIXIN
YU XIAOXI
ZHANG ZHILIN

Content

1 EXECUTIVE SUMMARY

Nowadays, with the improvement of living standard, more and more people choose to travel abroad in their holidays. As one of the most popular tourist destinations in Asia, Singapore receives an increasing number of tourists every year, and tourism has become an important part of Singapore's revenue. In Singapore, tourists can not only see the beautiful natural scenery, such as islands, forests and parks, but also see the prosperous city and landscape, feeling the combination of technology and nature.

However, many visitors do not know much about this city. They do not know what kind of attractions they are or which ones represent the characteristics of Singapore. Moreover, for tourists of different ages, they want to visit different types of scenic spots. For example, the elderly prefer to visit natural scenery, while the young prefer to go to amusement parks and other places. For the same tourist, he wants to visit different scenic spots when he is traveling with different people. For example, when people travel with their families, they tend to choose more family-friendly attractions, such as zoos and Chinatown. When they travel with their partners or friends, they tend to choose busy and popular attractions, such as Sands hotel and city eye. Therefore, it is very important to recommend corresponding scenic spots to different tourists according to their preferences.

In addition, when tourists determine the scenic spots they will visit, the timing of the tour becomes an important issue. Most tourists do not know the content of scenic spots and the time needed to visit, so tourism planning is a very complicated and difficult thing for them. This takes into account not only the tour time of the sites, but also the travel time between the sites, as well as the time for people to rest and eat.

In order to solve these problems, our team has put forward a travel recommendation and schedule system. This system can recommend corresponding scenic spots for users according to their preferences, and make a travel schedule for tourists according to the detailed information of scenic spots and the travel time. With this information, tourists can enjoy their holiday time without worry.

2 BUSSINESS VALUE

First of all, although most travel apps on the market have the introduction of scenic spots, they do not have the function of recommendation for customers. Users need to know the landscape， tickets and opening hours of each scenic spot to make their own travel plans. This is inconvenient for the user. And when tourists have limited leisure time, they don't want to spend too much time making plans, they want more time to enjoy traveling. Therefore, our recommendation system is of great application and commercial value.

Another is traveling schedule. The arrangement of travel schedule, due to its complexity, is a time-consuming and laborious work for tourists. This arduous task can be divided into several steps. The first is the prediction of scenic spots. It asked visitors to know the opening hours, tour times and commuting times of each attraction. The next step is to allocate the sites to different times according to the daily travel time limit. This takes into account rest and meal times each day. Meeting these constraints while achieving a satisfactory timetable can be a lengthy and painful process that requires a great deal of coordination.

Because of these tedious processes, tourists hope to have an automated system to help them achieve all the work from scenic spot recommendation to time planning, so that tourists only need to input their preferences, and then can get a complete tourism planning.

In this project, our goal is to first recommend scenic spots for users according to the rules established in the rule system, and attach the introduction of scenic spots according to the preferences input by users, such as whether they like lively places or travel with their families. The knowledge gained from the course of reasoning system is applied to construct the prototype of automated travel time planning to promote more effective time planning. In addition, we strive to evaluate the benefits and limitations of these systems to help business users make smarter decisions and make our projects more meaningful.

3 Knowledge Model

Knowledge model is combination of 2 main aspects:

- Knowledge Acquisition
- Knowledge Representation

3.1 Knowledge Acquisition

The main data acquisition technique we use is crawler. We've scoured about all the spots in Singapore from different travel websites, including spot types, opening hours, spot profiles, location, etc. In addition, we conducted case study to better design and implement our project. The details are as follows.

No.	Information Source	Acquisition Technique	Information
1	Online Media	Web scraping to get all spots detail	It provides basic information on each spot such as: -Open time -Location -Type -Introduction
2	Case Study	-In-depth analysis of Intelligent Rental Recommendation System -In-depth analysis of Course Schedule Optimization	Design, architecture and business process.

Table 1 Knowledge Acquisition Method

3.2 Knowledge Representation

3.2.1 Recommendation Part

Acquired knowledge was further refined to identify the crux of requirements and represented in the form of Inference Diagram. This process is very important as it clearly highlights the goals of the system and can be used directly to start implementation of knowledge-based system.

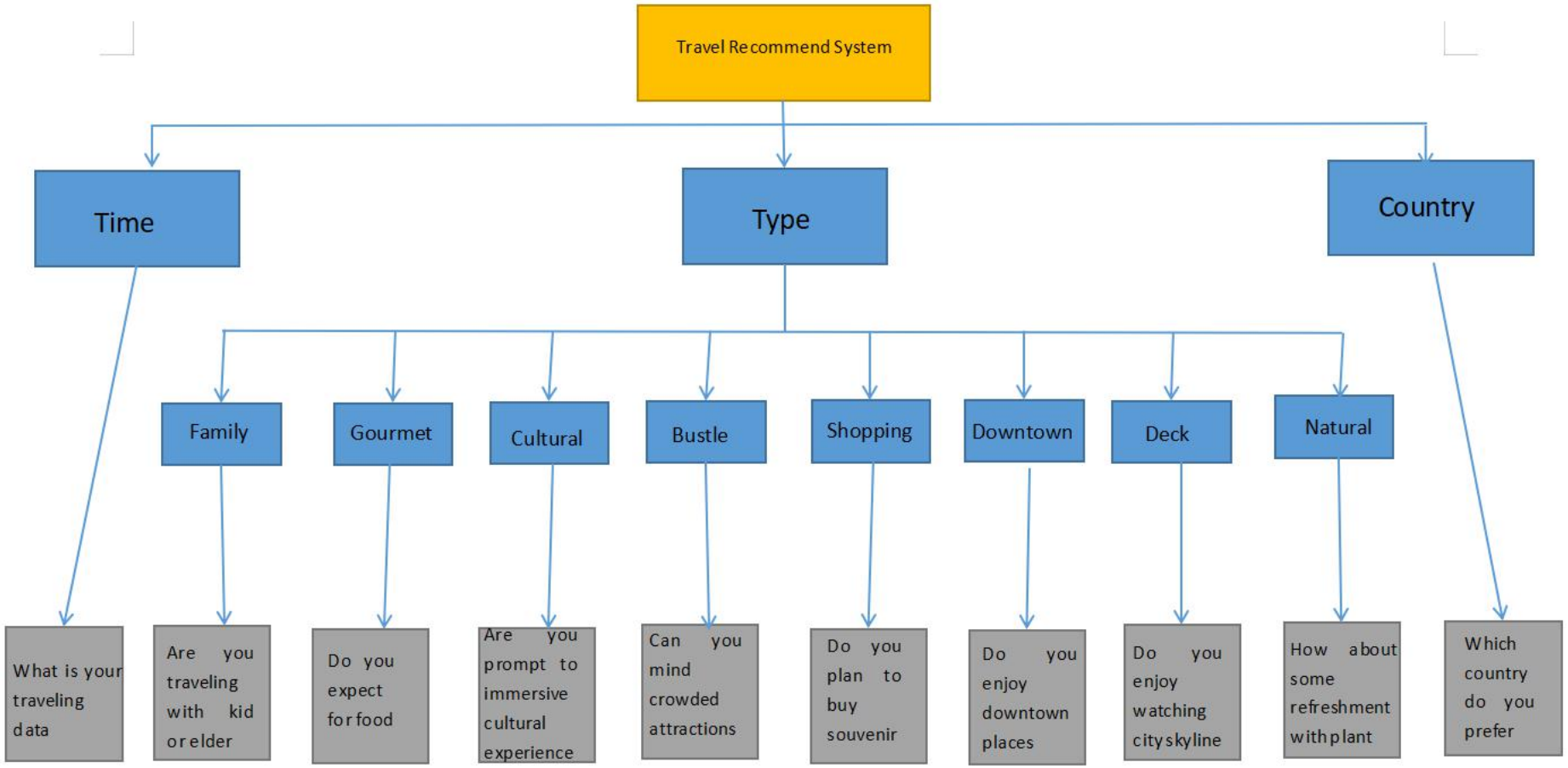

Figure 1 Inference Diagram of Traveling Recommend System

As shown in the Figure 1, main goal of recommendation system can be divided into 3 inferable sub goals which are further divided into sub-parts to get the information from observables (Questions asked from the user). The process is to get all the information from the user as answers of the Observables, infer the requirements by going through the set of rules assigned for each observable and defining the conclusion based on the results of the sub-goals.

The rules of the system are based on user preferences and spots types. First, we need to match each scenic spot with its corresponding tag. For example, the tag of Singapore eye is “SceneWatching”, and the tag of orchard road is “Shopping”, etc. In this way, when we recommend scenic spots to users, we can input corresponding tags for them to return the scenic spots under the same tag. Secondly, according to the survey of tourists and their travel experience, we determine the types of scenic spots corresponding to different tourists' preferences. For example, when tourists travel with the elderly and children, we recommend scenic spots under the Family tag. When users want to contact with nature, we will recommend scenic spots under Natural label. In this way we can match our front-end user input to the appropriate rules.

3.2.2 Schedule Part

Soft Constraints

Soft constraints are a set of conditions that should be satisfied whenever possible.

Item	Description
Minimize Travel Time	Visit the next spot that’s closest to the previous spot in geography.
Minimize Unmatched SpotSnippet	The less spot not arranged, the better; The more important (with higher weight) spot getting arranged, the better.
Arrange Spot to its Fullest	The closer the length of time period arranged to one spot to its estimated visit duration, the better.

Table 2 Soft Constrains

Hard Constraints

Hard constraints are a set of conditions that must be satisfied.

Item	Description
One TimeCapsule to One Spotsnippet	Exclusively assign time slot (capsule) to each spot snippet
Visit Spot Once	No interruption within period assigned to one spot by other spot
Reserve Gourmet Spots Exclusive to Meal Time	Spots of "gourmet" type (e.g. Maxwell Hawker Center) should be visited within meal time exclusively.
Reserve Travel Time Between Spots	TimeCapules whose neighbours are assigned to different spots should be left blank for travel time
No Exceed Time of Spot	Spot cannot arranged too early or too late, given the start and end time of each day.
Visit Within Time Window of Spot	Spots should be visited within their operation hours.

Table 3 Hard Constrains

Problem Modeling

Travel Recommend System is a two-dimensional timetabling problem that deals with the assignment of different spots in different time of each day to certain constraints.

Schedule	Day 1	Day 2	...	Day N
Time 1	Spot 1	Spot 1	Spot 1	Spot 1
Time 2	Spot 2	Traveling Time	Spot 2	Rest Time
Time 3	Eating Time	Spot2	Spot 3	Spot 2
...	...	...	...	...
Time 4	Spot N	Spot N	Rest Time	Spot N

Table 4 Model of Schedule

4 SYSTEM IMPLEMENTATION

4.1 System Architecture

We used the MVC framework to design an overall application, which is easy to use for anyone looking for a traveling recommendation. The design of the system considers the main aspects of user friendly and provides ideal recommendations based on user preferences.

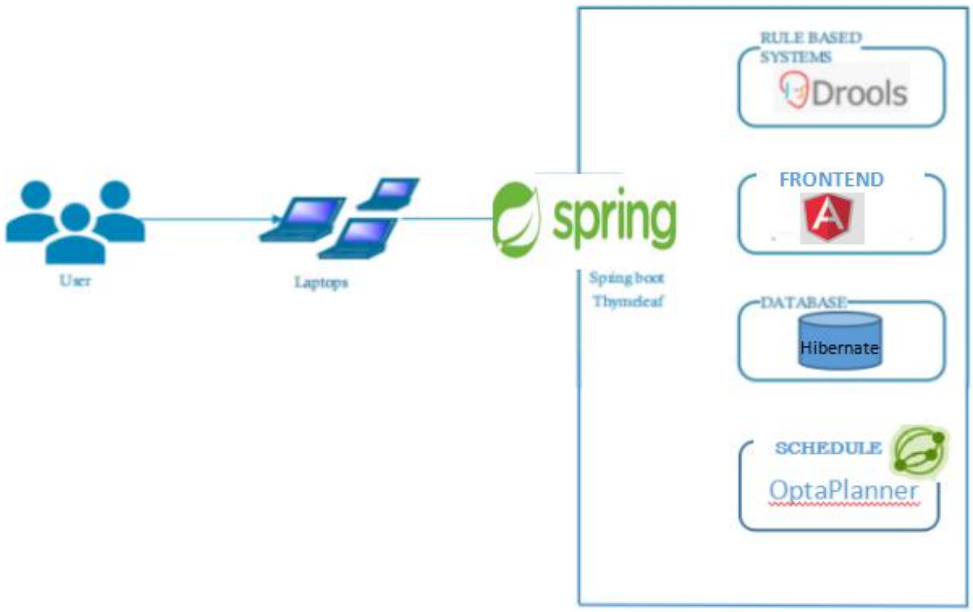

Figure2 System Architecture

The application was designed on the Spring bot, a lightweight MVC framework that provides an easy-to-use web-based front end that interacts with rules designed in kie-drools. It is a modern server-side Java template engine for web and standalone environments. Springboot completes the controller portion of the application by defining queries based on answers in the GUI and guidelines in the kie-drools. The Excel spreadsheet contains all the attractions options with defined properties. Spring boot collects all the recommended attractions from Excel and displays them in the front end.

Data scraping methodology

Our input data is crawled from various travel sites, including Ctrip, TripAdvisior, and Wiki. We used a Python based tool called Beautiful Soup to scrap and extract data

from individual listing pages from above listed sites. The information we crawl mainly includes the opening time of scenic spots, type of scenic spots, geographical location and so on. All the data are sorted out and stored in .csv format

4.2 Scope and Assumption

The system is mining the data sets available in the database, which are created based on the various properties that spots can contain. The purpose of the project is to show that an effective inference system can be used from the available data sets my ideal choice is based on the rule set defined in the rule KIEDrools guidance. The system will generate a set of recommended listing criteria based on the user's various questions, we will query the database based on search criteria.

To effectively demonstrate the business value and functionality of the product, the following scope and assumptions are defined.

(1) For this project, we only support attractions and spots within Singapore, and the default duration of each “time capsule” to 30min.

(2) Planning days are defined by users’ selection.

Certain assumptions are made in the development process to simplify the process of recommendation wherever it is required.

Assumption 1: Target audience is mainly people want to visit Singapore. The system is not only beneficial to those who have some knowledge of Singapore's attractions, but also to those who have no knowledge of Singapore tourism at all。

Assumption2: The rules of our system are determined according to some tourists around and their own experience, that is, the sample size of the survey is limited. Here we assume that these rules apply to most visitors.

4.3 KIE Rules Design

The inference diagram as shown in Table 6 is an important starting point to develop the system. The observables identified in the inference diagram are used to capture the preferences of the users and assigned as an attribute to the RULE BASED SYSTEM (KIE-Drools).

SUBGOAL	Questions	Attribute	Inferable or Observable	KIE Field Type & Value	Guided Rule
KIE Data Model: Data Object	Front End	KIE Data Model: Objective Field	KIE Form: User Interface	Data Type	
Time	What is your traveling data	Traveling Duration	Observable	Float	
Type	Are you traveling with any kid or elder	QnsFamily	Inferable	Boolean	If(with kid or elder)
	Do you Expect for food	qnsFoodExpectation	Inferable	Boolean	IF(expect food)
	Are you prompt to immersive cultural experience	qnsCultrual	Inferable	Boolean	IF(experience culture)
	Can you mind crowded attractions	qnsCrowded	Inferable	Boolean	IF(mind crowded)
	Do you plan to buy souvenirs	qnsShopping	Inferable	Boolean	IF(shopping)
	Do you enjoy downtown places	qnsDowntown	Inferable	Boolean	IF(enjoy downtown)
	Do you enjoy watching city skyline	qnsView	Inferable	Boolean	IF(see skyline)
	How about some refreshment with plant	qnsNatural	Inferable	Boolean	IF(connect with nature)
Country	Which country do you prefer	qnsCountryID	Observable	int	

Table 5 Attribute Worksheet

The Attribute Worksheet defines the data model and introduces the rules which is assigned in the KIE-Drools system. The Questions asked from the user at the front end (Web-UI) are directly linked to the attributes in data model and Guided Rules. Next, Guided Rules are clearly identified by the Rule Numbers as shown below:

Rule No.	Condition	Subgoal
F1	qnsKidElder = True	Family = True
F2	qnsFoodExpectation = True	Gourmet = True
F3	qnsCultural = True	Cultural = True
F4	qnsCrowded = True	Bustle = True
F5	qnsSouvenires = True	Shopping = True
F6	qnsDowntown = True	Downtown = True
F7	qnsNatural = True	Deck = True
R1	Downtown = False	Shoppingmall = False
R2	Bustle = False	themepark, zoo, landmark = False
R3	Shopping = False	shoppingmall = False
R4	Scene watching = False	bridge, deck, landmark = False
R5	Cultural = False	museum, religious, historical, neighborhood = False
R6	Natural = False	park, garden, island, beach = False

Table 5 Guided Rules

Based on the knowledge found in interviews and surveys, we generated a list of business rules tasks that satisfy each type of knowledge pattern. For example, old people prefer to visit natural scenery, while young people prefer to go to places like amusement parks. Another example is that when people travel with their families, they tend to choose more family-friendly attractions, such as zoos, Chinatown, etc. We used this knowledge to build a process flow that checks each category in turn to evaluate the ideal tourist attractions to query for the user.

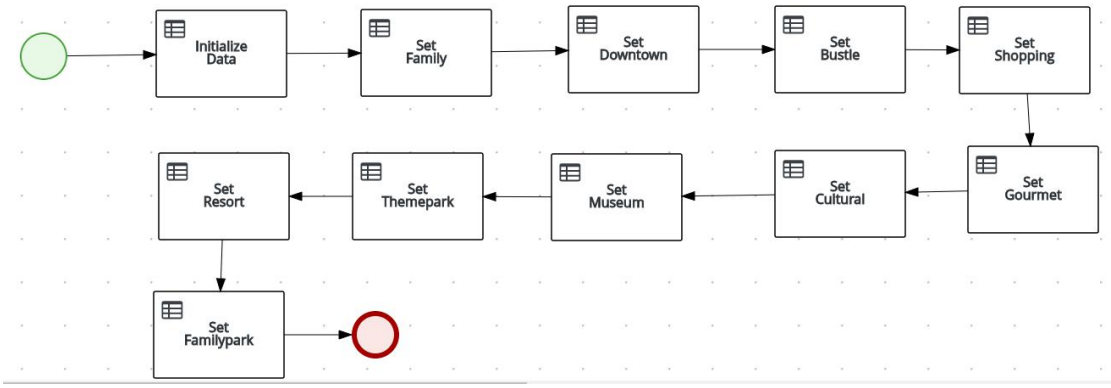

Figure 3 KIE Business Process

Figure 4 KIE Guided Rule Example

4.4 Schedule Design

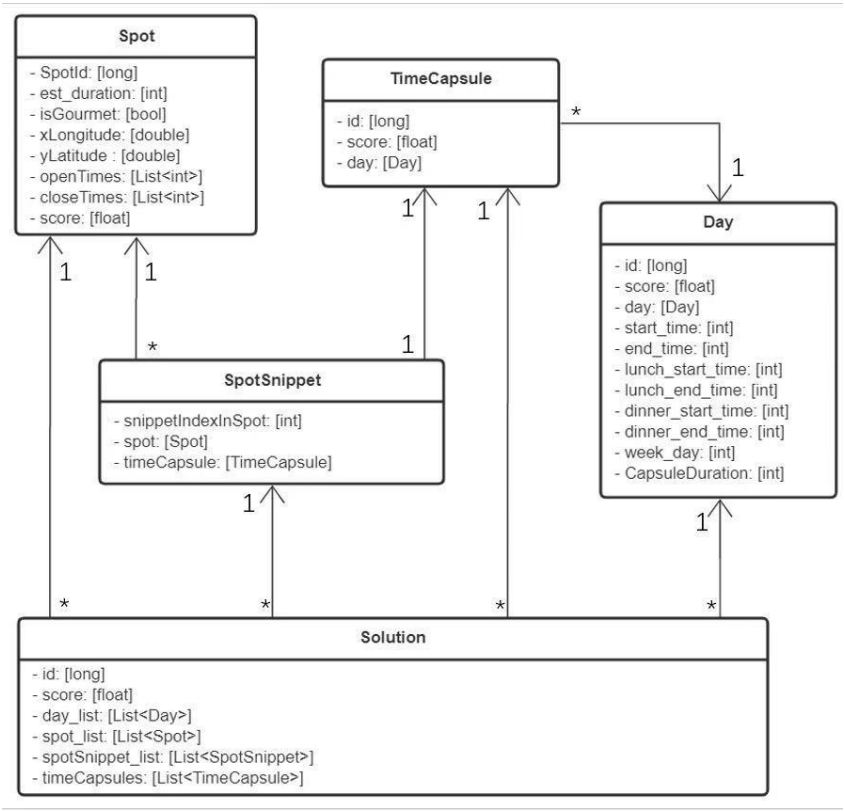

Figure 5 Domain Model of Schedule

5 SYSTEM FEATURES

5.1 System Capabilities

Starting from knowledge acquisition, we obtain data from different sources and stir data. We use different knowledge models to find useful information from data and defined rules, and then mine data from available resources.

Because the system is designed as a web application, accessing the application through the user interface is simple. See the user manual for installation details of the application. The UI is designed on SpringBoot, a modern server-side Java template engine for web and standalone environments.

Ease of access

The front end of the system is a web interface that allows users to select data as back-end input. In order to free users from the tedious data input work, users only need to choose their preferences according to the system questions at the beginning, and the system will automatically recommend the corresponding scenic spots for users and complete the schedule of the travel plan.

System Scalability

The system was developed as a micro-service that could be extended to accommodate future high workloads.

5.2 Area for Improvement

Since our system design is based on the micro service-oriented architecture provided, it is flexible and scalable. The team believes that enhancements may be added in the following areas in the future

Muti countries

Due to data limitations and time constraints, our project currently only recommends and schedules scenic spots in Singapore. In the future, this project can be further extended to recommend and schedule attractions in other countries. And it can recommend corresponding countries and scenic spots for them through user input.

Enhance System function

Spot recommended result can be effected by many features and these factors vary with the population. Our current system is based on a number of important factors. Some less influential features are not considered in our system. In the future development, we can add more features to the system to make our rules more perfect, so as to make more humanized and targeted recommendations.

Enhanced User Interface

At present, our UI is relatively simple and can meet the current needs. As our system functions further expand, our UI will also be optimized to meet higher user requirements.

6 CONCLUSION

Our system is a good example of intelligent reasoning systems and machine reasoning. It contains recommendations and schedules for tourist attractions. First of all, this project has high commercial value, which can be promoted to the market for application and make up for the vacancy of travel recommendation and scheduling in the market. Second, the system is also very practical, it can meet the needs of different tourists, whether they know the destination or not. At the same time, our system has a friendly user interface, users can use our system very easily.