ISY5001-MACHINE REASONING

Master of Technology in Intelligent Systems

ITINERARY PLANNER (SINGAPORE)

"Personalised Travel Strategist"

TEAM : SANGAM

SANTHOSH KUMAR MOHAN (A0198528L)
GAUTHAM BALASUBRAMANIAM (A0198478B)
MERCEDES PREMALATHA RAMESH (A0198411A)

TABLE OF CONTENTS

1. Abstract

1.1. Description

A Singapore vacation reveals the many charms of a leading regional hub of culture, retail, and tourism. Tourism is one of the fastest growing sectors in Singapore. According to 2018 Q4 report of Singapore Tourism Board, tourism sector achieved record highs in International Visitor Arrivals and Tourism receipts for the third year consecutively. Around 18.5 million visitors visited Singapore in 2018 and it accounted for Tourism Receipts of S$26.9 billion.

Due to the rapid growth of this sector, tourists can easily be overwhelmed with the options they have. However, they are faced with the challenge of identifying popular places aligned with their personal interests. Surfing through the internet, sifting through blogs, magazine articles and sites like TripAdvisor can be time-consuming and unorganised. In order to give the customer a better personalised travel experience and to facilitate further growth in country's tourism sector, an application was built that proposes a set of scheduled itinerary. Every itinerary is tailor-made from scratch to fit the user requirements.

1.2. Vision

The complete aid to bestow a unique hassle-free experience to explore authentic destinations of Singapore.

1.3. Contribution

In Artificial Intelligence filled future, the team built an application that proposes and recommends solutions in seconds. This application starts with a questionnaire where the user inputs details like type of traveller, number of days of travel. The application then processes through a set of rules, optimises the places and provides list of itineraries to fit the individual requirements. The team was able to apply the concepts of Machine Reasoning, Knowledge Representation, Knowledge Acquisition,

Knowledge Discovery, Contemporary Reasoning, Search Based Intelligent Systems, Reasoning using Optimisation Techniques etc.

2. Introduction

2.1. Overview

Problem: Touring is a popular but time-consuming activity due to the need to identify and structure tourist attractions or places of interest in the form of a time-constrained tour itinerary. Each user has specific characteristics, such as preferred number of days and time, budget, convenience, interests etc., hence a standard itinerary cannot satisfy all users, as the places covered under that may not fit everyone's interests. Navigating through hundreds of websites manually and analysing the places to visit considering the time, location, ratings, price etc is a tedious and hectic process. Planning and scheduling are crucial to satisfy the trip.

Solution: To overcome the above stated problem we propose to build an application that recommends users with a list of tour itineraries based on their interests.

2.2. About

This application provides unique personalised travel recommendations. The recommendation is done considering the type of travellers, number of days of travel and their basic interests. The application finally provides list of itineraries that covers the places that has maximum user rating, schedules and maximises the number of places to cover. The user can also obtain a cost friendly itinerary upon request. The list also has the details of user ratings and entry fees for each place in final User Interface.

3. Data Processing

3.1. Data Extraction

To garner data regarding the tourist attractions in Singapore two sources were utilised.

- Application Programming Interface (API) from the Singapore Government's Tourism site: 'Tourism Info Hub(TIH)'
- Data scraping from 'TripAdvisor'

3.2. Data Cleaning and Organising

The data from the API had many missing values. In order to overcome this inconsistency additional data was garnered by scraping from TripAdvisor. Several transformations were applied to this additional data before incorporating into the original dataset to achieve homogeneity.

Since the process involved amalgamation of data from two different sources, data processing and cleaning were done before-hand. This method reduces the turn around time for responses from clips and OptaPlanner.

4. Application Architecture

Languages	: Python, Java, HTML, CSS, Javascript
Framework	: Django, Bootstrap, Spring Boot
Expert System	: CLIPS
Optimisation Tool	: OptaPlanner
Map API	: Bing Distance Matrix API, Google Map Embed API

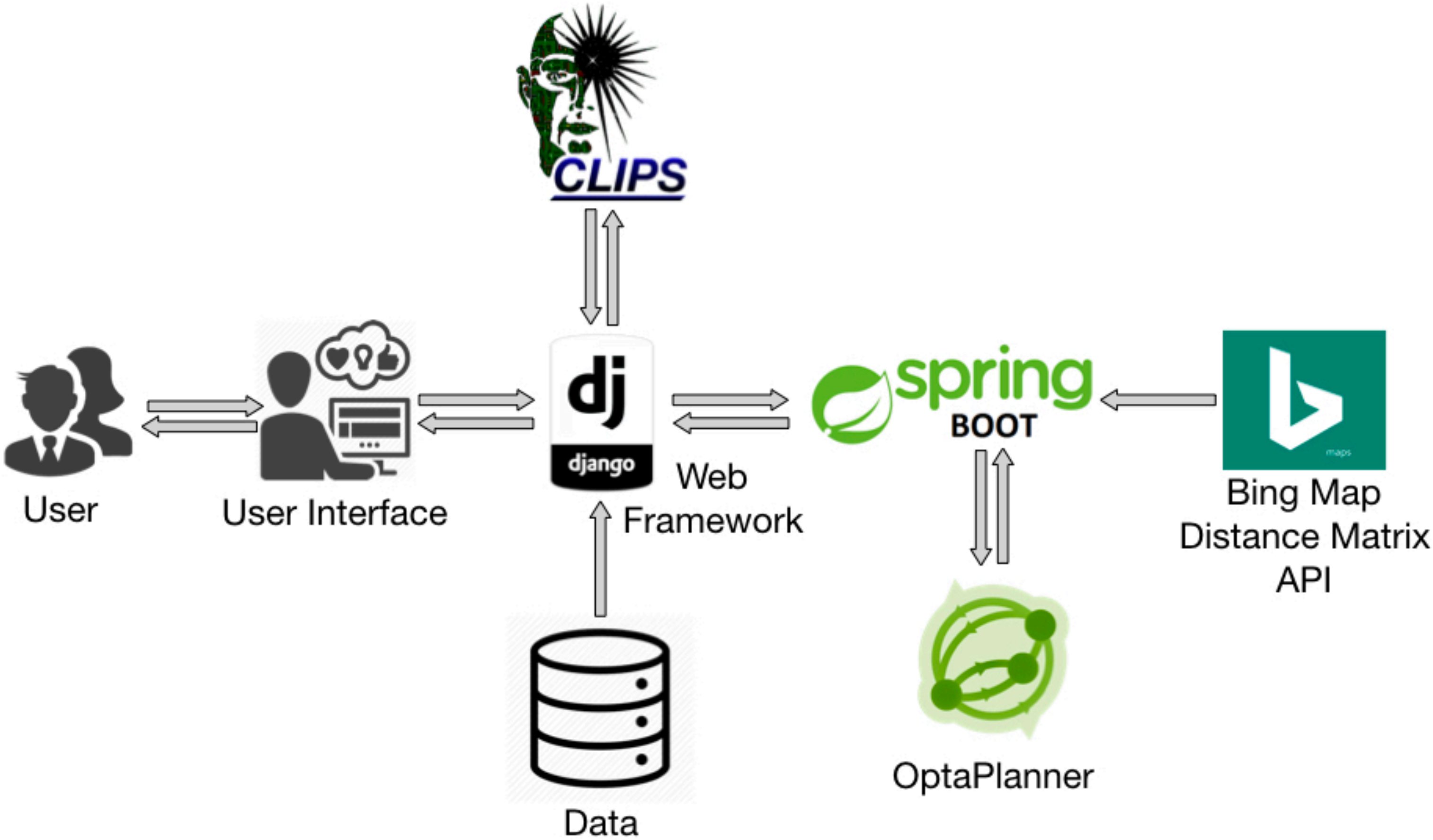

Figure 1. Architecture

- The user inputs the details like number of days, type of traveller, basic interest etc. in a webpage.
- The user data is sent to the CLIPS expert system and after every response the followup question is fired to the user.
- When all the user data is captured, the expert system based on the knowledge acquired, filters all the places that fits the rule.
- The above acquired set of places is sent to the OptaPlanner
- The OptaPlanner schedules the places to cover based on the user rating and location.
- The final itinerary is then displayed to the user.

5. Knowledge Modelling

5.1. Knowledge Acquisition

The collected data was carefully analysed and categorised into buckets. In addition to that certain assumptions were made after studying past tourism reports.

The segmented buckets and assumptions serve as the knowledge base for this application.

5.2. Assumptions

- If the traveller is visiting Singapore for first time, the top rated sight seeing places are recommended as default. If they have already visited Singapore then the followup question will demand whether they require to include the routine sightseeing places too.
- Solo traveller and Friends Group will have less or no interest for shopping, Arts and Culture, so the above stated domain places' bucket list is omitted.
- Couple and Family will have less or no interest in Fun and Entertainment, so that bucket list is omitted.
- For Family with Kids, the list of additional places related to education is included.
- For Family without Kids, the list of additional places related to Arts and Culture is included.

6. Knowledge Representation

The obtained knowledge is then programmed into fact templates and rules in CLIPS Programming Language.

6.1. Decision Tree

A Decision Tree was designed to represent the knowledge.

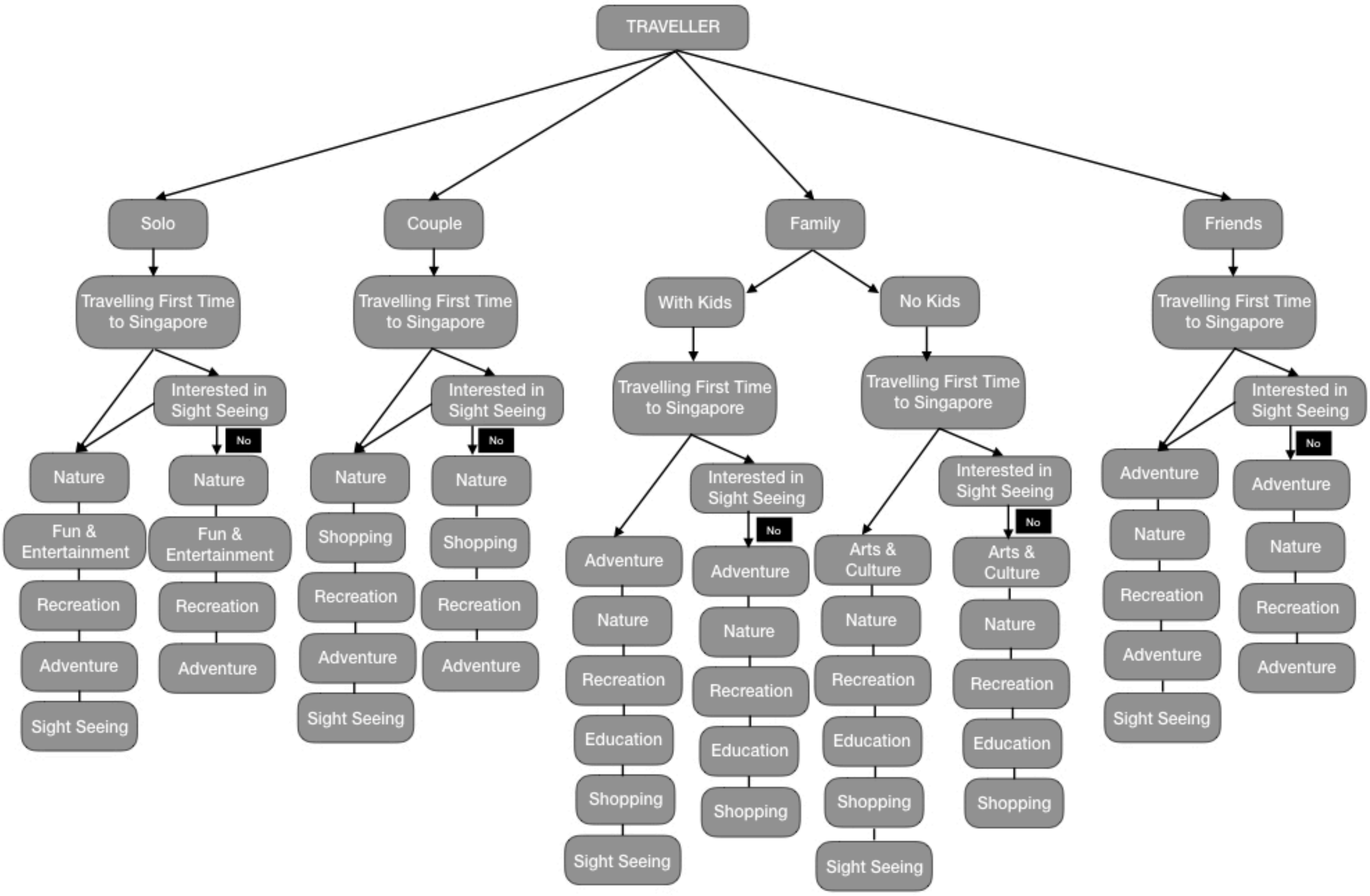

Figure 2. Decision Tree

6.2. Forward Chaining Expert System

Data-driven reasoning is sufficient as the problem requires inferences to be made based on previously gathered data. This is also known as Forward Chaining, where each time the topmost rule is executed. When fired, the rule may add new facts in the clips environment which may further activate many other rules to the agenda. The fire cycle stops when all the rules are executed. Upon execution, the list of all places that fits the rule engine will be sent to the OptaPlanner for scheduling.

7. Scheduling Using OptaPlanner

7.1. Optimisation Requirements

Output data from the Expert System is unorganised. There might be many places that match the user interest. The user will not be able to cover all the places in stipulated days, on contrary the places cannot be expunged. Hence, based on the user rating and the interests of the traveller, the places to visit are prioritised and scheduled efficiently. Those places are organised in such a way to reduce the distance travelled by the user. The shortest possible routes are calculated. An itinerary is prepared with places minimising the time spent traveling simultaneously reducing the cost incurred.

To schedule the order of places to visit and the routes, OptaPlanner with sophisticated AI optimisation algorithms were employed. First Fit Decreasing Algorithm with custom score calculation is used. The optimal order to visit destinations and the number of places to visit in a stipulated time is calculated.

Planning and scheduling is done considering the below criteria.

- Maximising the number of places.

- Prioritising the places based on user rating.
- Organising the places with routes.
- Minimising the cost (budget-friendly).

7.2. Constraints and Assumptions

- A user can utmost spend 12 hours per day.
- A user can cover maximum of 5 places per day (5 slots per day).

Based on the knowledge acquisition the number of hours required to visit a place was estimated. The estimated data was used to bucket places such that the places to see does not lump up overlapping in a single day.

For example:
Universal studios will take minimum 8 hours and hence user cannot to travel to many other places in that day. Also based on average performance, we have included a custom constraint of maximum 5 places in any one day (5 slots per day).

7.2.1. Hard constraints:

- Avoid overlap of slots and days between locations.
- At-least one place is filled per day.

7.2.2. Medium Constraints:

- Maximum number of hours to be spent by the user
- Maximise the visits to top attractions (maximise possibility of visiting places with high user rating)
- Minimise the total ticket price of the visited places (budget friendly)

7.2.3. Soft Constraints:

- Decrease the total distance travelled between attractions per day.

7.3. Class Diagram

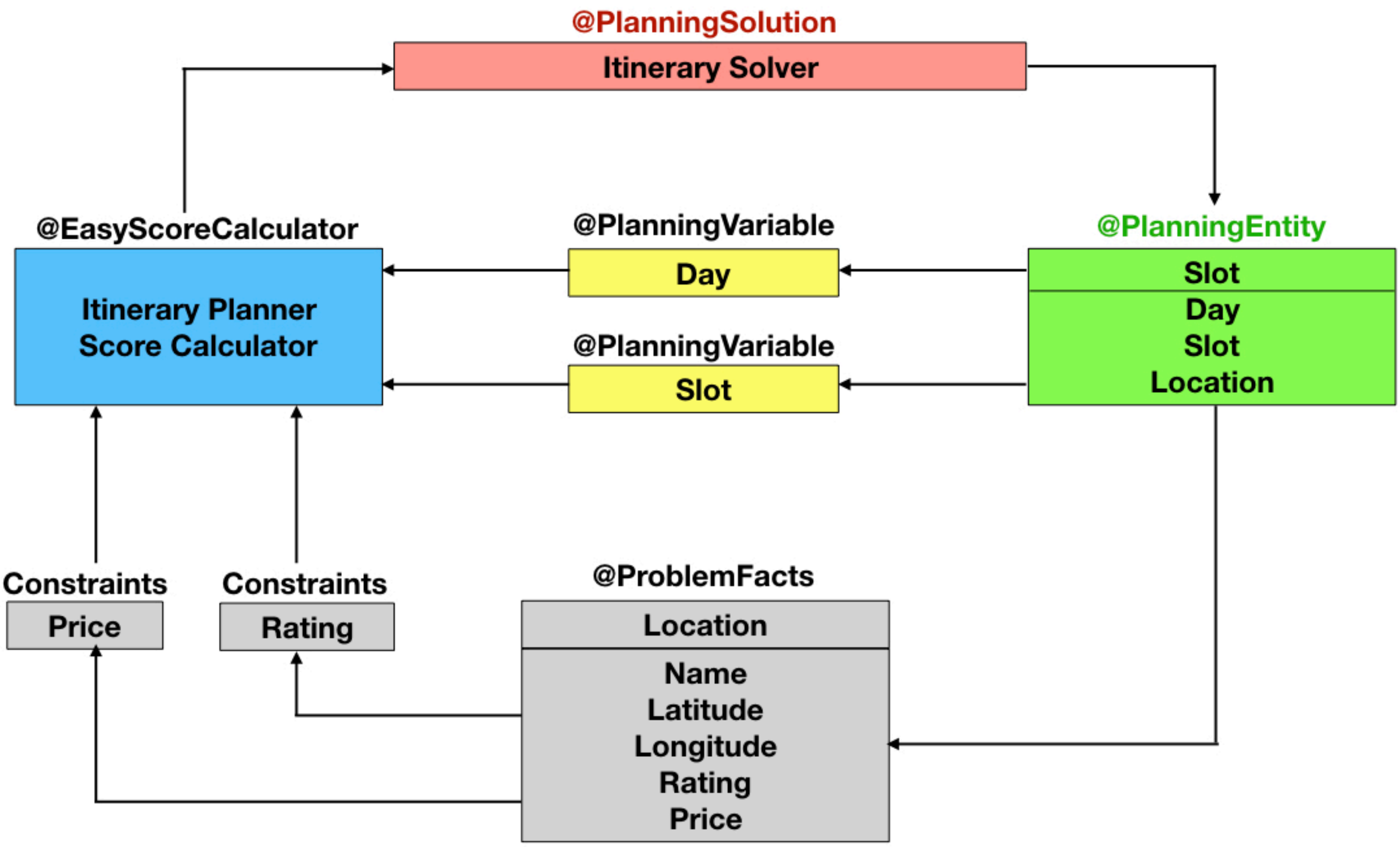

Figure 3. OptaPlanner Flow

7.3.1. Planning Solution (ItinerarySolver)

Planning solution implements solution interface and represents a possible solution in the search space. It contains planning entity list and problem facts for score calculation.

7.3.2. Problem Facts

Problem facts are immutable objects in the solution that are used to check the feasibility of the solution and for score calculation. Some of the facts used in this problem are:

- Distance Matrix(distance between any two locations)
- Duration Matrix(duration of travel between any two locations)
- Locations (contains user rating and ticket price of attraction)

7.3.3. Planning Entity (Slot)

Mutable entity class that contains planning variables (days and slots). One entity slot is created for each input locations. Days and slots are changed for each location while searching for an optimal solution.

7.3.4. Planning Variables (Days and Slot)

Two planning variables are used (days and slot) to change the itinerary and include or exclude various locations. These variables are used to go through the search space to create different itineraries and test their feasibility.

7.3.5. Easy Score Calculator (ItineraryPlannerScoreCalculator)

Java easy score calculator is used to check for the feasibility of the solution (hard score) and also compute the medium and soft score for the solution.

8. Additional Elements

8.1. Web Framework and User Interface

A web application was developed using python – Django framework. This application is used to communicate between user, CLIPS and OptaPlanner. The backend of the application which is responsible for handling the requests and generating response is developed using Django framework. The frontend web pages were developed in HTML, CSS and Javascript using Bootsrtap framework and Django Template System (jinja). Bootstrap handles the responsiveness of the UI on different devices with different resolutions. Django Template System is used to generate dynamic HTML contents from the backend application. This application prototype runs in Django development server which comes inhouse with the Django package.

8.2. Integration with Maps

Google maps API service was initially used to calculate the distance matrix, but the number of locations that can be used in the free tier is capped at 20. Alternatively, Bing Distance Matrix API was used to calculate the distance matrix. This distance matrix is used by the OptaPlanner to optimise the routes. The obtained route with multiple waypoints were embedded in the User Interface using Google maps API.

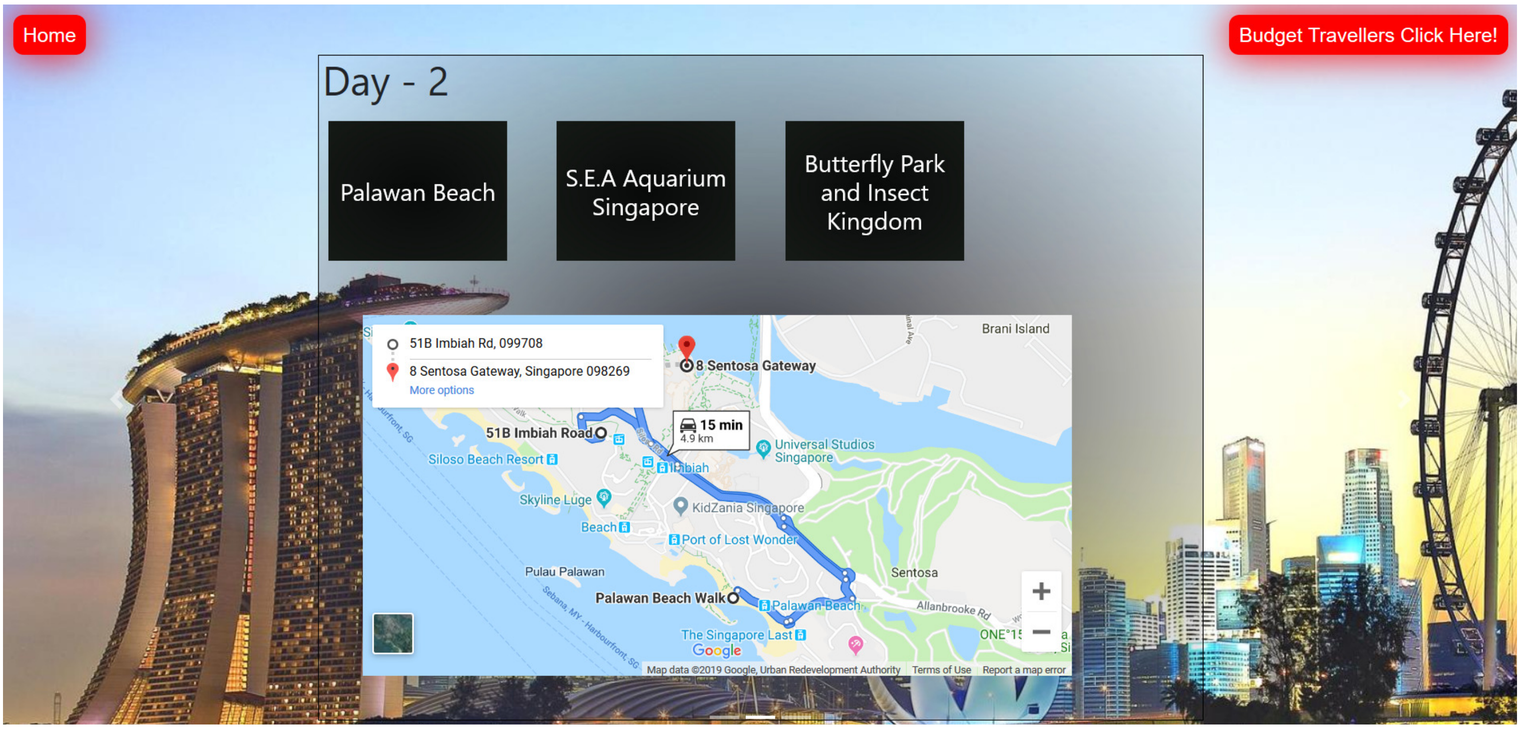

Figure 4. Integrated Map

9. Outlook of Application

9.1. Benefits

A Digital Travel Agent always with you!

- Choose from a unique inventory of local authentic attractions.
- Perfect for experiential trips.
- Flexible itineraries to match your interest.

A Recommendation Engine always working for you!

- Personalised advice based on your travel needs.
- Best places and the required time to visit.

A Planning Guide helping you to organise your days!

- Last minute planning.
- Recommends how much time to spend.
- Provides the opening hours.

A Complete Maestro in one place!

- Witness your entire must do activities through a single window.
- Efficiently organise your trips.

9.2. Challenges Faced

- One of the main challenges faced was cleaning and processing of data since the data was scraped from internet.
- Faced difficulty in organising places and Preparation of bucket list.
- Adding multiple PlanningVariables in OptaPlanner
- Integrating CLIPS with Python and OptaPlanner
- Developing a responsive User Interface.

10. Conclusion

10.1. Future Enhancements

In addition to the developed features, the following improvements are part of the roadmap for future.

- Recommendations to groups of tourists with diverse interest preferences.
- Automatically build a tourist interest profile, possibly by analysing a tourist’s social media profile.
- Incorporating ongoing promotions and offers on the entry fee of the attractions.
- Predict and consider weather forecast for recommendation.
- Recommending tour itineraries by considering the public transport arrival & departure time to facilitate realistic tour planning and minimise the public transport waiting time. Real time uncertainty of public transport could be modelled.
- Hosting the Web Application in a Cloud server.

10.2. Contented Cessation

The outcome of the project is a solution for a pragmatic implementation that can be commercially employed. We developed our intellectual and team work skills. The discussions among us pitched in increasing the familiarity of Machine Reasoning, Reasoning Systems and its practical implementation. It stimulated our thinking and deepened our understanding. This opportunity encouraged us to apply, test and consolidate our own learning.

Thank you.