GROUP-11 PROJECT

MTECH (INTELLIGENT SYSTEMS)

SEMESTER 1 - INTELLIGENT REASONING SYSTEMS

TRIP @ US

A0213523U	*MEHTA VIDISH PRANAV*
A0213496B	*ANKEIT TAKSH*
A0198539H	*WANG KUAN-KAI*
A0213514U	*ANANDAN NATARAJAN*

TABLE OF CONTENTS

1. EXECUTIVE SUMMARY

Imagine yourself planning for your next overseas trip to Singapore, doing countless hours of research on the attractions to visit and getting vexed over the perfect hotel for your family. The task of planning a trip can become extremely daunting for tourists who need to plan from selecting the attractions, booking the accommodations and charting their daily itinerary.

Indeed, according to the 2018 Q4 report published by the Singapore Tourism Board, the tourism sector had achieved record highs in International Visitor Arrivals and Tourism receipts for the third consecutive year. With the exponential increase in tourists to Singapore, the primary challenge of making full use of their trips while taking into consideration their limited time here is enormous. Based on our research, an average tourist usually takes up to 12 hours to plan their trips. Currently, there exists a sizable market gap in the tourism industry with modern travel tech players like Traveloka and Agoda provide limited active recommendations whilst the more traditional travel agencies tend to have a strict adherence to schedule and lack the much-desired flexibility. From our initial survey, we discovered a strong demand for a middle ground that combines the flexibility of planning your own trip yet provides active recommendations on places to visit.

These thought-provoking discoveries led our team to explore an efficient recommendation system which allows tourists to effectively plan their trip in Singapore. Leveraging on our core competencies in intelligent reasoning, cognitive and optimisation systems, we have developed a dynamic itinerary planner which optimises both the hotel location as well as daily itinerary for our end users, thereby relieving them from the painstaking planning process. Our simple and intuitive front-end user interface hosts a dropdown form for tourists to input their preferences for common categories of attractions in less than 10 seconds. Linear programming optimization solver optimizes the attractions based on earlier defined user preferences and curates a list of attractions for our end users. With attractions selected, our end users can now book their recommended hotel, obtained by minimizing the distance to each of the attractions. At the heart of our itinerary planning process is a hybrid reasoning engine that combines a greedy best first search tree with the permutations of genetic algorithms. This computationally intensive task is executed asynchronously. All these with an aim of allowing our end users to relax and to receive a specially curated and customised itinerary sent directly for their use.

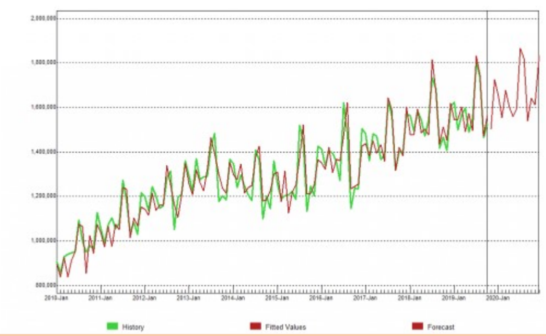
But wait, there’s more. We do not just stop at planning as we understand tourists desire a dynamic planner that can plan on the go. As the founder and CEO of Amazon, Jeff Bezos puts it “Any plan won't survive its first encounter with reality. The reality will always be different”. Our humanoid telebot accompanies our tourists throughout their trip in Singapore and dynamically plans through any changes in the schedule or preference. Our aim at Trip@Us is to be your friendly trip recommender system that follows you throughout your journey.

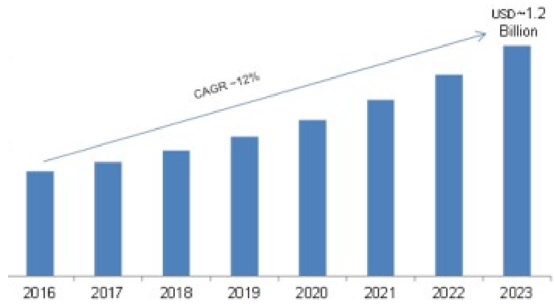
2. BUSINESS PROBLEM BACKGROUND

2.1 MARKET RESEARCH - TOURISM INDUSTRY

The number of tourists in Singapore has shown a steady increase over the years. Post COVID-19 we expect the tourism industry to boom further

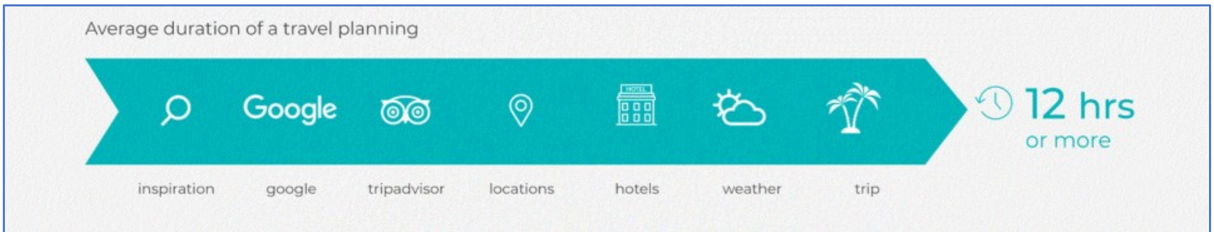
The tourism industry in Singapore has seen a projectile growth in the previous years and is projected to further increase





The tourism industry is a multi-billion-dollar industry in Singapore. Based on the 2019 report published by the Singapore Tourism Board, the tourism sector had achieved record highs in International Visitor Arrivals and Tourism receipts for the third consecutive year. Singapore has attracted approximately 19.1 million visitors in 2019, which is 3 times the population in Singapore, with a revenue of SGD 27.1 billion dollars generated. Year on year, the tourism industry in Singapore is growing at an effective rate of 6.2%. The challenges of planning and maximising a traveller's itinerary still persists and there is no viable solution that bridges a dynamically smart recommender system with the flexibility of travel at your own pace in the market yet.

2.2 PROBLEM AND OPPORTUNITY IDENTIFICATION



From our initial survey, we realised that present-day tourists tend to shelve the idea of travelling with tourism agencies due to its rigid and routine schedule. Instead, they desire a recommender system which is able to ease their planning load. On an average, it takes up to 12 hours to plan for a trip and the biggest pain point faced by most planners is not having all in-one-stop channel and therefore relying on various agency platforms and different travel websites. Tourists planning their trips to Singapore need to go through countless blogs, Instagram posts and travel websites before charting out an itinerary that coincides with their preferences.

Yet, even after spending countless hours on planning, research suggests that only 10% of the trips actually pan out the way they had been planned. Thus, in reality trips are more dynamic in nature and could constantly change according to unexpected situations, fatigue and user preferences. In such scenarios, tourists often desire a portable tool that can plan their trip on the go.

2.3 MARKET COMPETITOR ANALYSIS

<table>
<tr><th>TRAVEL SERVICE PROVIDER</th><th>TRIP PLANNING AGENCIES</th></tr>
<tr><td>agoda
Booking.com
airbnb</td><td>

CITS
中国国旅</td></tr>
<tr><td>1) No optimisation of hotel’s location based on traveller’s needs and selected attractions
2) No proactive recommendation
3) No leverage of Intelligence System</td><td>1) No flexibility in the planned itinerary
2) No leverage of an intelligence system to provide an itinerary which saves the most time
3) No proactively recommendation of an attraction near you</td></tr>
</table>

The market competitor analysis aims to study major players in the tourism industry. Our dominant competitors fall mainly in two categories travel service providers and travel planning agencies. From the above analysis, it is clear that there is a vacuum in the tourism industry where there is an absence of a service provider that caters to both smart recommendations and offer flexibility in the planning of itineraries. While there are smaller players like Pebblar (a map-based itinerary service provider) targeting this niche market, their services lack dynamic planning and are priced too expensively. Our Trip@us core team believes that there is a sea of opportunity that lies in the middle market – one that offers personalised recommendations and flexibility in schedules.

We can conclude from the above chart that there is no dominant competitor that offers a solution to this catch-22 problem currently. There will be room for our product to grow in the market and be recognised as a service provider who provides an end-to-end solution to travellers using our intelligent systems.

3. PROJECT OBJECTIVE & SUCCESS MEASUREMENT

3.1 PROJECT OBJECTIVE & SCOPE

With a clear perspective on the target market and a thorough market analysis, we can proceed to set our main objective.

Our project objective is to leverage on our team's strong conceptual knowledge in search algorithms to design an intelligent reasoning system that accompanies our end users in every stage of their planning process from selecting the attractions, to booking their accommodations and planning their trip. Our search optimisation techniques mainly aim at minimizing the distance travelled by our tourists so that they are able to spend more time at their sightseeing attractions. In order to achieve our objective, the optimization of accommodation is mandatory. Assisting our primary objective, our system must also be able to dynamically respond and react to any changes in schedule, thereby actively planning and assisting our end users during their actual trip.

Another key objective is to also ensure the scalability of our system to support the strong user traffic flow to our application. Due to the computational intensity of the core hybrid reasoning engine, for the system architecture to sustain the large amount of requests from our end users, asynchronous task processing becomes imperative. Our system also needs to incorporate a credential management system to log user data for enabling active planning during their trips. With the enablement of active planning via our telebot interface, it is essential for our system to strictly adhere to the Personal Data Protection Act of Singapore by requesting user authorization before obtaining their location. This ensures that we do not encroach our tourists privacy but rather provide active recommendations whenever they need them.

Feedback systems can be an avenue for our tourists to recommend further improvements to our systems. In this way our system can complete the cycle of interactions with our end users and foster a machine learning pipeline that will facilitate further strengthening of our specially curated and customised backend databases.

3.2 SUCCESS MEASUREMENTS

Success measurement metrics are intrinsic to the execution of this project. We developed SMART (Specific, Measurable, Attainable, Relevant and Time-based) goals for the timely delivery of our project. Due to the scale of the project and ensuing challenges faced during the development of our system, our team came up with a few key metrics to prevent us from being too myopic in solving problems and missing on focusing on the bigger picture at hand. Furthermore, we leveraged on the key strengths and core competencies of our team members to run this project from inception, conceptualizing the solution, program logic, User Interface definition, development and operations, and accompanied by structured meeting minutes and project documents.

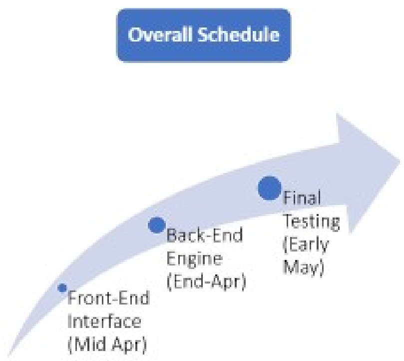
At the inception of the project, the success measurement of the project was divided into the following key zones:

1. Schedule
2. Solution Quality
3. Customer Satisfaction

In order to meet the process objectives in the stipulated timeline, the project was divided into three main sub objectives (front-end interface, back-end engine and final testing) with proposed timeline for each of the sub-objective. The objectives were met as per the specified timeline, as shown in the image below.

In order to test for the solution quality, we tested for the robustness of our solutions with providing extreme inputs in terms of hotel rating and price as well as travelling preferences. We continued to enhance and test the system until we were able to achieve 100% success rate in execution for boundary condition as well as system accuracy.

Finally for customer satisfaction, we have created an avenue for feedback on using our system. This will form the key metrics of our success measurement.



Solution Quality

	System Score
Solution Robustness	100% at all system boundary conditions
Solution Accuracy = total success runs/total tries	100%

4. KNOWLEDGE ACQUISITION

OTR (Online Travel Reviews), eWOM (electronic Word-Of-Mouth)

Knowledge acquisition, modelling and representation is a critical part in the journey of creating an intelligent travel recommender system. Our team spent a considerable portion of time researching and studying the information flow and knowledge acquisition in tourism industry.

In today's information technology age, many travellers search for travel articles/recommendation posts through the internet. These articles/posts include the experience and knowledge of a traveller, which can be used as a reference for tourism planning and attraction selection. At present, there exists a hue amount of travel experience and knowledge is available in Online Travel Reviews (OTR). OTR and eWOM (electronic Word-Of-Mouth) contain a lot of self-help knowledge and advices for solo and group travellers alike. Many travellers often look for OTR content through virtual communities, blogs, and search engine. However, OTR searches have some limitations of its own. For instance, the search results often cause information overload problems. In addition, it still requires the use of keywords. In a foreign context, however, we recognise most travellers are limited by what they know and would most likely not know the name of the attraction. Therefore, they might not be able to search using the appropriate keywords, resulting in insufficient information gathered from OTR to make the best travel plan. Recognising this constraint, our project has opted to use an ontology-based tourist knowledge representation and recommendation method. The purpose is to search for popular attractions from OTR content and construct a tourist knowledge structure for our travellers.

Using semantic ontology, we were not only able to group attractions most often visited together, but also able to categorise attractions into various subsets such as architectural and cultural places, natural places, shopping districts, outdoor adventures etc. This categorisation of attraction helped in the formulation of linear programming needed for attraction selection as shall be explained in detail under project solution.

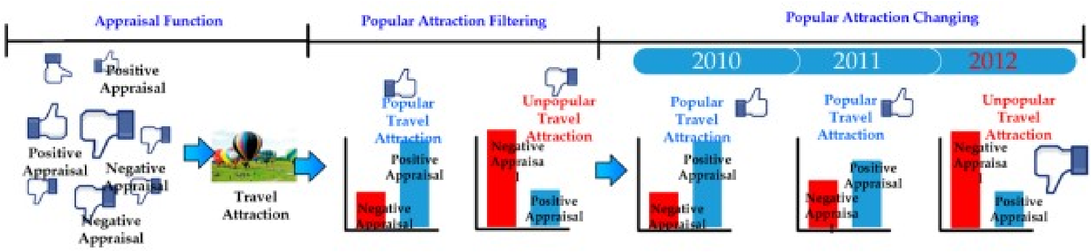
5.1 DATA COLLECTION

Data extraction forms a key aspect for the development of smart database systems, which eventually leads to smart solution. In order to harvest large amount of web data, a web data extraction system is required to parse through raw html and obtain information in plain text. Our team used web crawling as well as sound human judgement to construct a specially curated database of attraction and hotel list.

- WebCrawler for extracting information from travel service providers including traveloka, Agoda and TripAdvisor.
- Official government database on tourist and hotel attractions in Singapore
- Route info database from Google database

Apart from web extraction of data, we also studied several case studies to understand more on the changing landscape of tourism industry. Through a case study on Taiwanese citizens, we found that they pay more attention to the quality of their leisure life. Many families take advantage of holidays and summer and winter vacations to arrange simple travel itineraries or multiday travel plans. According to the statistics of the Taiwan Tourism Bureau among the many possible vacation styles for Taiwanese residents, self-planned trips account for the highest proportion. The statistics show that many people prefer to create custom-made trips that meet all of their expectations. However, to plan an itinerary, travellers must invest time and effort in advance. Therefore, a system that arranges personal travel itineraries, provides options for dining, entertainment, and accommodation, and uses a simple and quick procedure will reduce the time it takes for vacationers to schedule their travel.

Many studies have also shown that online appraisals determine user's willingness to purchase which also means that the attractions with higher positive appraisals is the consumers' favourite. Manually scanning through reviews on web platforms like TripAdvisor and Klook allowed us to extract a popularity score for each attraction amongst different group of travellers and this further allowed us to comprehend on the requirements of our tourists. It is also imperative to ensure that the pipeline of data extraction is continually active to detect and identify changing trends and patterns in our user's demands.



5. KNOWLEDGE REPRESENTATION & RECOMMENDATION

Before deep-diving into finding an effective solution, we proceeded to study some of the key factors that we will need to consider in our automated planner.

5.1 FACTORS INVOLVING THE PLANNING:

TOURIST SPOTS AND USER PREFERENCES

In order to automatically generate a travel itinerary according to the user's preferences, the system must search for attractions that match user interests when selecting stops in the itinerary. With the popularity score attained from scanning through multiple reviews, we were able to define a parameter called the likeability score. The likeability score $\boldsymbol{L_i}$ was represented for attraction i was represented as follows:

$$\boldsymbol{L_i = P_i \times U_i}$$

$$\boldsymbol{where\ P_i\ is\ popularity\ of\ attraction\ i;}$$

$$\boldsymbol{U_i\ is\ user\ preference\ for\ attraction\ i}$$

COST

Some tourists are less concerned about the cost of travel. However, others are more careful about spending. Therefore, users are required to provide an evaluation of their cost concerns especially during hotel recommendation process.

DISTANCE:

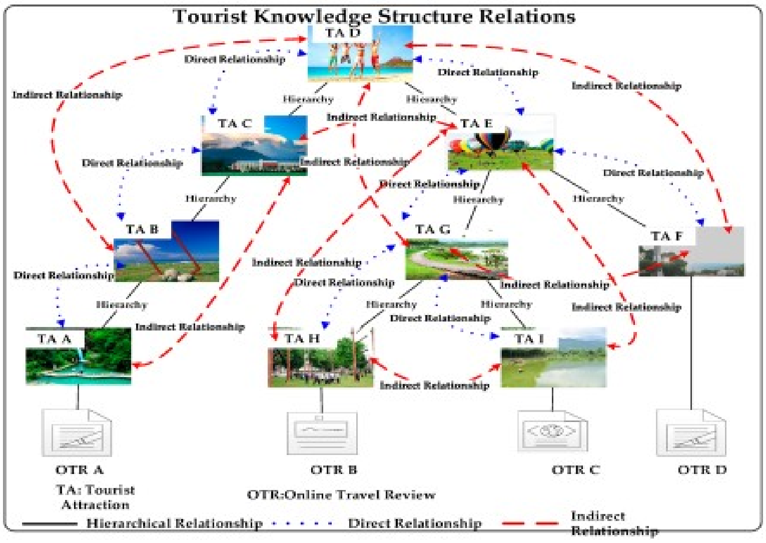
Distance can be divided into two parts. The first part is a calculation of whether the location is within the radius of the reference point according to its euclidean distance from latitude and longitude. The goal of the greedy algorithm is to identify the best tourist spot at the current stage; therefore, we require the tourist spot selection to be within a certain radius of the current position. Tourist spots beyond that radius are not included for consideration.

TIME:

The time factor will determine the itinerary number of days and the distance score. The goal is for the user to be entertained while traveling toward the destination. At the beginning of the journey, the system will select tourist spots relatively close to the departure point. As the tour progresses to the middle of the trip, the system will try to select tourist spots close to the midpoint between the start and destination.

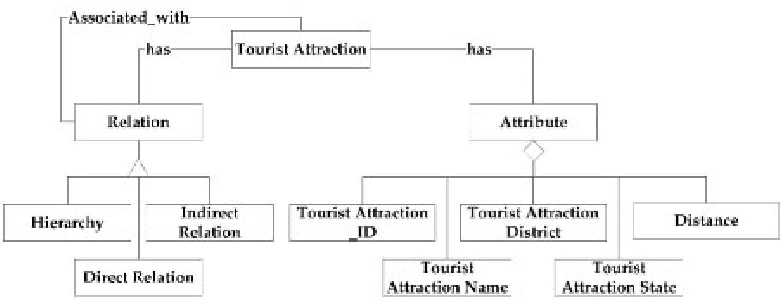
5.2 KNOWLEDGE STRUCTURE FOR TOURIST SPOTS RELATION:

This section defines the tourist knowledge structure, based on the hierarchy relation, direct relation, and indirect relation.



- Hierarchy relation: A hierarchy relation indicates OTR content that includes tourist attraction information.
- Direct relation: In addition to a hierarchy relation, there can also be a direct relation between attractions. There is a direct relation between a father node and a child node.
- Indirect relation: Since there is a direct relation between a grandfather node and a father node, and a direct relation also exists between a father node and a child node

5.3 CONCEPT SCHEMA

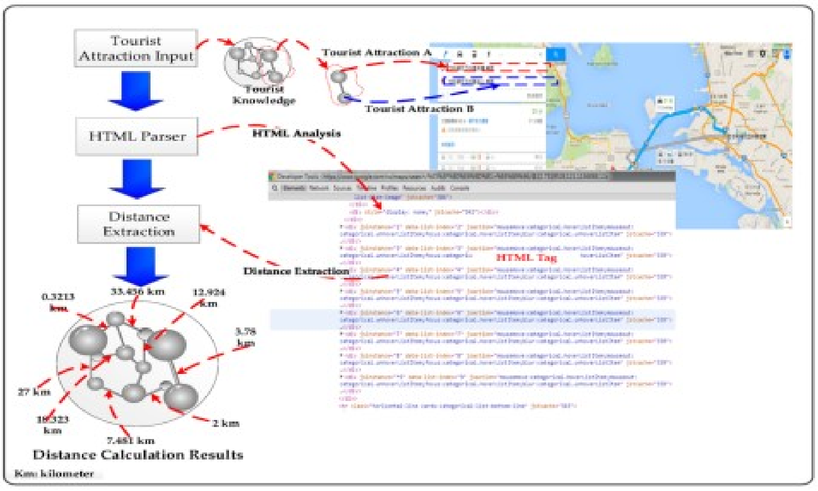


- Tourist attraction: Tourist attractions are the basic units that make up the tourist knowledge ontology, and this part of the schema records tourist attractions related to OTR content.

- Attribute: The tourist attraction attributes include (i) tourist attraction ID (number), (ii) tourist attraction name, (iii) tourist attraction district, (iv) distance.

5.4 DISTANCE CALCULATION

The distance calculation measures the distance between tourist attractions in the tourist knowledge ontology. In order to calculate this automatically, this study applied Google Maps API. The main procedures in the distance calculation are as follows: (i) obtain the tourist attraction input, (ii) use the HTML parser, and (iii) carry out distance extraction



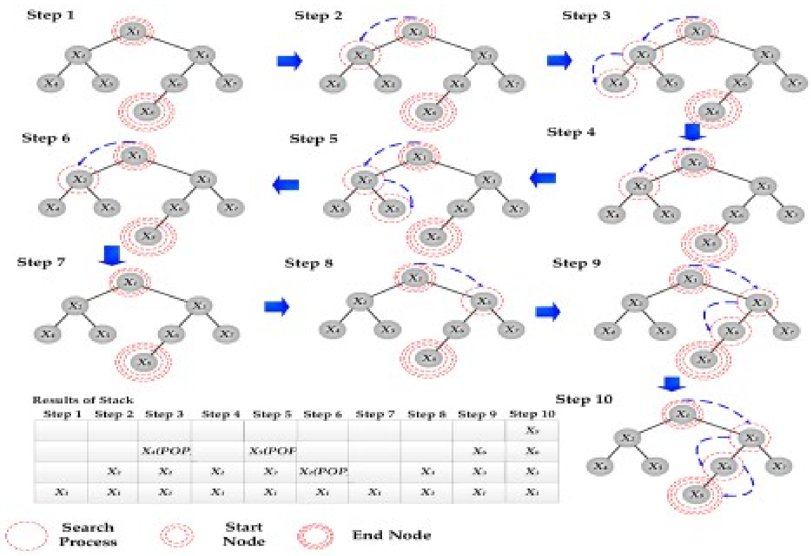
TOURIST ATTRACTION INPUT: The names of tourist attractions from the tourist knowledge ontology were used as the data for Google Maps' API. As shown in the figure above, if there is a relationship between tourist attractions A and B in tourist knowledge ontology, then they are found on Google Maps to obtain the distance between the two points.

HTML PARSER: Google Maps platform has its own HTML tag format. In order to get the accurate distance between two tourist attractions, the HTML parser was used to analyse the HTML structure of the Google Maps' platform.

DISTANCE EXTRACTION: In this step, the HTML tags and web links were deleted, while the distance between tourist attractions was recorded and saved in the tourist knowledge ontology.

5.5 ITINERARY PLANNING

Based on many studies, the main procedure of itinerary planning included tourist attraction matching, tourist attraction reasoning, tour planning, and tourist attraction sorting, as explained below. The tourist attraction X1 is set as the starting point of a tour and X8 is set as the final destination. Using the tourist knowledge ontology, tourist attractions related to X1 and X8 are found. The results of the tour planning are X1, X3, X6, and X8, as shown below:



5.6 KNOWLEDGE BASE DESIGN IMPLEMENTATION: RULES

SEARCH ENGINE: Hotel search option based on pricing and rating requirements

Attractions filter by choice: system recommendation of route and activity by chaining through one spot to the next related spot based on distance and time spent

TRAVEL PLANNER: The main travel planner rule system integrates the functionalities of the search and location-centric recommender subsystems. In order to build a travel planner, we considered crucial constraints such as time, distance and likeability.

MAXIMISING LIKEABILITY SCORE: Where user can apply preferential weightage on categories of tourist spots.

ANALYTICAL OUTCOMES:

- Search for information concerning tourist entities: spots, attractions, and accommodation based on some preferences.
- Search route information between any two spots.
- Provide travel route tailored to visiting user-preferred spots
- Recommend touristic route of spots
- Recommend a location-centric tour for a user-preferred spots
- Generate an attraction-only travel plan
- Generate an Itinerary plan based on number of days of user vacation.

6. PROJECT SOLUTION

This section provides a brief overview of our system solution, with specific details discussed in depth in the system design section. Trip@Us has a simple and intuitive user interface that uses search and optimization techniques to help tourists plan their trip in Singapore. Our key project objective is to automate the entire planning process for our tourists with the project solution for Trip@Us divided into following four essential segments:

6.1 ATTRACTION SELECTION

Upon logging in, our user is asked to provide a preferences for attractions. We then combine these preferences with our specially curated list of attractions. By performing linear optimization, we are able to select a list of attractions that maximizes the likeability score for our tourists. The curated list of attractions are then displayed aesthetically and our users are then led to the next stage of planning: booking the hotel.

6.2 BOOK MY HOTEL

Based on our initial market research, we discovered that tourists tend to prioritize hotel selection by its location, price and rating. Following the selection of our attractions, our users will be prompted to input their price and rating requirements for the hotel. "Book My Hotel" is a smart hotel recommender feature that uses heuristic to traverse the entire search space and then applies machine reasoning to find the ideal hotel for our users. After all, booking the hotel at the right location is a prerequisite for an optimised itinerary.

6.3 PLAN MY TRIP

Our core hybrid reasoning engine churns out thousands of combinations of visiting the selected attractions and envelopes a greedy best first search tree around it. Each combination is further evaluated in its distinct permutations using the technique of genetic algorithm to determine the best possible route for travelling through each attractions. Due to the computational intensity of the task, our backend engine runs asynchronously and our users will be provided with a specially crafted itinerary by email.

6.4 DYNAMIC PLANNING

Dynamic planning is the essential to ensure that our itinerary is flexible so as to adapt to the changing needs of our users. Our dynamic planning feature is built upon a telegram bot interface that provides a friendly human touch to our travellers. Using pre-existing commands, users can dynamically request to add, replace or delete new attractions and our planner optimises based on previously logged user preferences.

7. SYSTEM DESIGN

7.1 KEY DESIGN CONSIDERATIONS

The system design provides an in-depth analysis on the problem formulation, variable assignment and results analysis for each of the above discussed features. We have broadly classified our end users into these 3 groups of travellers, as shown below:

1. Solo travellers, embarking on a journey to explore our Garden city
2. Family, looking forward to a getaway from school or work
3. Friends, who eagerly anticipate a catch-up vacation.

Secondly, our database of tourist hotspots/must-visit attractions in Singapore is categorised into the following types:

1. Cultural & architectural attractions consisting of popular monuments, statues, museums and places of significant historical importance as well as modern day architectural marvels
2. Natural places to rest and relax after a long day of travelling comprising of Gardens by the Bay, Botanical Garden and several renowned parks
3. Shopping attractions to explore Singapore's busiest Orchard district and heartland malls
4. Sightseeing attractions including Esplanade, Night Safari, S.E.A Aquarium, For Canning, Chinatown etc
5. Outdoor attractions including the popular beaches in Singapore and also include trips to our sister islands.
6. Fun-Things-To-Do attractions including adventurous activities such as Universal Studio Singapore, Adventure cove, Wild Wild Wet as well as several theme parks

7.2 ATTRACTION SELECTION

The attraction selection feature in our application helps to match user-specified preferences for certain categories of sightseeing attractions with actual attractions using linear programming fundamentals. These fundamentals maximise the "Likeability" score to help our travellers make full use of their time.

Each attraction in our database is neatly categorised. The categorisation of attractions was done while we curated our attraction database. With our strong knowledge modelling and data pre-processing, a likeability score is generated for each attraction through web crawling of various travel related websites and blogs such as TripAdvisor, traveloka, Klook. The likeability score represents the popularity of each attraction amongst our groups of travellers. From our data pre-processing, we also pulled the minimum number of hours that travellers will need to spend at particular attractions to enjoy it fully.

Once the database is ready, a subset of Linear Programming called Binary Integer programming is formulated as shown below:

1. **Objective Function:** The objective function of our binary integer programming is the overall Likeability score which is obtained on the basis of the below mathematical formula. The preference to each individual locations is obtained from the user himself based on preference for the categories of location.

$$max(objective) \; = \sum_{i=0}^{k} L_i \; \times P_i;$$

where L_i *=Likeability score for attraction i*

P_i *=Preference score for category of attraction i*

k refers to the total number of attractions present in our database

2. **Decision Variables:** The list of attractions represents our binary decision variables (1 representing that the location should be visited whilst 0 meant that the location should not be visited). Therefore, the decision variable forms a matrix of k rows representing k attractions and 1 column.

3. **Constraints:**
The first constraint relates to the **binary nature** of decision variables.

$$x \; = \; Binary$$

The second constraint is the **time constraint**. With the duration of each trip known and an average of 10 hours per day of sightseeing, we are able to compute the total time our travellers will spend doing sightseeing. The minimum time required per attraction is obtained from our database. Thus, the optimised decision variable times minimum hours spent at each attraction must be less than the total sightseeing duration of the trip

$$\sum_{i}^{k} x_i \times T_i \leq D;$$

where x_i *is the binary decision variable for attraction i*

T_i *is the minimum time to be spent at attraction i*

D is the total duration of trip computed in hours

It is interesting to note that although time spent per day on sightseeing is a hard constraint in attraction selection, this parameter transforms into a soft constraint during dynamic planning of the trip.

With the formulation of binary integer programming completed, the decision variables are computed and the list of attractions that our travellers love will be displayed to them.

7.3 BOOK MY HOTEL FEATURE

With the attractions selected by our binary integer programming model, it is imperative to optimise the hotel location so as to achieve the overall objective of minimizing the travel time for our tourists. Minimally, optimisation of the hotel location will involve minimization of total distance to each of the attraction sites.

For our hotel recommender system, we are using a brute force search algorithm that traverses the search space by computing the actual (road) distance to each of the sightseeing locations. Firstly, a rectangular search space is encompassed around all the selected sightseeing locations. The search space is computed based on obtaining geopositional coordinates of each attraction and then selecting the highest and smallest latitude and longitudinal coordinates to be the vertices. This ensures that the search space covers all the selected attractions. The search space is then further dissected into smaller nodes and the distance from the centre of each node to each attraction is computed. This formulates into multiple single depth search trees for each node. Mathematically, the model can be formulated as below:

Minimise (D) where D = $\sum_{i=0}^{k} D_{ai}$;

where D_{ai} *the distance from hotel a to attraction i*

k is the total number of selected attractions

Consequently, the node with the least overall node score in terms of distance is the optimised node and hotels nearby meeting both price and rating requirements of our tourists are recommended. The figure below provides a representation of the algorithm at work.

7.4 PLAN MY TRIP FEATURE

Having optimized the hotel location, the final part of the planning process is to curate our customised itinerary. The objective of the itinerary is to minimise travelling distance so that our users can spend more time at their favourite sightseeing attractions. Trip@Us has a unique solution to this imposing challenge.

Our core engine features a hybrid reasoning system that combines Greedy Best First Search (G-BFS) with Genetic Algorithm (GA). With the number of attractions to visit, duration of each attraction and total trip duration known, it becomes possible to analyse combinations of traversing a search state. A G-BFS tree can then be formulated from the analysis of various combinations of traversing the search space. The mathematical formulation of the search algorithm is shown below. Each state is an attraction with a binary representation. States of 0 imply that an attraction is not visited whilst states of 1 represent that the state is visited. Depth of the search tree represents the days of travel.

Start State = [0,0,0,0.....,0]1xm;

where 0 represents a state is not visited and m represents total number of attractions

Goal State = [1,1,1,1.....,1]1xm;

where 1 represents a state is visited and m total number of attractions

Goal Test = If CumState == Goal State:
success
else:
Fail;

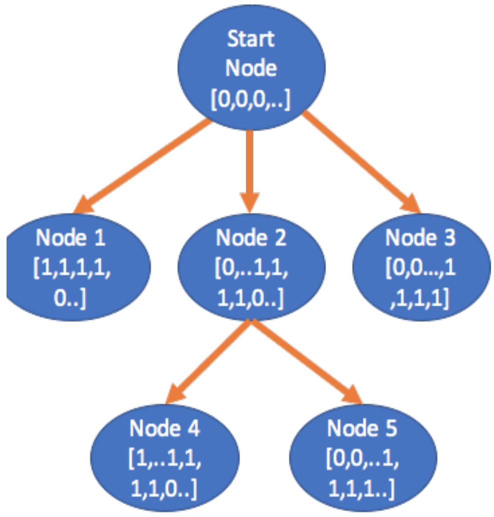
where CumState are elements in the closed list representing a sum addition of all states visited from the start node

Successor Functions = Roads linking to each attraction

Greedy BFS search tree is charted by the various combinations in which tourists can visit these places. As previously mentioned, the depth of the G-BFS tree represents the number of days of travel and hence the number of locations to visit per day is ceiled to the total attractions to visit divided by the duration of the trip.

Locations to visit per day = ceil(Total Attractions/Duration of the trip)

The below table shows G-BFS in action. In the below example, there are 12 selected attractions during a 3 day trip duration implying visiting 4 attractions per day. With the start node defined, the number of combinations analysed on the first day are 495 from choosing 4 attractions to visit from a total of 12 attractions.

Locations: 12 Locations/day: 4 Day 1: 12C4 → 495 places Day 2: 8C4 → 70 places Day 3: 4C4 → 1 place Problem Definition Start State: [0,0,...0] 0 → not visited 1 → visited Goal State: [1,1...1] Successor Function: roads	

With the large number of combinations analysed per day, it is evident that the task is computationally intensive. In order to traverse the path of least cost, GA is employed to compute the fitness function for each combination. For each combination or node, GA is used to permute the different chronological paths to visit the attractions as selected per the combination. The start and stop location for each day is the location of the hotel. Problem formulation of GA is as follows:

Chromosome Definition: Permutations of paths taken to visit selected attractions

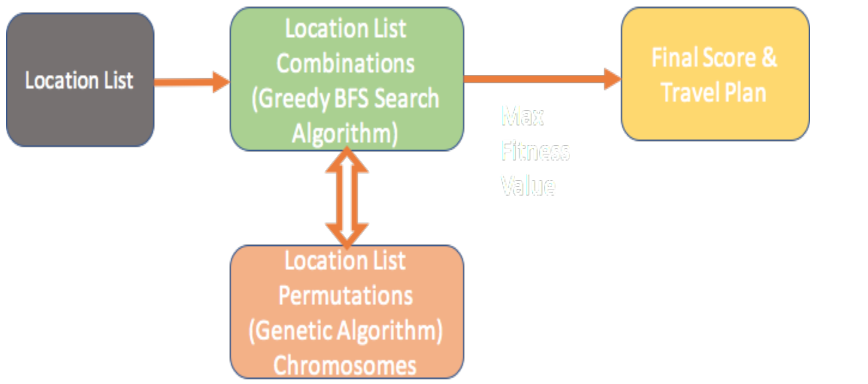
Fitness Function: Max(1/Route Distance)

For each run of GA, we evaluated 300 generations, with the elite size formulated as half of the population size and a mutation rate of 0.01. The maximised fitness function is relayed for each combination and the G-BFS search tree finds the path of least cost (i.e. shortest distance or maximised fitness value).

The below table shows the structure of the hybrid reasoning engine whereby each combination is fed to GA for evaluation of the shortest path to be taken. This path of least cost is then traversed in the G-BFS search tree until the cumulative states traversed equate to the goal node.

Genetic Algorithm (Permutations)

Day 1 to Day 3: 4! (24 ways to travel) Fitness function = 1/total distance



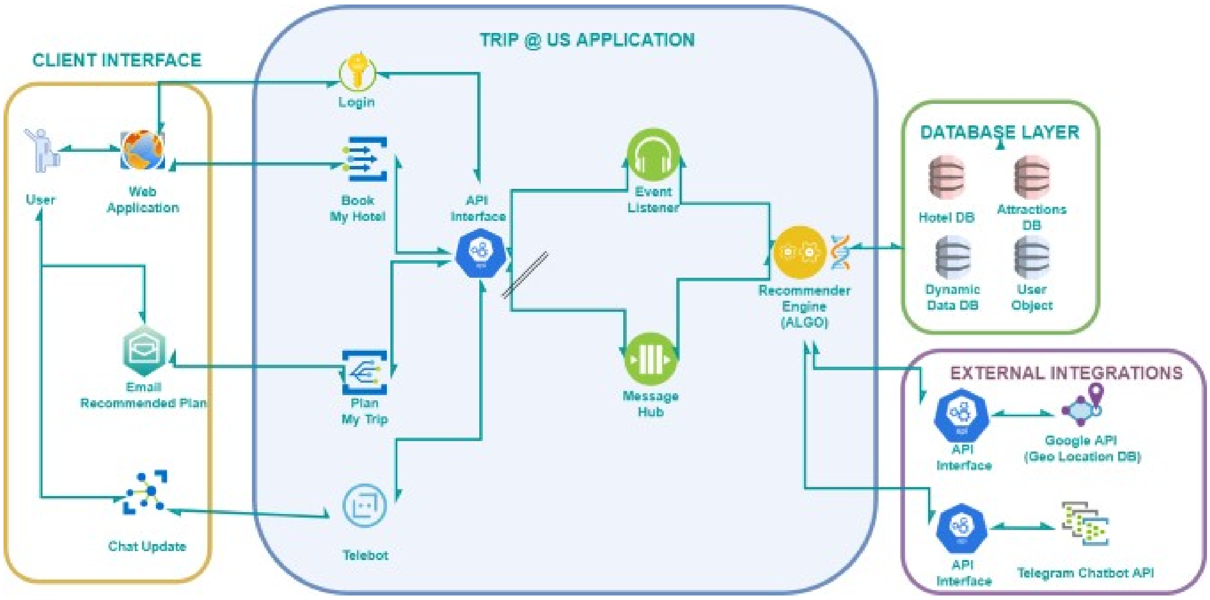
7.5 DYNAMIC PLANNER

In reality, plans will change. Our dynamic planner features our cloud-hosted telebot called "travelBot". Our travelBot is packed with many features that assists our users on the go from extracting your daily itinerary to adding new attractions near our travellers to even interactive entertainment. This ensures that we are always there for our travellers when they need us.

In order to strictly adhere to Personal Data Protection Act of Singapore, we do not actively track our tourist. However, we do request their permission to provide us their location. Once they approve our request and share us their location, we are able to dynamically plan their trip as per their preferences. Our team also believes that by doing so, our users are aware that we are not encroaching their privacy during their travel. This allows us to build a strong user trust with our tourists.

8. SYSTEM ARCHITECTURE

8.1 SYSTEM ARCHITECTURE & FLOW MODEL



Trip@Us application follows a three-tier architecture.

8.1.1 CLIENT INTERFACE LAYER

Client Interface has three adopters for users to interact with the application.

1. **Web Application:** A web based user interface to access the application through the web browser.
2. **Email Recommender:** An simple mail transfer protocol (SMTP) gateway to communicate back to the user on the travel itinerary. This feature was designed with the aim of providing the user with a hardcopy print-out as back-up in situations where they might have internet connectivity issues.
3. **Chat Update Bot:** A chatbot based on Telegram API, for our user to dynamically interact with the application, and **dynamically** plan / re plan his daily travel plans in real-time.

8.1.2 TRIP@US CORE APPLICATION LAYER

The application Layer has multiple major components as shown below:

1. **Message Hub:** A message service bus, to ensure uniform format & interfaces with the application core programs.
2. **Event Listener:** An event listener to trigger appropriate action / program logic based on the user action.

3. **Recommender Engine:** The multi-threaded parallel processing engine hosts all the algorithms; it is simply the brain of the application.
4. **API Interface:** An API interface to interact with the core Recommender Engine as well interact with external Open Source Google API & Telegram Chatbot API Programs
5. **Database Access Object (DAO):** An abstract layer to query/update/interface with the database objects.

8.1.3 DATABASE LAYER (WITH SQL LITE & MONGO DB)

The Database Access Layer consists of two databases for permanent persistence and transient storage.

1. **SQL Lite :** To store user credentials from Django forms.
2. **Mongo DB :** A simple and flexible database highly integrable with front-end architecture to store the attraction, hotel list, and dynamic plan details.

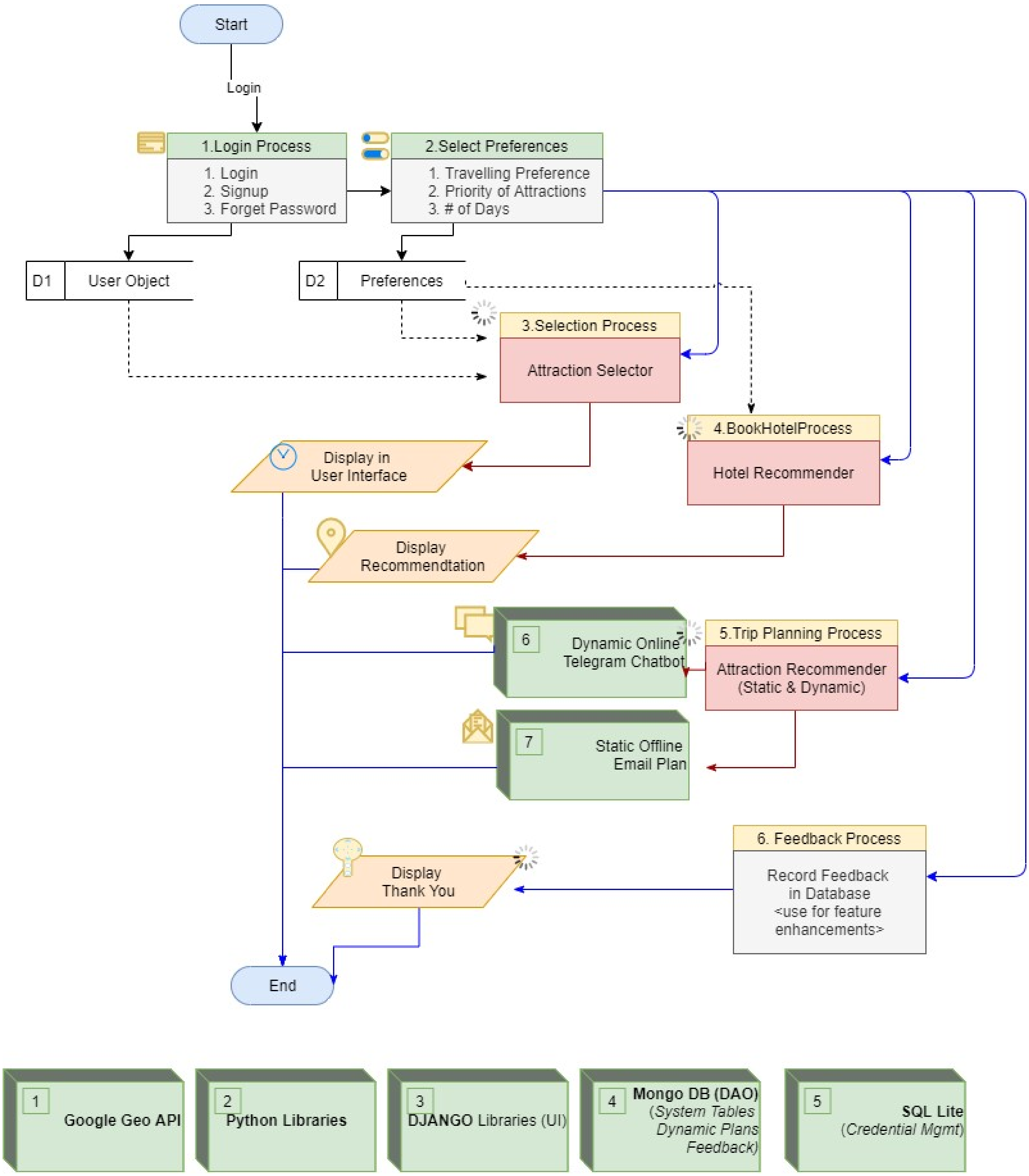
The system architecture diagram illustrates how the application in the front-end interfaces with our internal core backend engine as well external API's such as Google MAP APIs, Telegram Open API.

Our application data flow diagram is divided into 4 parts. First of which is the client interface for our tourists to interact with the system. Our Trip@Us application interface consists of the API backend with an event listener and a message broker for triggering our core backend services. The backend services are connected to our curated database system for hotel and sightseeing attractions in Singapore as well as to the google API to obtain the geopositional coordinates.

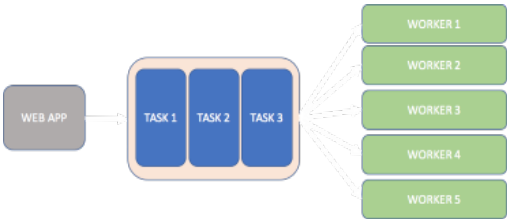
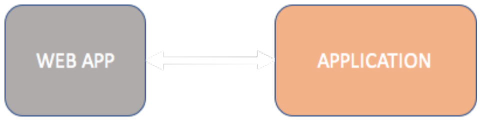
8.2 DATA FLOW MODEL

With the system architecture focusing on the connection amongst our application components, our data flow model focuses on the specific information flow within our application. User authentication is important for us to associate our request to a particular user as well as in the building of future machine learning pipelines. Once our users login, they fill up a simple form indicating their travelling group, preferences as well as the duration of the trip. The travel selection process optimises the attractions and communicates the selected attraction via our web interface. Our "Book My Hotel" feature also requires minimum rating and maximum price per night our users are willing to pay. All of these user preferences are saved in our backend MongoDB.

Data Flow Diagram

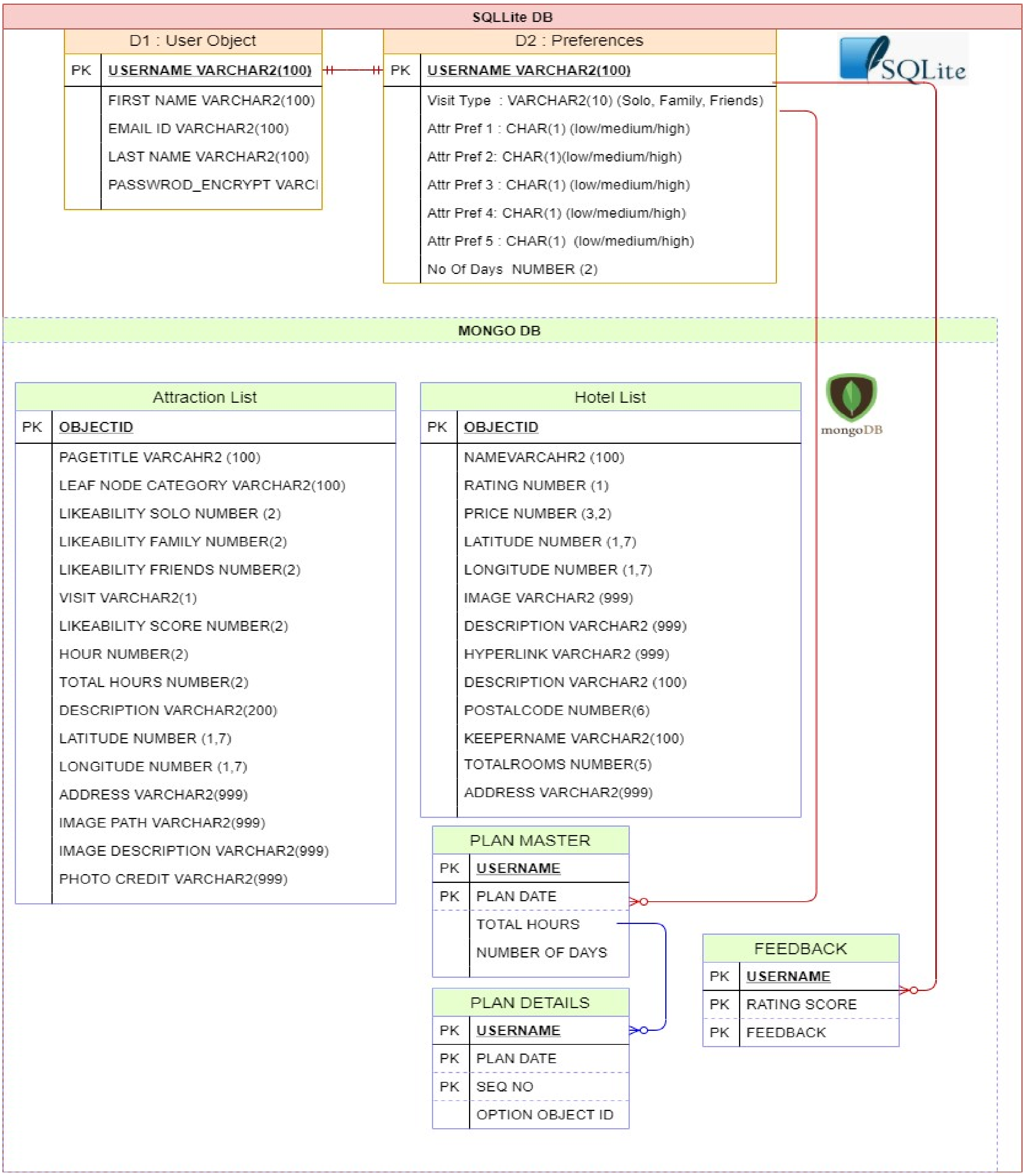


Due to the computational intensity of Plan my trip feature, asynchronous task processing with RabbitMQ message brokerage service and Celery workers is used to execute requests in parallel thus making our application scalable.

Asynchronous Task Processing - Plan My Trip	Synchronous Task Processing - Book My Hotel
	

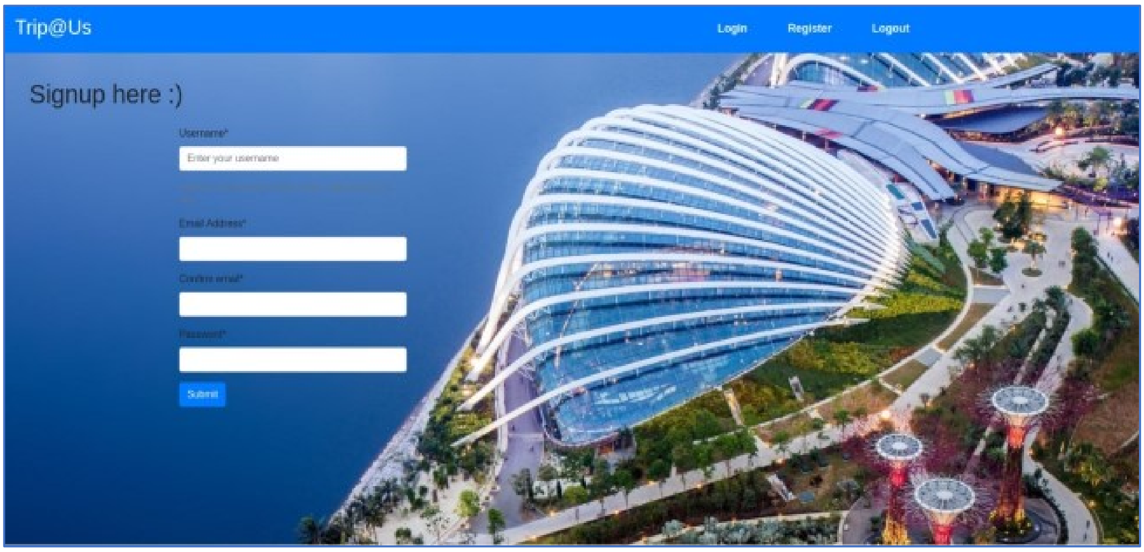
9.2.3 DATABASE DESIGN

The following data model has been used in the application and represents the entity relationship diagram.

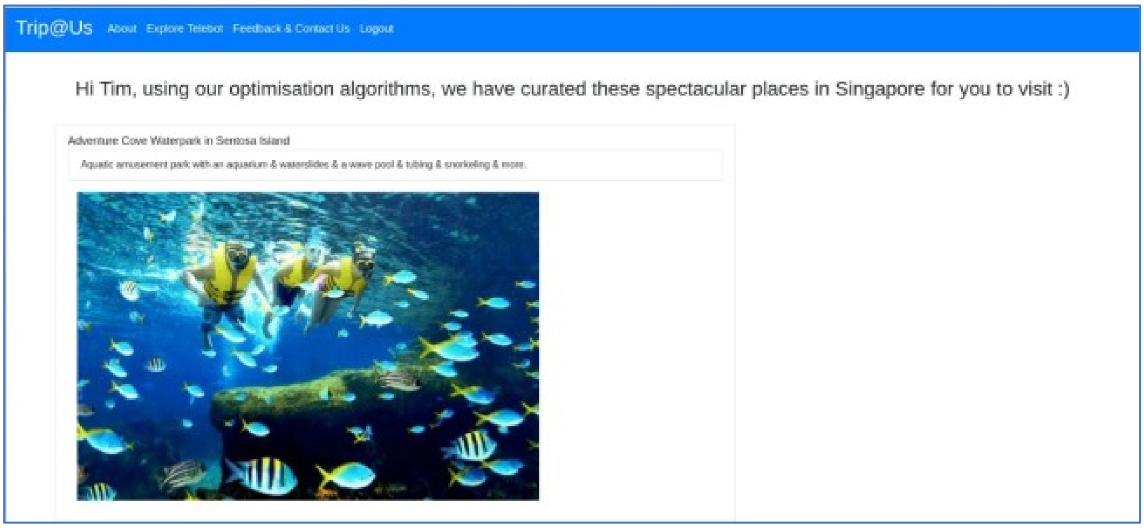


9. APPLICATION SAMPLE OUTPUT

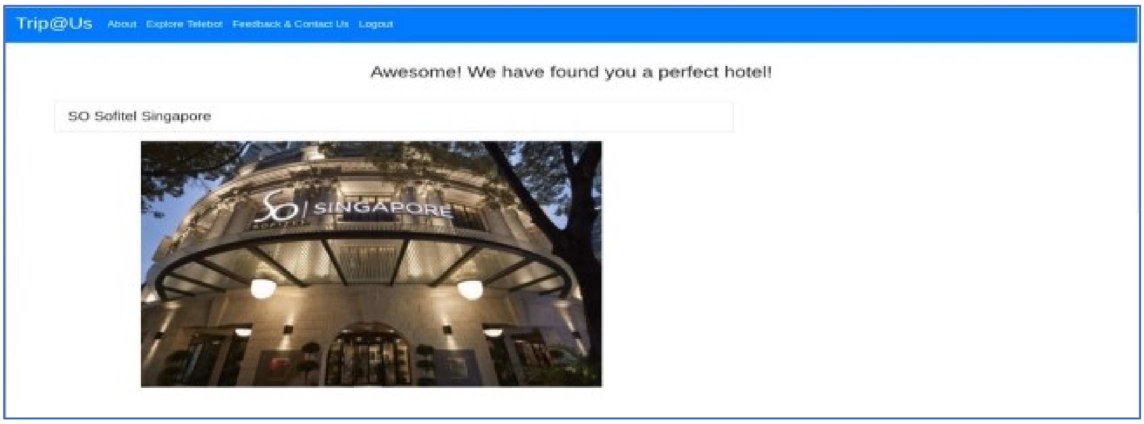
9.1 SCREENSHOT OF USER LOGIN / SIGN UP



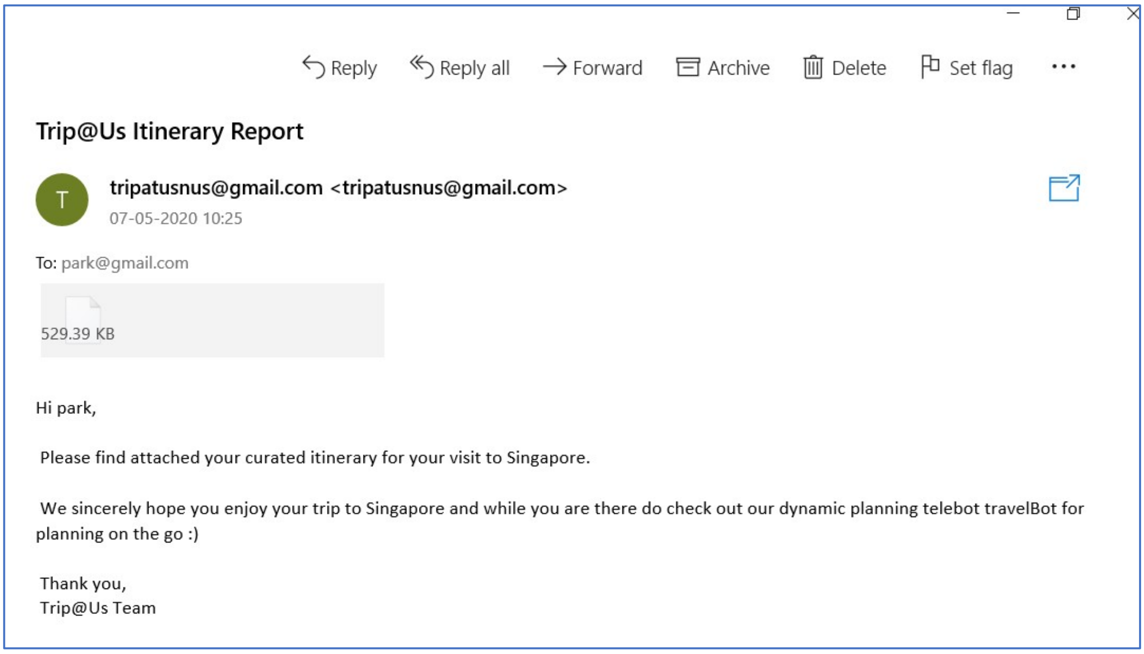
9.2 SCREENSHOT OF ATTRACTION LIST



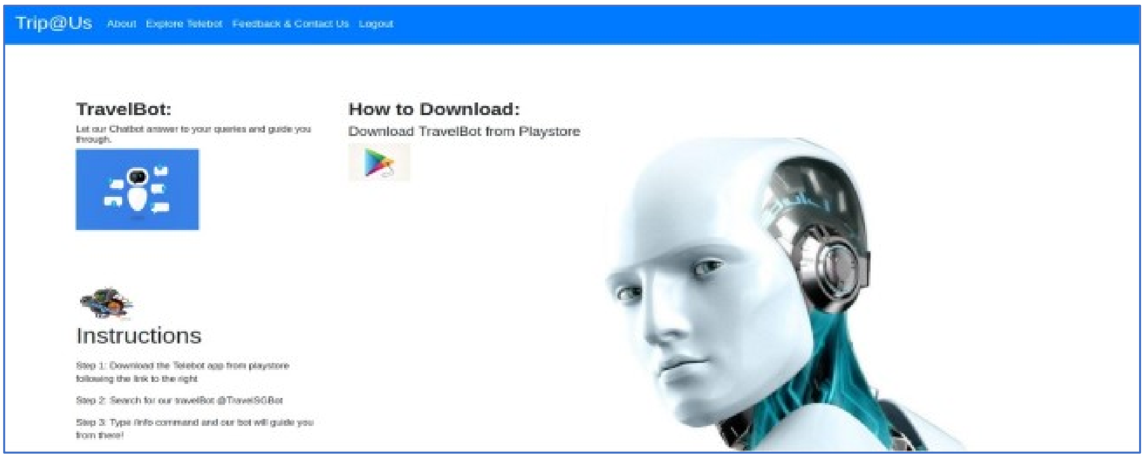
9.3 SCREENSHOT OF HOTEL LIST

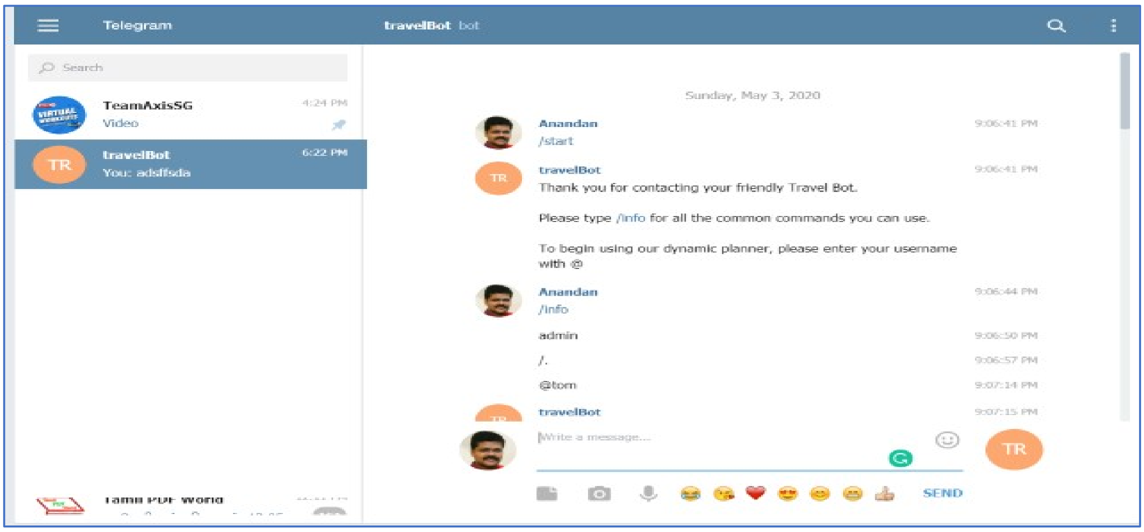


9.4 EMAIL OUTPUT ON ITENARY RECOMMENDATION



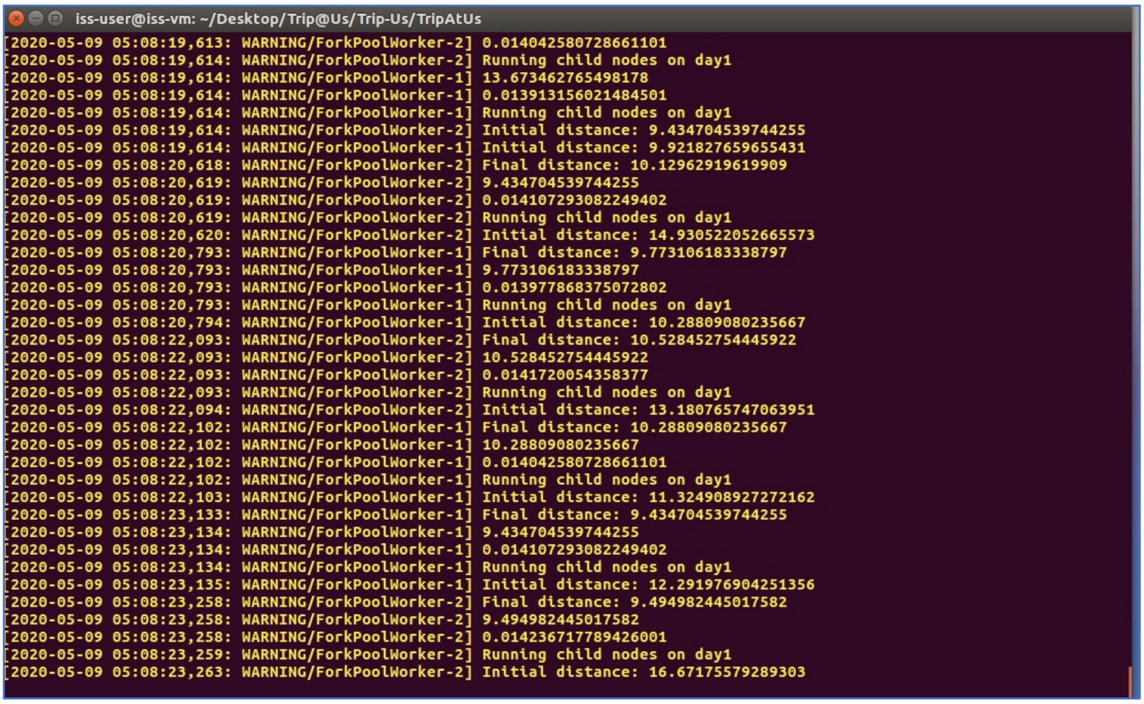
9.5 TRAVELBOT (CHATBOT) OUTPUT





9.6 FEEDBACK FORM

10. CELERY WORKER RUNTIME VERBOSE


iss-user@iss-vm: ~/Desktop/Trip@Us/Trip-Us/TripAtUs

```
[2020-05-09 05:08:19,613: WARNING/ForkPoolWorker-2] 0.014042580728661101
[2020-05-09 05:08:19,614: WARNING/ForkPoolWorker-2] Running child nodes on day1
[2020-05-09 05:08:19,614: WARNING/ForkPoolWorker-1] 13.673462765498178
[2020-05-09 05:08:19,614: WARNING/ForkPoolWorker-1] 0.013913156021484501
[2020-05-09 05:08:19,614: WARNING/ForkPoolWorker-1] Running child nodes on day1
[2020-05-09 05:08:19,614: WARNING/ForkPoolWorker-2] Initial distance: 9.434704539744255
[2020-05-09 05:08:19,614: WARNING/ForkPoolWorker-1] Initial distance: 9.921827659655431
[2020-05-09 05:08:20,618: WARNING/ForkPoolWorker-2] Final distance: 10.12962919619909
[2020-05-09 05:08:20,619: WARNING/ForkPoolWorker-2] 9.434704539744255
[2020-05-09 05:08:20,619: WARNING/ForkPoolWorker-2] 0.014107293082249402
[2020-05-09 05:08:20,619: WARNING/ForkPoolWorker-2] Running child nodes on day1
[2020-05-09 05:08:20,620: WARNING/ForkPoolWorker-2] Initial distance: 14.93052205266573
[2020-05-09 05:08:20,793: WARNING/ForkPoolWorker-1] Final distance: 9.773106183338797
[2020-05-09 05:08:20,793: WARNING/ForkPoolWorker-1] 9.773106183338797
[2020-05-09 05:08:20,793: WARNING/ForkPoolWorker-1] 0.013977868375072802
[2020-05-09 05:08:20,793: WARNING/ForkPoolWorker-1] Running child nodes on day1
[2020-05-09 05:08:20,794: WARNING/ForkPoolWorker-1] Initial distance: 10.28809080235667
[2020-05-09 05:08:22,093: WARNING/ForkPoolWorker-2] Final distance: 10.528452754445922
[2020-05-09 05:08:22,093: WARNING/ForkPoolWorker-2] 10.528452754445922
[2020-05-09 05:08:22,093: WARNING/ForkPoolWorker-2] 0.0141720054358377
[2020-05-09 05:08:22,093: WARNING/ForkPoolWorker-2] Running child nodes on day1
[2020-05-09 05:08:22,094: WARNING/ForkPoolWorker-2] Initial distance: 13.180765747063951
[2020-05-09 05:08:22,102: WARNING/ForkPoolWorker-1] Final distance: 10.28809080235667
[2020-05-09 05:08:22,102: WARNING/ForkPoolWorker-1] 10.28809080235667
[2020-05-09 05:08:22,102: WARNING/ForkPoolWorker-1] 0.014042580728661101
[2020-05-09 05:08:22,102: WARNING/ForkPoolWorker-1] Running child nodes on day1
[2020-05-09 05:08:22,103: WARNING/ForkPoolWorker-1] Initial distance: 11.324908927272162
[2020-05-09 05:08:23,133: WARNING/ForkPoolWorker-1] Final distance: 9.434704539744255
[2020-05-09 05:08:23,134: WARNING/ForkPoolWorker-1] 9.434704539744255
[2020-05-09 05:08:23,134: WARNING/ForkPoolWorker-1] 0.014107293082249402
[2020-05-09 05:08:23,134: WARNING/ForkPoolWorker-1] Running child nodes on day1
[2020-05-09 05:08:23,135: WARNING/ForkPoolWorker-1] Initial distance: 12.291976904251356
[2020-05-09 05:08:23,258: WARNING/ForkPoolWorker-2] Final distance: 9.494982445017582
[2020-05-09 05:08:23,258: WARNING/ForkPoolWorker-2] 9.494982445017582
[2020-05-09 05:08:23,258: WARNING/ForkPoolWorker-2] 0.014236717789426001
[2020-05-09 05:08:23,259: WARNING/ForkPoolWorker-2] Running child nodes on day1
[2020-05-09 05:08:23,263: WARNING/ForkPoolWorker-2] Initial distance: 16.67175579289303
```

10. IMPROVEMENTS AND OPPORTUNITY

From its inception, our Trip@Us application model is designed to be scalable in terms of both accommodating more user traffic flow through its parallel multi-thread processing system architecture as well as geographical scalability through our strong data processing pipelines. There is a very strong opportunity to expand our systems geographically with reduced work as our current data processing pipelines can be employed to extract the relevant information. Our application also focuses on continual improvements through our user feedback system.

Currently there are a number of improvement opportunities we are working on. At our core backend engine, we would like to reduce the combinations analysed to satisfy a mix-id so as to ensure that our users do not travel to similar categories of attraction. Although we may not achieve global optimum, this will definitely prevent boredom and keeps our tourists vivacious. Furthermore, we are testing and actively working on reducing the time and space complexity in the use of A* search to traverse the attraction states. Combining a mix id with A* search will allow us to reduce the complexity of the A* search.

Another potential enhancement we are actively working on is the use of google dialog flow with our humanoid TravelBot. This will allow our users to verbally communicate with our systems and even resolve the hassle of typing. With all of these enhancements already in the pipeline for future upgrades, we are confident that Trip@Us has the right target market insight, scalable design and structure and most importantly the right team.

11. CONCLUSION

To conclude, Trip@Us started with addressing a niche target market of tourists who prefer to combine the flexibility of planning their own trip yet require active recommendations on places to visit. Leveraging on our sound conceptual knowledge in machine reasoning, reasoning systems as well as cognitive systems, we have devised an intelligent trip planning tool that guides you in every stage of your journey from overall planning involving selecting attraction, booking hotel and planning itinerary to trip execution with our dynamic planner. The 12 hrs or more of painful planning is reduced to less than 30 mins of smart recommendation systems with further benefits derived from reduced travelling time. With this I would like to wish all Happy Tripping!

12. KEY LEARNING OBJECTIVES MET

The section of the report maps system functionalities against knowledge, technique and skills learnt in the modular courses: Machine Reasoning, Reasoning Systems and Cognitive Systems. It also contains the installation and user guide, individual reflection of the project journey and the references.

12.1 MAPPING SYSTEM FUNCTIONALITIES TO MODULAR COURSES

Trip@Us employs the materials from all of the above modules to build an intelligent travel planning recommender system. The core hybrid reasoning engine built to curate our travel itineraries was developed from scratch and only leveraging on the materials taught. Further development in terms of A* search to curate our itinerary were also considered but the time and space complexity of the solution impeded us from investing further.

Feature	Mapped Functionalities
Attraction Selection	Solves Binary Integer Programming for constraint optimisation, which is a subset of search optimisation algorithms a concept taught in the **Reasoning Systems.**
Book My Hotel Feature	Uses a combination of Brute force search algorithm and **Machine Reasoning** to determine a hotel that optimises the travelling time for our tourist and meets our tourists pricing and rating requirements.
Plan My Trip Feature	Core hybrid reasoning engine employs the Greedy Best First Search algorithm to traverse the search tree and the genetic algorithm to chart out the best route minimizing the time spent travelling. Both these algorithms were studied in detail during the **Reasoning System** Module.
Intelligent travelBot Telebot	Our telebot is an active humanoid agent that listens to the intents and commands from our tourists and uses **Cognitive Reasoning** to provide the optimal solution.

12.2 INDIVIDUAL REFLECTION

This section is team member's individual reflection on the project journey and consist of personal contribution as well.

12.2.1 VIDISH MEHTA – INDIVIDUAL REFLECTION

Through the meandering tides and stormy thunderstorms, we voyaged through a wonderful journey in developing Trip@Us. To me, personally, Trip@Us is my first major project involving a full stack scalable intelligent system outside of work. Needless to say, the journey did get frustrating sometimes with beyond mid nights debugging and researching alternative courses of action, but when I look back I come to realise how much I have learnt in this beautiful journey.

Initially when our team started the project, we were aiming to employ search optimisation algorithms for the development of an intelligent home renting application. However, we soon realised that projects may fail on sustainability and scalability due to the dynamic nature of the data and the amount of time needed to pre-process each data. Thus, we decided to pivot to the tourism industry which provided the advantage of a more static hotel and attraction database. Trip@Us is also quite different from some of its predecessors in that it offers a complete package of planning our tourists' trips from attraction selection to optimising the hotel location and finally to planning the entire itinerary. Furthermore, it's a first of a kind application to enable dynamic planning with a fully functioning telebot and it completes all this while taking less than 15s of user's time in filling its simple form.

This project strikes a special chord in me as personally I dislike planning for trips. Although the excitement of travelling keeps me engaged, the whole process of planning eventually becomes too draining. In many of my trips, I have also realised that we do not stick to our core itinerary as preferences and schedule change. I thoroughly enjoyed the process of developing Trip@Us as solving each small problem along the way gave me immense satisfaction. Our team's strong perseverance and fighting spirit led us to completion of a project that we are most certainly very proud of. I would like to take this opportunity to thank our professors for their continuous guidance during the execution of our project.

- ***Personal Contribution***

I have been thoroughly involved from the start of the initial project proposal and submission to the mid project presentation and final project report and submission. Personally, I have contributed to the development of our core backend programming logic for attraction selection, book my hotel and plan my trip feature. Leveraging on the materials learned and invoking my creativity, I was a key solutionist behind the formulation of the binary linear programming, brute force search in book my hotel and the core hybrid reasoning engine in plan my trip feature.

In developing Trip@Us, I embarked on a continual journey of learning (mainly through reading hundreds of pages of documentation) with the use of PuLP's coin-or-branch and cut (cbc) solver to initialisation and setting up the cloud-based Mongo-DB for saving our specially curated hotel and attraction database. Reading on implementation of A* search and Greedy Best First search motivated me to construct these algorithms from scratch. Due to time and space complexity of the A* search algorithm, we decided to drop A* search and move to Greedy Best First Search. Reading on the core implementation of these algorithms, I was able to understand the in-depth working of the open and closed list, start state, goal state and goal test. I was also able to differentiate between the goal test for greedy best first search and A* search. I have also contributed to the development of our dynamic planning telebot, which is now hosted, active and running in the cloud.

I was also part of the core front-end django web framework development team. I had the opportunity to deep dive into JavaScript's Leaflet module for dynamic map plotting and use ajax query for plotting geojson coordinates dynamically on the map. System scalability can be a major challenge and with high user traffic flow; even simple applications can break down. We knew from the start that it was imperative to use asynchronous task processing for computing our tourists itinerary. I read through countless documentations on best in use message brokerage systems such as redis and rabbitmq to communicate to our server workers. I also hit a roadblock when I realised that djangoq async framework was incompatible with the Windows operating system. Finally, I decided to go ahead with celery worker deployment listening into an active rabbitmq caching and message brokering system. Running on the iss-vm, the system became extremely scalable and able to handle large incoming user traffic.

- ***Application of skills developed***

I am currently working for a major semiconductor company in Singapore in its process control system department. I am frequently involved in various development projects and I firmly believe that the technical knowledge I gained through this project is easily applicable to these projects. For example, handling asynchronous task processing will be very helpful in scaling the applications I have created. Mongo Db could be an essential inclusion in our application as it offers a unique flexibility in data storage and high integrability with front-end architectures.

Through deployment of Trip@Us in our iss-vm, I have also been able to strengthen my knowledge in the ubuntu operating system. I have already started applying principles of search algorithms in some of the automation solutions I have initiated at work. All in all, Trip@Us, being my first large scale full stack application, has provided me with a lot of opportunities to explore the large cloud of AI and has turned me into an effective solutionist.

12.2.2 ANANDAN NATARAJAN – INDIVIDUAL REFLECTION

We are the team of 4 sharp individuals, technologist, entrepreneurs with strong technology background and with the eager to learn and apply new futuristic tech subjects. That worked effectively together. We understood each of our strengths and made sure they were used. One very good aspect was the way we did a detailed study of the requirements of the project, and brainstormed the ideas to find a project that factually can be used at the same time will be a test bed to apply the new learnings we have had during the course. As a team we failed to see the complexity of the problem and took an approach that was too focussed. We found that we concentrated far too much on only defining the search logic, and core logic and missed to see what else we can do to establish the cognitive searches / dynamic replanning features can be extended as a functionality to the modern world. This has come as a learning from our mid-project presentation and the guidelines provided by the professor, which we highly appreciate.

This has demonstrated that a team is more than the sum of its component parts. Crucial to what we did was assessing and recognising our strengths and bringing them together to create something more powerful that any of us individually. We certainly came out as learned individuals with the structured time-bounded implementation run it as a proper project managed delivery. The main learning point is not to rush into the task and spend some time evaluating the issue fully. In retrospect we should have invited others onto the team and also been prepared to take time talking to more of the key players than we did. We also immediately assumed that the problem was far more complex than it was.

This was a successful team and no major changes are needed but each of us can use this experience of a way of enhancing the effectiveness of other teams we work with. We will challenge each other more. In retrospect it was clear that we all knew that things were not progressing as well as they should. We all misinterpreted the fact that we worked well together as re-assurance that we were achieving our objectives. In essence, we all became too comfortable as members of the team – and all now feel disappointed with the outcome. As a team we need to be confident to speak out and challenge.

The learning from this project going to be a stepping stone in my next projects on how much I can upfront prepare myself in taking technology challenges. I am from banking industry, and I could see the intelligent reasoning systems have a bigger role to play in finding fraudulent payment transactions by running through various historic behaviours of the clients. I am hoping to use this opportunity to learn more on that.

14.2.3 SONIC WANG – INDIVIDUAL REFLECTION

This was an interesting project, and everything was fresh to me. Patience and persistence were the crucial and essential factor that was needed in such a project in my view. Fortunately, I learned from my lovely teammates. Besides, communication was another extremely significant factor. It was essential to communicate and discuss with the course mates regularly. During the process, since that everyone is from different background, the way of work is different too which allow me to learn from them.

The project itself contains lots of techniques from frontend, backend, VM, database…etc. For a non-developer like me, at first, I was so lost during the discussion. Also thanks to my teammates, I slowly pick up what are they saying and learn what is the difference between each coding language, platform, and framework. And what I also helped to manage the database. It's such an exciting experience for me because I get to contribute some technical part for our project. I do enjoy collecting data, cleaning database, making category, and making the database useful and tidy.

The project is a very interesting project for me as well. As a frequent traveler, I will visit more than 20 countries per year. Whenever I visit a new city, it will always take me lots of time to research online for the attractions, food, and hotels. It will definitely be my pleasure if we can utilize the project to some other cities as well. Hopefully, in the near future, there won't be too much time spent on planning as I always did.

Lastly, I would like to give thanks to the professors and my teammates. During the first semester, I truly learned a lots from the course and also from the project regarding to the algorithms and the techniques I can potential use in my near future. Hopefully, the COVID19 situation will get better soon so that we can all gather back to classrooms. Furthermore, we can all start traveling around the world again. Wish you guys a great health during this period.

14.2.4 ANKEIT TAKSH – INDIVIDUAL REFLECTION

It was an awesome opportunity to be with an excellent team and do this project. The project did help me enhance my technical skillset as well is something we created keeping in mind the user experience enhancement.

WHAT I DID ? I did understand the core logic of intelligent systems. I enhanced my skills I with the python code and its multiple modules and we did new things like making machine scalable, presentation skills and logical enhancement which will be serving the path of my future. Worked with language processing, Maps and API calls with new algorithm.

WHAT DID I LEARN? I am better at my profession and have some good hands-on with advanced technologies in market. Allowed me to probe the aspects of latest technologies. It definitely enabled me to face the bright future of technologies. One critical thing I learnt is its not just tech knowledge but how to use technical abilities for making life smoother and enable world better with making better solution implementation.

WHAT I CAN DO BETTER? Learnt my capabilities needs to further be honed and hence I need to implement and learn further new technologies. Need to look into every small thing and decide on how to do things better.

WHAT WENT WRONG? We had issues with the timelines and project planning which were haywire and could have been better but perhaps since we were new team I believe we can do a lot better in future.

I am looking forward to give in more dedicated time and be better with upcoming projects in every aspect.

13. INSTALLATION & USER GUIDE

The installation and user guide provide an overview of the steps involved in the deployment of the project. The installation of the system requires downloading of the pre-built virtual machine from ISS-VM. A complete and a comprehensive guide of the step required for the deployment of the project is provided in the video.

15.1 USER GUIDE

The user guide has been stored in the GitHub, under 'User Guide' Folder. Please refer for further details.

https://github.com/vid1994/Trip-Us

Folder: UserGuide

14. RESEARCH & REFERENCES

1) Singapore Hotel Database: https://docs.google.com/spreadsheets/d/1YT8sHUpjMoEwOIZcLg8cu_4AJ_ehhZkZ9HoDIWg5OEE/edit#gid=449494551
2) Singapore Attraction Database: https://docs.google.com/spreadsheets/d/15NCqk8FpFeFcVCNHp2sXj3sEp09u0YkVTgPEUGVpcVU/edit#gid=1857804898
3) Group 11 Presentation Slides: https://docs.google.com/presentation/d/1wWPLE9htIY6tUVy62XUd99RzPl_dje-k7ghP3iyx5JE/edit#slide=id.g835450d152_0_40

[1] S. H. D. Set, “Singapore Hotel Data Set,” [Online].

[2] S. T. A. D. Set, “Set, Singapore Tourist Attraction Data,” [Online].

[3] A. S. Algorithm, “A* Search Algorithm,” [Online].

[4] I. t. A. Intelligence, “Introduction to Artificial Intelligence,” [Online].

[5] B. A. w. Django, “Build Apps with Django,” [Online].

[6] M. T. i. Python, “Multi Threading in Python,” [Online].

[7] O. M. Queries, “Optimize MangoDB Queries,” [Online].

[8] B. T. ChatBot, “Building Telegram ChatBot,” [Online].

[9] S. T. B. 2. Q. Result, “Singapore Tourism Board 2018 Q4 Result,” [Online].

[10] Traveloka, “Traveloka,” [Online].

[11] Agoda, “Agoda,” [Online].