Requirement Specification

Zhixuan Xu 201377035
Hongyi Wu 201376889
Zhuo Wang 201376856
Xiaoyu Qin 201376441
Yifan Hu 201375824
Weiqi Chen 201375504
Department of Computer Science

14 February 2019

Author: Group 15

Monitor: Sebastian Coope

Reviewer: Shagufta Scanlon

Module: COMP208

Lecturer: Michele Zito

Group Member Signature:

吴弘毅 胡亦凡 许志轩 陈伟奇
王卓 秦晓雨

Contents

1 Project Description

The Travel Counsellor System is an online application consists of a database with travel preferences of registered customers, information on traveling destinations and tourist attractions. The purpose of the system is to receive users’ desired destination and recommend a route. It was based on the departure point provided from the user, the users’ preferences, which were stored into the database system when the user choose to register the account, and the actual situation of the tourist attractions, for example, the opening hour of the tourist attractions and fewest bus or subway fares.

The intended audience for the system includes but is not limited to backpackers, travellers who prefer personalized travel routes in a short period time, and inexperienced travellers, for example, university students, who usually prefer more freedom during their tours and are less consistent with their original travel plans.

There are multiple means for the users to acquire their personalized travelling route. The standard recommendation is solely based on the destination and starting point inputted by the user, the system would provide the shortest path available to visit most tourist attractions possible in a day. The preference recommendation is based on the preferences from the users, which is stored in the database by the choices made in the questionnaire when the user had finished in the registration process. The system will recommend three candidate travel destinations all around the world, and provides a suggestion route and days of travel. The advanced recommendation will take both the preference and actual travelling destination into account. The final recommendation is more likely catering to the users’ ultimate requirement of travelling experience. The route recommendation may rely on other navigation application like Google map, with origin and destination provided by the system.

The detailed of mission objectives are as follows, To maintain (enter, update and delete data) on Countries.

To maintain (enter, update and delete data) on Cities.

To maintain (enter, update and delete data) on Tourist attractions.

To maintain (enter, update and delete data) on User account information.

To maintain (enter, update and delete data) on User travel preferences.

To maintain (enter, update and delete data) on Registered users.

To perform searches on countries.

To perform searches on cities.

To perform searches on tourist attractions.

To track the status on tourist attractions.

To report on users activities.

To report on tourist attractions.

To report on countries and cities.

Figure 1 Mission Objectives

2 Statement of Deliverables

2.1 Anticipated Documentation

Apart from the requirement specification documentation, the anticipated documentation should include a preference questionnaire to collect the data of user preference, this the essential for the function of preference recommendation. There is no need to provide a separate user manual since the user interface should be straightforward and the user should query the travelling route under the instruction without difficulties, however, some frequent asked question could be still provided on the website for reference. A feedback survey could also be included during testing period and system demonstration phase.

2.2 Anticipated Software

2.2.1 System Boundary Diagram

There is a user interface, a database, a processing algorithm and an external Map API in the scope of System. Only the administrator can access and maintain these system components. The database stored the information of user preferences and history queries, basic information of travelling destinations and the tourist attractions are labelled referring to user preference questionnaires to filter the destination for future recommendation. The processing algorithms are aimed to provide the ultimate route solution for users. Travelling time and distance, travel costs, and actual situations, for example, the opening hours of the tourist attractions will be taken into account. Exist algorithms like Travelling Salesman algorithm and Dijkstra algorithm might be adopted during the design of the algorithm. The Map API may rely on external navigation application since the lack of experience and resource on location and navigation service of the development team.

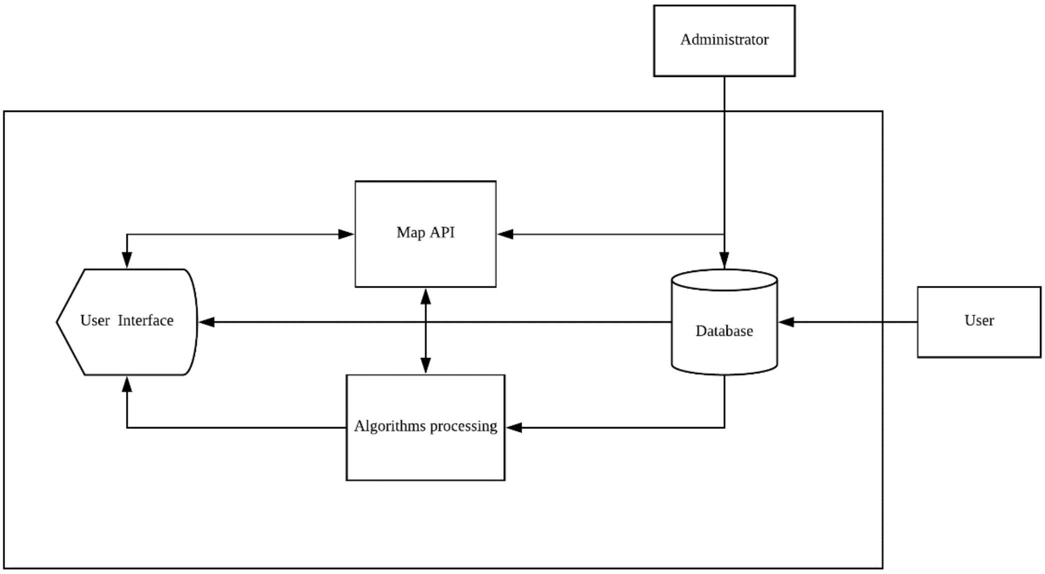

Figure 2 System Boundary Diagram

2.2.2 Use Case Diagram

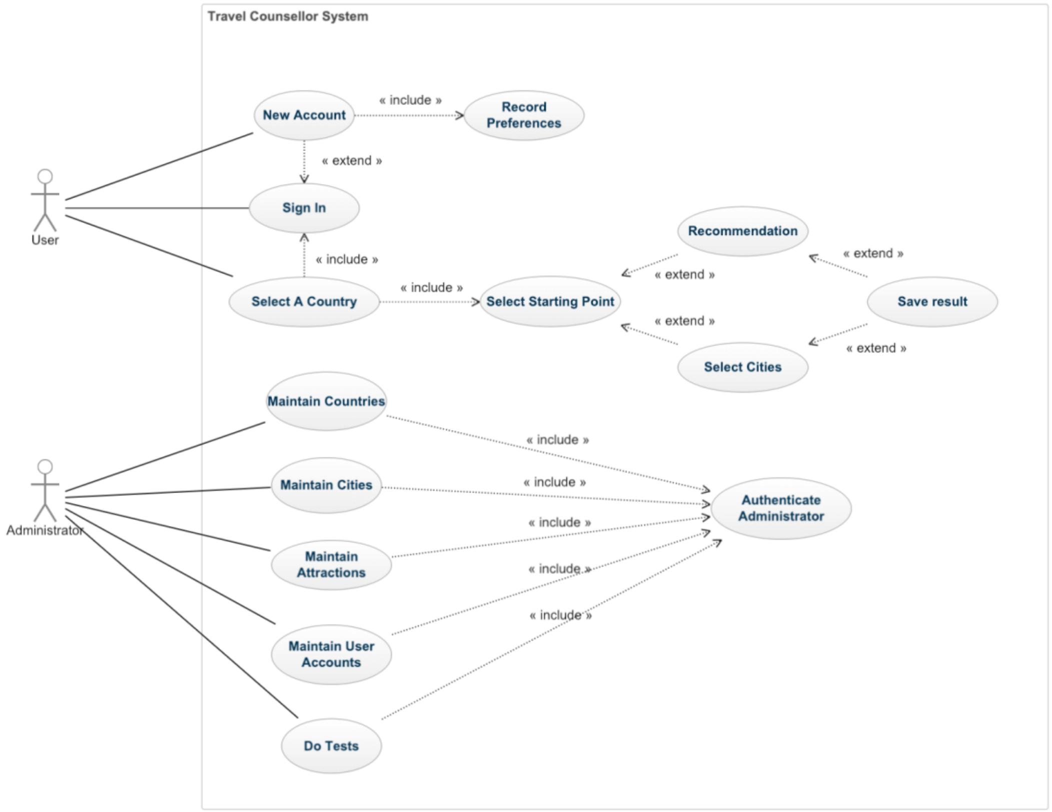

Figure 3 Use Case Diagram

ID	**UC1**
Name	New Account
Description	The user signs up a new account
Pre-condition	1. Travel Counsellor System in service (internet connected) 2. The database is not full
Event flow	The user clicks the Sign-Up button and set username and password
Post-condition	The account created
Includes	Use case 2 “Record Preference”
Extensions	
Triggers	User clicking the Sign-up button

ID	**UC2**
Name	Record Preference
Description	The user chooses their preferences
Pre-condition	1. Travel Counsellor System in

	service (internet connected) 2. The database is not full 3. A new account is created just now
Event flow	The user chooses their preference for travelling
Post-condition	The account preference recorded
Includes	
Extensions	
Triggers	User clicking the personalized preference button

ID	**UC3**
Name	Sign In
Description	The user signs in the system
Pre-condition	Travel Counsellor System in service (internet connected)
Event flow	The user signs in the system with username and password
Post-condition	Log in successfully or wrong password
Includes	
Extensions	User case 2 “Record Preference”
Triggers	User clicking the login button

ID	**UC4**
Name	Select A Country
Description	The user chooses a country to start
Pre-condition	Travel Counsellor System in service (internet connected)
Event flow	1. The user signs in successfully 2. The user chooses a country to travel
Post-condition	Country selected
Includes	User case 3 “Sign In”
Extensions	
Triggers	User clicking the countries button

ID	UC5
Name	Select Starting Point
Description	The user chooses a point to start
Pre-condition	The user chooses a country to travel
Event flow	1. The user signs in successfully 2. The user chooses a country to travel 3. The user chooses a point to start
Post-condition	Starting point selected
Includes	
Extensions	User case 6 “Recommendation” User case 7 “Select Cities”
Triggers	User clicking the starting point

ID	UC6
Name	Select Starting Point
Description	The user chooses a point to start
Pre-condition	The user chooses a country to travel
Event flow	1. The user signs in successfully 2. The user chooses a country to travel 3. The user chooses a point to start
Post-condition	Starting point selected
Includes	
Extensions	User case 6 “Recommendation” User case 7 “Select Cities”
Triggers	User clicking the starting point

ID	UC7
Name	Recommendation
Description	The user chooses the recommended cities from the system
Pre-condition	The user chooses a point to start
Event flow	1. The user signs in successfully

	2. The user chooses a country to travel 3. The user chooses a point to start 4. The user chooses the recommended cities from the system
Post-condition	The cities selected and the route presented
Includes	
Extensions	
Triggers	User clicking the recommendation button

ID	**UC8**
Name	Select Cities
Description	The user chooses one’s preferred cities from the system
Pre-condition	The user chooses a point to start
Event flow	5. The user signs in successfully 6. The user chooses a country to travel 7. The user chooses a point to start 8. The user chooses one’s preferred cities from the system
Post-condition	The cities selected and the route presented
Includes	
Extensions	
Triggers	User clicking the cities and Done button

ID	**UC9**
Name	Maintain Countries
Description	The administrator maintains (enter, update, and delete) data from countries

Pre-condition	Travel Counsellor System in service (internet connected)
Event flow	1. Include Use Case 14 “Authenticate Administrator” 2. Maintain countries
Post-condition	
Includes	Use Case 14 “Authenticate Administrator”
Extensions	
Triggers	Maintain Countries requested

ID	**UC10**
Name	Maintain Cities
Description	The administrator maintains (enter, update, and delete) data from cities
Pre-condition	Travel Counsellor System in service (internet connected)
Event flow	3. Include Use Case 14 “Authenticate Administrator” 4. Maintain cities
Post-condition	
Includes	Use Case 14 “Authenticate Administrator”
Extensions	
Triggers	Maintain cities requested

ID	**UC11**
Name	Maintain Attractions
Description	The administrator maintains (enter, update, and delete) data from tourist attractions
Pre-condition	Travel Counsellor System in service (internet connected)
Event flow	5. Include Use Case 14 “Authenticate Administrator” 6. Maintain attractions
Post-condition	

Includes	Use Case 14 “Authenticate Administrator”
Extensions	
Triggers	Maintain attractions requested

ID	UC12
Name	Maintain User Account
Description	The administrator maintains (enter, update, and delete) data from user accounts
Pre-condition	Travel Counsellor System in service (internet connected)
Event flow	7. Include Use Case 14 “Authenticate Administrator” 8. Maintain user account
Post-condition	
Includes	Use Case 14 “Authenticate Administrator”
Extensions	
Triggers	Maintain user account requested

ID	UC13
Name	Do Test
Description	The administrator does series of test
Pre-condition	Travel Counsellor System in service (internet connected)
Event flow	9. Include Use Case 14 “Authenticate Administrator” 10.Do test
Post-condition	
Includes	Use Case 14 “Authenticate Administrator”
Extensions	
Triggers	Test requested

ID	UC14

Name	Authenticate Administrator
Description	The administrator proves the identity to the system
Pre-condition	Travel Counsellor System in service (internet connected)
Event flow	The administrator enters a username and passcode and be authenticated
Post-condition	Permitted to the system if the identity is administrator
Includes	
Extensions	
Triggers	There is someone who intends to do test or maintain data

Figure 4 Use Case Description

2.2.3 User Views

Data	Access Type	User	Administrator
Countries	Maintain		X
	Search	X	X
	Report		X
Cities	Maintain		X
	Search	X	X
	Report		X
Attractions	Maintain		X
	Search	X	X
	Report		X
Registered Users	Maintain		X
	Search		X
	Report		X

Figure 5 User Views

2.2.4 Non- Functional Requirement

System:

The route programming system should be developed in Java and the subsystem that involves machine learning should be developed in python. The database should be maintained in MySQL.

Speed:

1. The recommendation route should be provided to the multiple users within one second after each request.
2. The number of processed transactions per second is above 1000
3. The event response time should below 100ms.

Usability:

Inexperienced users shall be able to use all the system functions under the instruction without any training. The average number of cancelled queries due to inability to operate shall not exceed five per day.

Security:

1. The password for registered users should be longer than eight letters, with at least one uppercase letter, one lowercase letter and one number.
2. The user password and database should be protected by cryptography like *Advanced Encryption Standard* (*AES*).

Portability:

The user interface should be compatible on any screen size for all available devices (desktop, mobile phone, and tablet).

Size:

The system size should not exceed 200 megabytes.

For method of evaluating reliability and robustness, there are not specific criteria values to judge the requirement for the time being since some metrics are hard to quantize at this stage for the development team. In terms of reliability, the mean time to failure, probability of unavailability, rate of failure happened are important metrics to verify the reliability. In terms of robustness, time to restart after failure, percentage of events causing failure and probability of data corruption on failure should be considered.

2.2.5 Constraint on the Software/Hardware

There are no obvious constraints for this software projects desired features, however, this might be changed in the course of development.

2.3 Anticipated Experiment

System owns many properties, for instance, speed, size, ease of use, reliability, robustness and portability. Experiments are carried out on the system in order to verify these properties. Goal-based evaluation will be adopted, which specific goals are prior to be evaluated, compared with existing condition. After confirming goals, evaluators need to assess to which extent these predefined goals are achieved in program. In this project, our team confirms function which system provides. After testing and debugging problem of system, our team will assess whether system functions satisfy the requirements. Additionally, our team also uses formative evaluation method, which combines with goal-based evaluation, evaluates user actions and contrasts with perception to evaluate whether goals are suitable for users and system. For functionality and interface of system, black-box testing and white-box testing are applied, black-box testing is to test the software interface and the function. The white box testing is to exercise all program statement. The Cyclomatic complexity equals to the number of conditions in a program plus one, it is applied to describe effect of optimization. Stress testing is also adopted to evaluate system performance when beyond its maximum design load in order to evaluate reliability. Integration and cluster testing tests complete systems or subsystems composed of integrated components, which aims to evaluate system reparability. For evaluating usability of system, scenario-based testing is an appropriate method based on use cases.

2.4 Evaluation Method of Work

For methods of evaluation of work, we conduct a survey to find out whether our product is suitable or convenient for user. This survey will be published after the implementation of demo version. The team will modify product according to users' suggestion. //However, our team will not completely depend on feedback of survey.

Additionally, benchmark software will be applied to test. Testing result will be analysed on response speed, network security, robustness, reliability to improve or optimize the product.

3 Conduct of Project and Plan

3.1 Preparation stage

The team members were assigned tasks based on their skills. However, additional skills should be learnt to be capable of finishing this project.

Zhixuan Xu specialized in: Specification & documentation skills
Needs to learn: Ant, testing and JUnit
Hongyi Wu specialized in: SQL, python
Needs to learn: Web crawler and data mining
Zhuo Wang specialized in: SQL, python
Needs to learn: Web crawler and data mining
Xiaoyu Qin specialized in: HTML, UI design
Needs to learn: Adobe Photoshop, Responsive Web Design
Yifan Hu specialized in: AI, python machine learning
Needs to learn: Suitable training model, data processing
Weiqi Chen specialized in: Java programming
Needs to learn: Shortest path algorithm, dynamic programming

Figure 6 Task Allocation

3.2 Design Stage

The overall design of this project adopted the incremental design. The design stage consists of four parts. Firstly, the team needs to have an understanding about the problems in this case. Then the team will try to identify one or more solutions. After figuring out the potential solutions, the team will describe solutions abstractions. In the end, repeat the process above according to the requirements. Object-oriented design method is mainly implemented in this system design. Each objects in the system works together. They communicate by exchanging messages.

For the whole system design, it may be taken part into six phases: architectural design, abstract specification, interface design, component design, data structure design and algorithm design.

In the architectural design phase, the team will design a suitable system architecture according to the requirements specification.

In the abstract specification phase, the team will specify the sub-system and give abstractions on these designs.

In the interface design phase, according to the sub-system design, the team will describe the interfaces of sub-system.

In the component design phase, what the team will do is decomposing sub-system into components. Give specifications on these components of sub-system.

In the data structure design phase, the team will need to design a data structure to store data in this system. Find the most suitable structure, which can make the system efficient and safe.

In the algorithm design phase, the team will design algorithm for each functions in the system.

During the six phases of design, there are five principles need to be paid attention. There are linguistic modular units, few interfaces, small interfaces, explicit interfaces and information hiding.

3.3 Implementation Stage

Hardware: Any device that supports browsing websites that are compatible with HTML mark-up language.

Software: HBuilder, Visual Studio, Adobe Dreamweaver, Adobe Photoshop, Mozilla Firefox, WordPress.

During the implementation stage, from the perspective of testing, integration testing, interface testing, stress testing, black box testing and white box testing and will be carried out. For integration testing, every single function will be tested individually and afterwards the integrated system function will be tested as a whole part. During the process of integration testing, the method of black box testing is applied as the tester only need to give valid or invalid inputs, rather than being required the code level knowledge of the program. In addition, for interface testing, the interface with external applications like Google maps should be tested whether the API is always valid and available. As for stress testing, the overload due to enormous amount of access should be concerned. Some simulation of overload conditions should be rehearsed in advance. In all, if any errors were encountered with, the method of white box testing is applied from the internal perspective of the system, to go through all lines of codes to fix the problem, which occurs in a single unit.

3.4 Gantt Chart

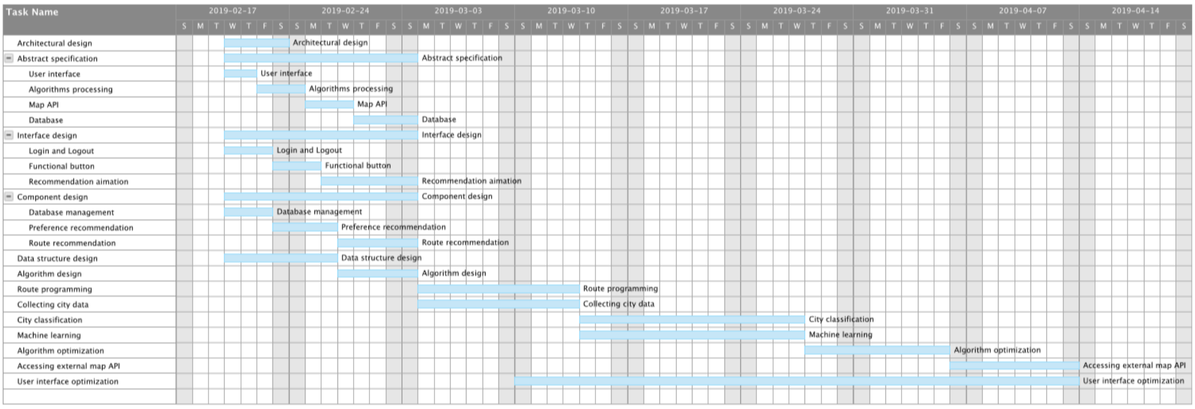

Figure 7 Gantt chart

3.5 Risk Assessment and Major Challenges

The problem for the development team is how to recommend a route based on start point, destination and user preference, additionally, if user does not input destination, system need to recommend route based on information of user preference. In order to solve this problem, it is necessary to acquire geographical location popular city or place and optimization algorithm to recommend suitable travel route. Our project‘s algorithm is based on Dijkstra algorithm to optimize shortest path because Dijkstra algorithm is suitable for condition that destination and departure are known. The problem was how to select city along route based on number of tourism attraction and convenient degree of transportation. To solve this problem, weighted matrices will be set based on large dataset. For condition which user does not input destination and departure, the system will recommend route based on preference of user, result will be obtained from model. The model is based on data of many travel routes of one country which given by many travellers who has different preferences. The development team will search popular cities or attractions, which derived from traveling route of other travellers who has similar preference with user. The solution about this function are acquiring data or information from personal blog, traveling website and training model based on data. When the model is trained, the recommended traveling route will be based on preference of user.

The major challenges in carrying out project are designing effective shortest path algorithm, train model which recommends traveling route based on preferences of user, constructing database consists of geographic locations of cities, geographic location of popular attractions places, opening hour of tourist attractions, transportation method of traveling to destination and building connection between database, training model and website.

The new skills are required for this project are constructing database, technique of web crawler, designing new algorithm based on Dijkstra algorithm and train model which is adopted for recommending traveling route based on preference of user. These skills are acquired by previous module, relevant books and internet source.

4 Bibliography

References

[Book]

[1] Ian Sommerville, *Software Engineering 10th ed.*, Pearson, Harlow, Essex, England, 2016

[2] Connolly and Begg, *Database Systems 6th ed.*, Pearson, Harlow, Essex, England, 2015

[Lecture Slides]
[3] Sebastian Coope "Software Engineering I," https://vital.liv.ac.uk/, February 2018

[4] Michele Zito "Group Software Project, " http://cgi.csc.liv.ac.uk/~comp208/, January 2019.

[Online Conference]
[5] Chen et.al "Information Systems Evaluation Methodlogies,"http://eprints.whiterose.ac.uk/74740/1/WRRO_74740.pdf, July 2011