CS491 - SENIOR DESIGN PROJECT

Travimize: Optimized Travel, Personalized Experience

PROJECT SPECIFICATIONS REPORT

Supervisor: Selim Aksoy

Innovation Expert: Perin Ünal

Team ID - T2330

Doğa Ece Ersoy - 21902487

Doruk Kantarcıoğlu - 21902319

Emre Erdal - 21901597

Esra Genç - 21901962

Kerem Erdal - 21901596

1. INTRODUCTION

1.1 Description

Travimize is an application which tries to facilitate people's lives while making plans for their holidays. First of all, it tries to apply the optimization process for the journeys by allowing the person to visit the places they want at the shortest time. User can first select his/her interests, the places he/she wants to visit during his holiday, and traveling options like walking, public transport,etc. By observing the opening, closing times, distance between two locations, and the choices of the user, the algorithm will provide the optimal route for the journey. By the way, users would also be able to get informed about the distance between two locations and which transportation would be fastest that would help their selection process. The application would just propose a schedule which can further be changed by the person according to his/her own desires, other plans or some schedule changes. Along with these scheduling plans, the user will be able to specify some price, eating, accommodation related choices. Accordingly, hotels and restaurants will be proposed by the application and the user can also be directed to associated websites for hotel reservations. Moreover, the user will be able to rate their plans or specifically places and can write comments which would provide feedback not only for the other customers but also to program admins. These feedback questions would be about the time spent in a specific location, if that place is open during the visit, and then future plans will be created by taking these into consideration. With the help of these evaluations, a user going to a specific location would be firstly informed about the high rated plans and then he/she may have a chance to select these previously liked plans by making some changes or applying them straightforwardly. Finally, the users would also be able to see their previous plans and may repeat or personalize it when they visit the same place again. As a result, Travimize helps people to optimize their holidays and personalize the schedules by specifying their interests at the beginning or changing them according to their desires at the end.

The type of innovation that is planned to be implemented for Travimize is service. As Travimize aims to bring a new perspective to traveling and ease the process of vacation planning, it is included in the category of service innovation that prioritizes customers and their experience. Travimize is also open to performance and functionality enhancements

according to the feedback coming from its customers. The application is a radical innovation as it combines accommodation, food, transportation, and tourism industries. [1].

1.2 Constraints

1.2.1 Implementation Constraints

- GitHub will be used for version control.
- The application will be a web-based application mainly used by websites.
- Java will be used as the programming language for back-end development.
- Spring will be used as the framework for back-end development.
- React will be used as the framework for front-end development.
- JavaScript will be used as the programming language for front-end development.

1.2.2 Usage Constraints

- Users should sign up before logging in to the system.
- During sign up, users should write their names and emails. Also create both username and password.
- Users should be logged to the system to use the application.
- Any event in a specific tour cannot be in the same hour.
- Users should receive a tour according to their constraints they provided with the form submitted before the tour is created.
- All the users must have a name, email, password and username. In the create table query of the User, these attributes are declared as not null.
- All the tours must have a name, creation status, approximate fee, rating and time. In the create table query of the Tour, these attributes should not be null at all.
- Users should be able to retrieve their saved tours.

1.2.3 Economic Constraints

- The cloud services that the application utilizes such as Google Cloud Platform (GCP) for Google Maps API should not exceed certain utilization budgets declared by the project group.
- The application will be free to use.

1.2.4 Environmental Constraints

- The routes and places that are presented to the user should be based on current locations, meaning location data should be up to date.

1.2.5 Social Constraints

- Users should be able to share their created trips with other users.

1.2.6 Language Constraints

- Application's main language will be English, hence the HTML accepted language locales will be en-US and en-GB.

1.2.7 Ethical Constraints

- Regardless of how accurate the data in the application is, Travimize would never claim to be able to 100% point out misinformation and should only be used as a guideline to create a travel route and plan.

1.2.8 Health and Safety Constraints

- The routes will be created by offering safe places and avoiding the roads on the way that pass through unsafe regions.
- The routes will be created by considering the maximum distance a person should walk in a day. Walking some distance may be possible theoretically, but it would not be offered if that much walking may result in some negative health consequences.

1.2.9 Sustainability Constraints

- The application would be developed by observing the previous feedback of users. When a route is preferred by some users, it will be proposed to other users traveling to the same location.
- When new restaurants, hotels, or tourist places open, Travimize will be updated to include those places.
- When some information comes about the change related to opening, closing hours of a location, it will directly update the database of Travimize.

1.3 Professional and Ethical Issues

According to GDPR, location data is considered as a sensitive piece of information [2]. Hence, we must first ask for the consent of the user in order to process their location data and also we are obliged to enforce paramount security in our data storage processes. Since the location data is directly linked to users, previously visited locations/routes need to be preserved properly in the application, preferably in a role-based access system.

In addition to location data, user credentials and authentication/authorization processes need to be handled with care. The database will not store unnecessary but sensitive information such as date of birth, gender, salary. In the worst case of a data breach, password information will be encrypted instead of being stored in plaintext fashion. This encryption step will also ensure that the application cannot read information it doesn't directly need or utilize.

Due to the scale of personal data the application utilizes, the source code, environment variables such as database credentials and crucial config files will be accessible only by developers.

Finally, us developers will follow IEEE Code of Ethics to preserve the highest level of integrity and responsibility [3]. Non-anonymized user data will not be shared with any third-party applications at all.

2. REQUIREMENTS

2.1 Functional Requirements

- Personalized Schedule

  Users will be able to request for a personalized traveling plan that satisfies their requests and constraints. These requests and constraints will be explained in the next bullet point. When users receive their personalized schedule, they will be able to see a plan with possible hotel options to stay at, possible restaurant options to eat breakfast, lunch or dinner at. They will also be able to see the touristic landmarks, natural beauties, historic places, museums, etc. in their plans. There will also be time slots to visit famous parks, beaches, etc. Moreover, shopping malls and other famous shopping streets and stores will be taken into consideration in the schedule if the user prefers it. They will know the temperature of every place they will visit exactly. Not only the user will be able to see at which time they will visit these places, they will also be able to get the information about how to reach those places, with which transportation vehicle, and how long it will take. They will know how much money they will need to spend for entrance fees, transportation fees, etc. In summary, the users will have an optimized traveling plan that shows where to be and what to do every minute of the day for how long their plan is.

- Constraints

  Users will determine what constraints their schedule will have. They will fill out a form before getting the traveling plan and the plan will be created according to the users' constraints and preferences. These include:

  - Where in Turkey they will go, and at most how far can a place be for them to visit.
  - For how long their trip will last. And between which dates they will visit.
  - How much money they want to spend in total for the trip (a range).
  - For how long do they want to sleep a day? Minimum, maximum and ideal sleep times.
  - Which transportation vehicles they prefer, or will never use. Do they have a car?

- For how long they can walk, and for how long they can climb a hill or a ramp. They can provide an average number of steps for one day.
- What type of tourist attractions they prefer. For example,
  - landmarks
  - historic places
  - shopping areas
  - famous local food
  - beaches
  - forests
  - natural parks
  - other parks
  - museums
  - religious places, etc
- What type of food they like to eat. What are their allergies? Which local foods they want to try.

- <u>Providing necessary resources for completing the plan</u>

  Users will be able to see possible hotel options to stay at, possible restaurant options to eat their meals at. They will also find the necessary links to the websites that they can make reservations to, or get information from.

- <u>Modifying the personalized plan</u>

  When the personalized traveling plan is created and displayed to users, they can modify the plan if they are not satisfied with any part of it. For example they can request more time to visit a museum, they might not like the suggested touristic places and request for new ones. Moreover they might say they do not want to go too far to visit a place or they can just want a calmer trip to rest.

- Feedback for the optimized plan

  If the user chooses a plan to follow, and go on their trip, they will be able to give feedback to the plan so that next time, their plan can be better. For the feedback, they will be asked to provide information about the questions below:

  - They will be asked to rate and a give a detailed review for a plan
  - They will be asked if the suggested places they visited were open at the reserved time slot, and if the information given about the plan was correct.
  - Did they feel unsafe at any time when they were on their trip? If so, when and where?
  - Were the given time slots enough to visit the place? Did they take any longer? If so, where and how much more time did they need?
  - They will be asked if they would recommend this plan to others. And if they do not recommend it, why?
  - They will also be asked to rate and give a review to the hotels and restaurants.
  - Lastly, they will be asked if there is anything they want to add.

- Favorite schedules

  Users will be able to see available traveling plans which were made as an example, or traveling plans other people made, and recommend. They will be able to choose and personalize these plans as they wish.

- History

  Users will be able to see the old plans they created and their feedback to them.

2.2 Non-Functional Requirements

2.2.1 Usability

- The user interface should be easy to use or learn for maintaining maximum convenience and user friendliness for the target audience.
- Each component of the user interface should have a meaningful and clear name.
- The user interface should offer a guide on navigating the website for users.
- All screens must be independent of one another, with the exception of pop-ups, to prevent users from having to navigate backward through the application.

2.2.2 Robustness

- The database should be created in such a way that it can easily respond to the requests from the application.
- The database should respond to an input that is prone to errors without any issues.

2.2.3 Security

- All of the personal data of the users (name, mail, phone number, etc.) should be protected by the application.

2.2.4 Recovery

- Regular database backups are advised. Data should be saved in the event of a system failure.

2.2.5 Scalability

- The application should handle the excessive number of users logging in to the system without crashing.

3. REFERENCES

[1] J. Desjardins, “10 types of innovation: The art of discovering a breakthrough product,” *Visual Capitalist*, 02-Jul-2020. [Online]. Available: https://www.visualcapitalist.com/10-types-of-innovation-the-art-of-discovering-a-breakthrough-product/. [Accessed: 16-Oct-2022].

[2] D. FRANCIOLI, “GDPR and location data,” Joinup, 15-Dec-2020. [Online]. Available: https://joinup.ec.europa.eu/collection/elise-european-location-interoperability-solutions-e-government/news/gdpr-and-location-data. [Accessed: 16-Oct-2022].

[3] IEEE, “IEEE Code of Ethics”. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 16-Oct-2022]