Project Report
CS 411 SP24 Team 65
AccommoSeek

Bo-Hao, Wu Chien-Kai, Kuo Si-An, Chen Sheng-Min, Lin

1. **Please list out changes in the directions of your project if the final project is different from your original proposal (based on your stage 1 proposal submission)**.

   Overall, the direction of our project mostly aligns with our proposal in stage 1. Some minor changes that have been made will be listed below. First, homepage design is mostly the same. The only part being altered is the interactive map we initially plan to embed, but it is removed due to the limited time we have to finish the whole project. Secondly, we originally planned to let users input Precipitation and CrimeRate to filter the result of hotels. However, we realize it is hard for users to estimate the number, so we decide to divide these into three levels so users can easily opt for the desired result.

2. **Discuss what you think your application achieved or failed to achieve regarding its usefulness.**

   As an accommodation seeking platform, our application successfully takes into account the weather, crime rate and living wage of where the hotel is located. These are some of the factors that no other existing accommodation booking platforms is using as their critirion. However, as a platform for accommodation searching, we didn't provide users the direct access to book their ideal accommodation. Therefore, even if the user finds the ideal hotel they are looking for, they will have to google it and find the link to book it.

3. **Discuss if you change the schema or source of the data for your application**

   No, we didn't change the source of our data for our application but we did make some changes to our schema compared to stage 1 proposal. Originally, the LivingWage and CrimeFrequency are two independent tables. However, after analyzing the relation we decide to put LivingWage and CrimeFrequency into the City table as two columns. In the original design, we have population, CrimeFrequency, City and State in the CrimeRate table. However, city and state is needed to create the City table, and the population of a city should also be a column of itself. Also, the original LivingWage table only contains two columns:

City and LivingWage. It won't make sense to put them in another independent table. That is why we decided to put these columns into the City table.

4. **Discuss what you change to your ER diagram and/or your table implementations. What are some differences between the original design and the final design? Why? What do you think is a more suitable design?**

   Most of our design is based on our original UML design. Some minor designs have been changed in order to make the database more robust. For example, the Hotel table, the initial idea is to use the hotel name as the primary key of the Hotel table. However, we found out that the name of hotels is not unique. Some hotels could have several branches in many different cities, so we change to use Name and CityName together as the primary key.

   The Review table is also changed accordingly, because it's a weak entity set referencing the primary key of the Hotel table. As a result, CityName is also included for the primary key of the Review table.

5. **Discuss what functionalities you added or removed. Why?**

   We initially included an interactive map on the homepage, but we didn't implement it at the end. We decided to put more effort on the core functionality of the project, for example, the robustness of the database and backend optimization. The interactive map is therefore removed. Also, we want to attach the url link of each accommodation in the search results. However, some of the accommodations in our data source don't have the official website attached, which might lead to inconsistency between results. There, we decided to remove the urls.

6. **Explain how you think your advanced database programs complement your application.**

   The use of a trigger DeleteCityHotels in our database ensures that whenever a city is deleted from the database, the corresponding hotel will also be removed. They ensure the consistency of the database.

   The use of stored procedures DivideLivingWage, DivideCrimeRate and DividePrecipitationenables enables our database to automatically update the threshold of these filters. Also, the use of stored procedures Search will be used when inputting keyword search, which enhances the performance of the search by saving time on parsing the SQL.

7. Each team member should describe one technical challenge that the team encountered. This should be sufficiently detailed such that another future team could use this as helpful advice if they were to start a similar project or where to maintain your project.
    - Sheng-Min, Lin: As I'm in charge of the database part, it is important to be able to write SQL fluently. However, since this is the part which is covered in the very beginning of the semester, I need to go through the course slide again to pick up some difficult parts. Also, some advanced database programs are very new to us. For example, we have never tried to implement the stored procedure and trigger before. It is very helpful to follow examples of triggers and stored procedures on the course slides.

    - Si-An, Chen: We try to build our MySQL database on the google cloud platform, it is a totally new experience for all of us. This can be more difficult than building a local database. That's why this is not a required but extra part for the project implementation. Some major issues that we run into is connecting to the database from our application, after we do some research we realize some configuration needs to be done on the GCP setting. As a result, I'd say that try to build a local database first and if that works out and you want to learn more, then try the GCP database afterward.

    - Bo-Hao, Wu: In our original plan of the database, the rating of a hotel is designed in the Review table. However, when we're implementing the search function on our homepage. The performance is very slow because the database will need to compute the rating every time. Since this is very inefficient, we decided to make some optimization. In the end, we decided to create another trigger to pre-compute the rating of each hotel. After the optimization, the search formance has significantly improved.

    - Chien-Kai, Kuo: We are using Next.js as the front end framework for the project. This is one of the most popular front end frameworks in the recent market. However, this is very new for me, I've never developed a front end with a framework. As a result, I spend a lot of time reading through documentation and understanding the structure of the framework. I would recommend maybe trying using plain HTML and CSS to develop the project. That will save a lot of time on researching a lot of front end knowledge which is not the key part of the project.

8. **Are there other things that changed comparing the final application with the original proposal?**
   In the original design, we plan to let users input some data like temperature, crime rate, living wage and precipitation. However, after we start implementing the project and designing the user experience step by step. We realized that it is against human nature to ask them to input numbers for some of these fields. As a result, we changed the design to set temperature, crime rate, living wage and precipitation into three categories. With the change of design, the user is more able to set search filters without worrying about the number they need to input.

9. **Describe future work that you think, other than the interface, that the application can improve on**
   For now, our application doesn't come with a user login process. As a result, most of the operation does not verify user identification. It would be nice to add this functionality.
   For accommodation recommendation websites, I think users would like to see pictures uploaded by other users. This will be closer to reality.
   More advanced filters can be introduced, for example the price range of the accommodation or if it comes with a free parking lot.
   Integrated Google map into our application, users will be able to have a quick glimpse of the location of a hotel with this feature.
   Link to an official website or other third party hotel booking system, with this feature users will be able to redirect to their ideal website to book the accommodation.

10. **Describe the final division of labor and how well you managed teamwork.**
    The final division of labor is approximately the same as described in the stage 1 project proposal. Chien-Kai is in charge of most of the front-end implementation and design. Bo-Hao is in charge of most of the back-end setup and implementation. Si-An and Sheng-Min work together for the database design and advanced database features. Though there are different parts that each member is responsible for, we usually meet up in person to work together so everyone can learn different aspects of the project. In conclusion, the teamwork distribution in our group is fairly balanced and dynamically adjusted so we can help each other when someone is stuck with the work.