

Project Report

- 1. Please list out changes in the directions of your project if the final project is different from your original proposal.**

The general directions of our project remain the same as what we stated in our original proposal. We aimed to build a reciprocal coordinate system that provides convenience for users to find properties easily and allows landlords to rent properties, and this is what we have accomplished in our final project.

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

Our application now supports users of different roles (e.g. users and administrators) to use our system by logging into our application using their phone number and password. Users can see the full list of available tenements and the according information such as names, property type, numbers of bedrooms and bathrooms, and price, etc. Administrators can see the full list of properties as well as the list of users, and they are able to add, edit, or delete the stored data of users, properties, or hosts. Furthermore, our system supports search queries that are input by users, for example, users can search for properties that are in a certain price range or with a certain number of beds. For administrators, our application also supports batch processing of the data of properties, for example, administrators can increase the price of all the properties that are owned by a specific host at once instead of having to increase the price one by one.

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

No, we did not change the schema or source of the data of our application. Because if I changed the schema or source of the data for my application, that would require some development work and would affect the whole application. It would involve changing the database structure, adding or removing tables, changing fields and data types, and changing any existing queries and other code that interacts with the database. It would also likely involve changing the front-end interface and the backend code. Depending on the extent of the changes, it could be a time consuming and expensive process.

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?

The original ER diagram and table implementations had some major differences in the way data was organized which required a change in the overall design. Firstly, the ER diagram lacked some table relationships which were necessary to make the design more efficient. Secondly, the table implementations did not include any foreign key constraints which were necessary to ensure data integrity. After implementing these two changes, the ER diagram and table implementations were more in line with a normalized design which is more suitable.

The primary differences between the original design and the final design lie in the relationships between tables. The original design did not include references between tables, and foreign key constraints were absent. The normalized design includes references between tables, and foreign key constraints which ensure data integrity. The normalized design is more suitable for larger data sets, as it reduces redundancy and eliminates potential data integrity problems.

The most suitable design for the data set ultimately depends on the number of records, data types, and the query requirements. If the data set is small with few records and simple set of queries, then the original design could be considered. However, if the data set is large and complex queries need to be implemented, then the normalized design should be implemented. In this case, the normalized design is more suitable as it ensures data integrity and allows for easier manipulation of the data.

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

We have added some advanced functions, such as helping homeowners quickly modify the rent prices of all their houses, and we can also query the transactions of the same house to help determine whether there is fraud activity.

However, due to the judgment of users' gender and age, we cannot provide the sharing renting function at present. We need to verify the user's relevant information before safely providing other users with this user's sharing renting invitation.

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

Using the relevant advanced programs, we utilize the procedure to help homeowners quickly modify their rental prices. This function can save many homeowners the time of

modifying prices one by one. However, at the same time, to maintain the balance of house leasing and avoid misoperation, we set triggers to limit the change of single house rent price. For the convenience of demonstration, we set it to 1000 dollars here.

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.**

ZHENGXU JIN: One technical challenge that our team encountered on designing our Shareroom application was making sure that the data was stored and retrieved from both the frontend and the backend correctly. We initially used local MySQL as the backend database, but quickly found that it was too rigid and slow when multiple users tried to access the database simultaneously. We then switched to Google Cloud Platform, which gave us the flexibility to scale our application as needed. We also had to work together to ensure that our frontend JavaScript code was able to properly access and handle the database data, as well as make any necessary updates and deletions. By testing and adjusting our code accordingly, we were ultimately able to create a reliable application that worked both on desktop browsers and mobile devices.

YANGSEN CHEN:

One technical challenge we faced was connecting the frontend and backend. We had a clear separation of our frontend framework, written in Javascript and html, and our backend technology, written in Flask. However, communication between the two had poor synchronization, and our API calls often relayed back errors due to the backend not receiving certain data fields. We eventually solved this issue by making sure that all of the data we sent our backend was encapsulated and labeled properly in one centralized object, which would then require only a single API call. This allowed us to successfully submit requests and reliably receive responses from our backend. This could be useful advice for other teams looking to effectively bridge the gap between their front- and back-end code.

ZIHAN LI:

One technical challenge on a database system might encounter performance degradation over time. As the database grows and is used more heavily, the performance of specific queries may suffer. This can be caused by many factors, including the size of the data being queried, the complexity of the queries being run, and the efficiency of the database's indexing strategy. To address this issue, the team may need to optimize the database's schema and implement more efficient indexing strategies. This can involve

analyzing the performance of individual queries and identifying which ones are slow, and implementing techniques such as denormalization and materialized views to improve query performance. Additionally, the team may consider using a more robust database server, such as a distributed database system, to handle the increased load. Moreover, it can involve setting up a cluster of database servers and implementing a load-balancing strategy to distribute the workload across the cluster.

HAOTIAN WANG:

I think one of the technical challenges is to design and implement a clean and efficient database. Since a good database is the most important and fundamental thing in this project, whether the design of the database is good or not really affects a lot on other tasks such as database manipulation in SQL, as well as some of the backend and frontend works. For the database, we made the design as concise as possible and we made sure that the data types and foreign keys match so there won't be any issues when we do the database manipulation.

8. Are there other things that changed comparing the final application with the original proposal?

Yes, there are a few changes. In addition to the features mentioned in the proposal, the final application also includes features such as payment system, search engine, user profile and messaging system. Additionally, the final application also includes security measures such as encryption, two-factor authentication, and automated fraud detection to keep users safe and secure while using the platform. Moreover, the implementation of the shared rental function was changed to allow users to search for roommates based on more than just gender, age and hobbies. Instead, users can now filter searches by a variety of criteria, from price range to amenities to location. Finally, the credit rating system was also changed to include features such as rewards and discounts for tenants and landlords with higher credit scores.

9. Describe future work that you think, other than the interface, that the application can improve on.

In the future, I think the application can be more user friendly by adding additional features such as search capabilities, automated listings, filters and sorting. Additionally, user login capabilities could be enhanced. This would allow Shareroom to easily individualize the experience of each user, helping them find the accommodation they are looking for.

The application should also be able to integrate with other third party services and applications such as payment systems, travel providers and calendar applications. This would enable users to quickly book their stays and access additional information, such as prices and discounts.

Finally, the application should be made more secure, with robust measures such as encryption for handling sensitive data. This will help protect users from potential security breaches, and give them the confidence that their information is being securely handled by the application.

10. Describe the final division of labor and how well you managed teamwork.

ZHENGXU JIN:

For the project, our final division of labor was fairly divided and we all worked together to create the Shareroom application. I worked on designing the database as well as the triggers and stored procedures that had to be implemented in the backend. Additionally, I also implemented some backend functionality and integrated the application into our API.

Overall, working as a team on the project was a great success. We were all able to contribute in different capacities and take advantage of our individual strengths. Our communication was always open and we worked effectively together to create an end product that met our customers' needs.

YANGSEN CHEN:

The final division of labor involved each of us taking on a different section of the project. I assigned myself the pre-production phase, while my partner handled the production and post-production sections. We both then tackled the tasks within those sections to the best of our ability.

I found that we worked pretty well together throughout the process. We communicated regularly and openly to figure out the details of the project and make sure we brought our best work to the table. We listened to each other's ideas and critiques and used them as foundations to improve on our respective sections. In the end, I believe we both managed to produce a project that neither of us could have achieved alone.

I always strive to ensure that everyone on the team is respected, and their input is taken into account. I aim to foster an atmosphere of collaboration, constructive criticism and problem solving. I also strive to make sure everyone has opportunities to lead, develop and strengthen their skills. I believe communication is key in any team, both within the team itself and between the team. I strive to create processes that ensure productive

communication and make sure everyone is properly informed about the progress of the project. Finally, I ensure that clear individual and team goals are set and that each team member is held accountable for doing their part to ensure successful completion of the project.

HAOTIAN WANG:

The work distribution among team members for this project is fair. I am happy to work with my team members and this is a good teamwork experience. My share of work mainly includes collecting and cleaning data sets that we need to put in the database, constructing and improving database design and implementation in GCP, some of the back-end implementation using Flask, and writing triggers and procedures in SQL.

ZIHAN LI:

My final division of labor is database design and related code implementation. Database design involves planning and creating the structure and organization of a database in order to store and manage data effectively. This typically involves defining the data types, relationships, and constraints used to organize the data in the database. It also involves designing the database schema, a visual representation of the database structure that shows how the different data elements are related. Code implementation we choose python to implement our functions. This typically involves using a programming language such as SQL to write the code that will be used to create the database structure, insert data into the database, and perform various operations on the data.

Effective teamwork is essential for us and our work. First, we communicate openly and regularly. Good communication is key to ensuring that all team members are on the same page and working towards the same goals. It can be achieved through regular meetings, email updates, and other forms of communication. Moreover, we set clear goals and expectations to make sure that everyone understands what they are working towards.