**Counting Sheep**

Amir Sadiev, Imran Ahmed, Maclean Frazier , Walt Brady

Table of Contents

**1. Project Definition (**100 - 200 words**)** –*Group responsibility*

The purpose of this project is to make a hotel management system that allows both front and back end users to access and use the system. Since this is an application that will allow users to book hotel rooms, it will need friendly and easy to navigate GUI so that booking a room, requesting services and other accommodations is not an issue. The application will also support back end needs to employees whom follow up on their customer’s requests in order to make sure that all necessities are in place and issues are taken care of, as a key feature of this application allows users to directly make requests to the employees of the hotel they are staying in. All of this will be achieved by having a good application that is easy to use on both front and back ends, will allow easy modification, an up to date database that will make sure that there are no miscommunication anywhere in the application.

**2. Project Requirements** – *Group responsibility*

* Functional

The application will serve as a way for clients to book different hotels and rooms to try to get the best prices, deals, request services. It will serve as the gap between the hotel and the user and be the middleman where the customer will try to get the best deal. On the back send, it will allow operators of different hotels to manage their task all in one place. Customer could request service, cleaning and other responsibilities and the staff would get on it to get them done while they can also see other aspects of the business on the back side.

* Usability
  + User interface

There will be a GUI application that will allow both users and hotel employees to retrieve hotel information and make requests in an intuitive manner.

* + Performance

The program will load itself within a few seconds. The system will also load information from the database to the user quickly to prevent wasting the user’s time.

* System
  + Hardware

The system will be a desktop application and can be converted to a mobile application so users can use it anytime they want.

* + Software

This will be a desktop application connected to a SQL database. It will use Java FX to display information to the user

* + Database

This system will use SQL database that updates in real time in response to user inputs.

* Security

Each user will have their own login credentials so that no one else can log in their account. Once the user logs in, there will be a security question the user will need to answer before they can access their entire account. They can reset their password using the question or another form of sensitive information like their social security number.

**3. Project Specification** – *Group responsibility*

* Focus / Domain / Area: Hotel Booking/Management for both front and back end users - Both the users or guests and the users as in management for the hotels will be able to use the application all in one place. You will not have to book in one place, and use another application to contact hotel staff or anything and the hotel will be able to contact you directly from the same application making it as smooth it can be.
* Libraries / Frameworks / Development Environment: This application will be programmed in JavaFX using the NetBeans IDE. It will use the Java Authentication and Authorization Service (JAAS) framework to maintain security in the whole system. The application will also use the JDBC API in order to connect the application to a database. The database itself will be created using mySQL in mySQL Workbench.
* Platform (Mobile, Desktop, Gaming, Etc): Desktop. The application will be a desktop application, and all the users will only be able to access it from a PC or laptop only. It can be developed into a mobile application but for now we will keep it PC only.
* Genre (Game, Application, etc): Travel/Management Application - The application is a travel application where it allows people to book rooms in hotels from anywhere. It will allow users to try to get the best rates possible and will allow the hotel owners to keep up with all the booking and accommodation from the website also.

**4. System – Design Perspective** – *Group responsibility*

* Identify subsystems – design point of view
  + - GUI

JavaFx - The reason I am using JavaFX is because it is a good language that has many features when designing a program. Since the other subsystems are being designed in java, it would be easier to connect all the subsystems if everything was written in the same language rather than different languages. JavaFx provides a lot of features on the design side that would help make a really good, easy to use GUI. It would be easy to use in the user side since the overall flow of the program would be smooth.

* + - Database

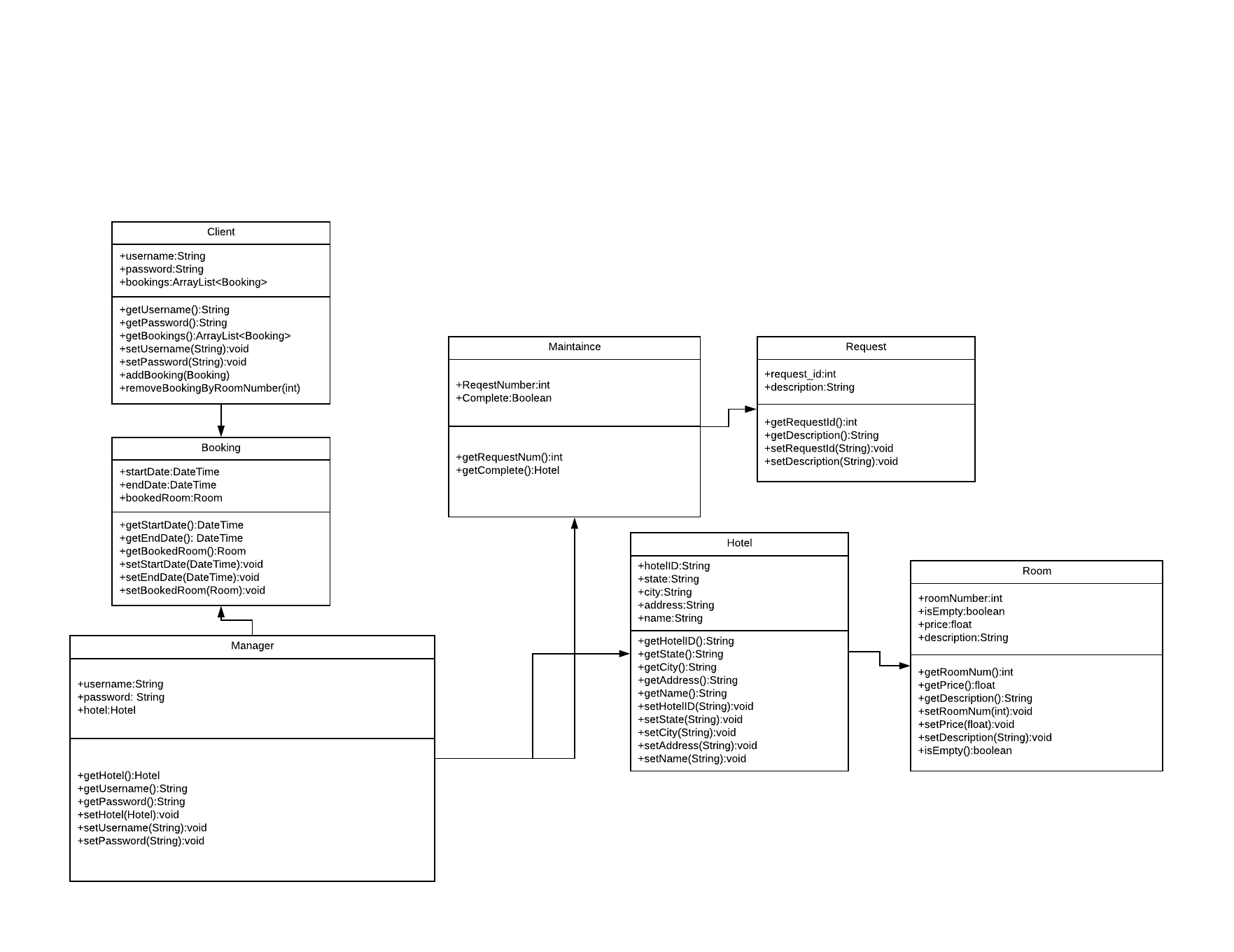
The database subsection is focused around developing a mySQL database and managing interactions between the main program and the database. The reason why mySQL is chosen over other options like noSQL is because mySQL is a relational database, meaning that querying data across multiple tables is easy and efficient and will prevent the database from slowing down the main application.

* + - Customer

Client side will use the UI to send requests such as choose date of bookings, list available hotels, as well as book rooms. These processes will send requests to the server for the respective information. Once processes are fully finished, the client will see a notification (confirmation) of completion. The whole project is based around Java language, Java FX (GUI) and mySQL database.

* + - Manager

The manager side, much like the customer subsystem, will be managed using Java. The subsystem will be designed to offer different services from the customer side, and will be linked to the database via room management. We are using java since it allows the creation of objects such as rooms and hotels that the manager will be able to easily manipulate through drop-down menus and text entry.

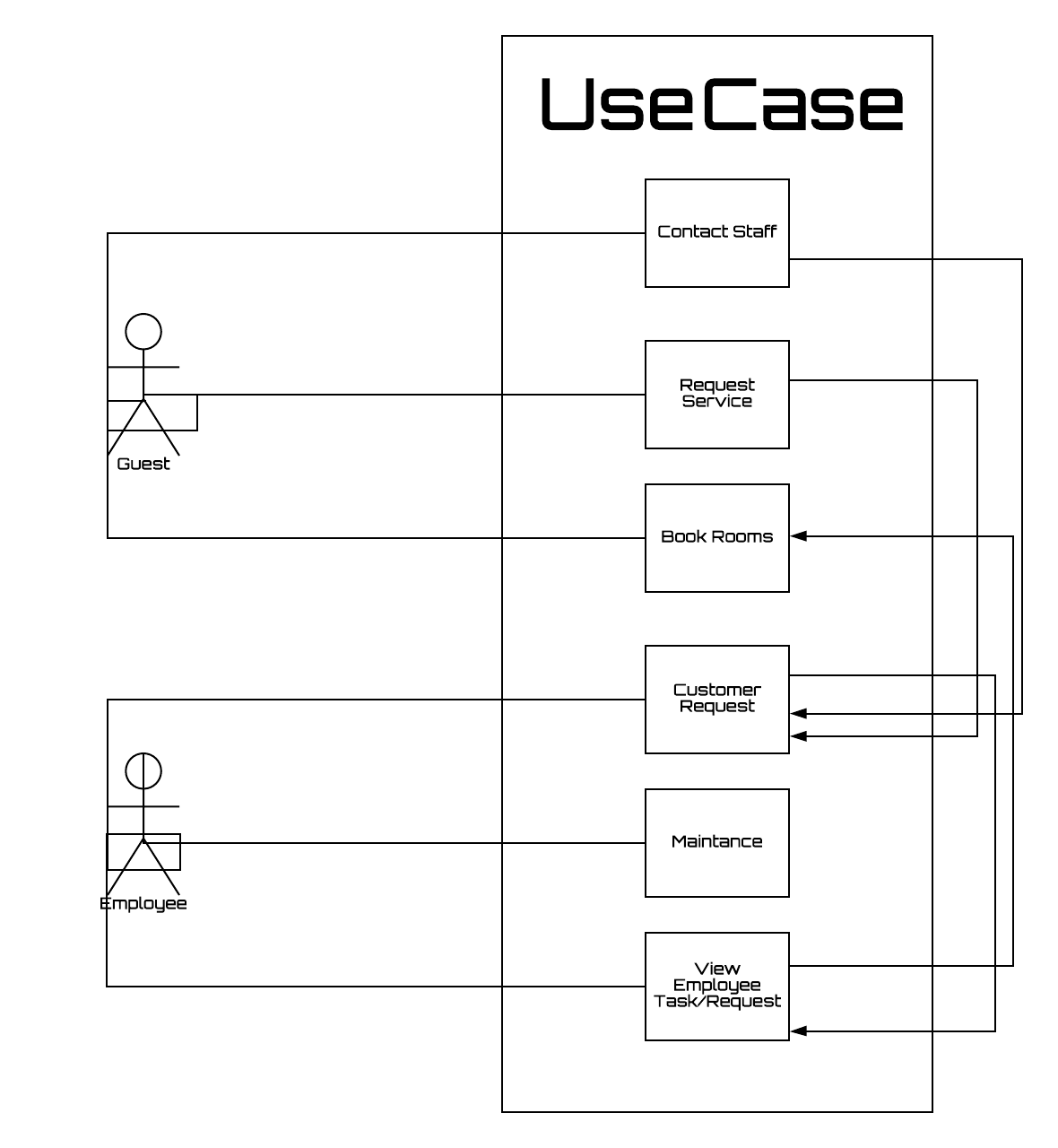


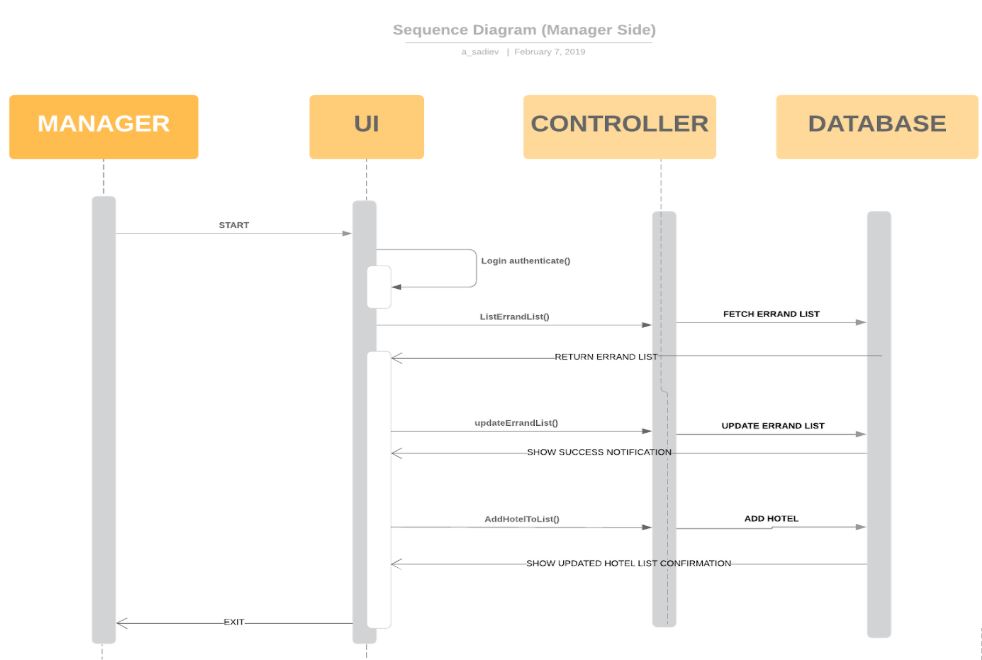
UML

Example UML for the Manager subsystem.

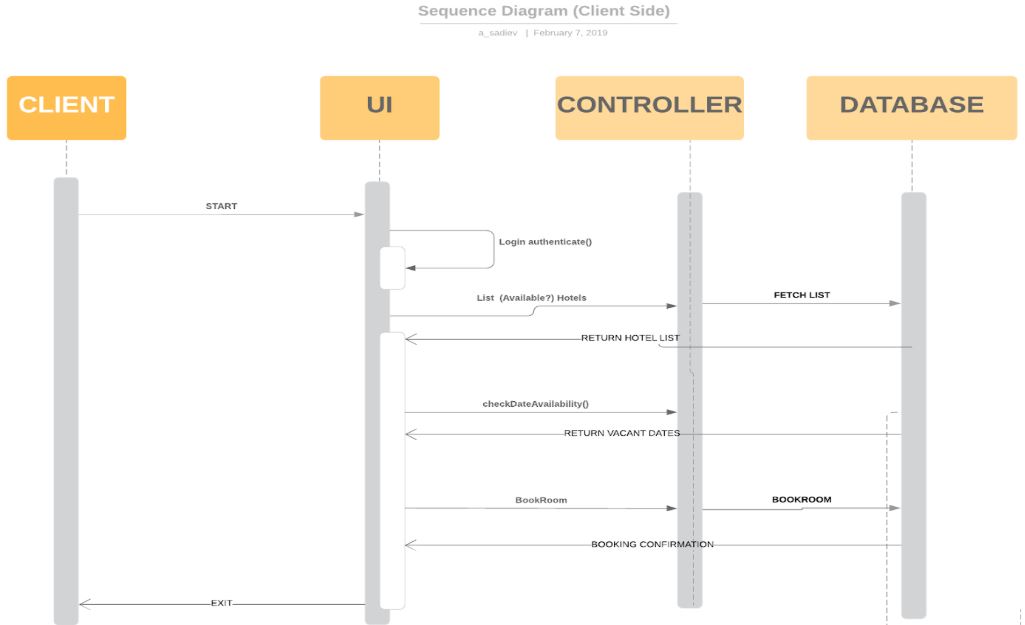
Use Case Diagram

The use case diagram represents both the guest and the employee. The Guest can contact the staff, request service and book rooms and then the employee would be notified and would take care of the request. One the employee side, they can follow up with customer request and/or complete their personal tasks assigned to them.





Sequence diagram for the manager side of the application, shows the sequence of events for a manager logging into the system.

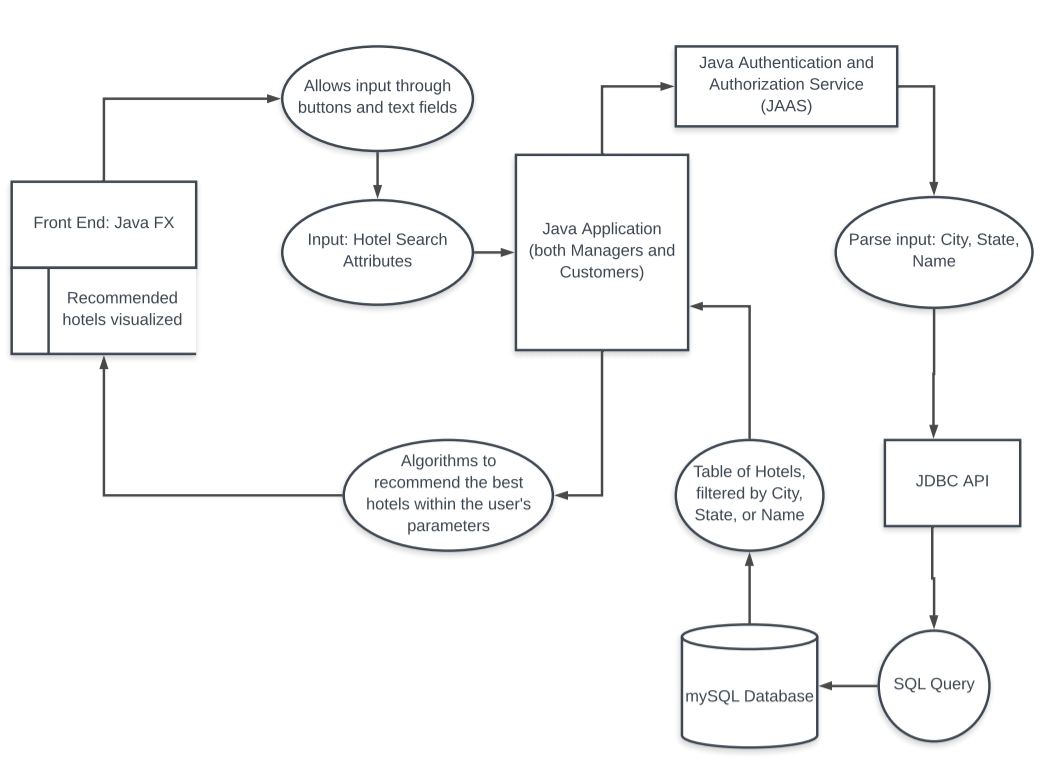


Sequence diagram for the customer side of the application, shows the sequence of events for a customer logging into the system.

* + Design choices

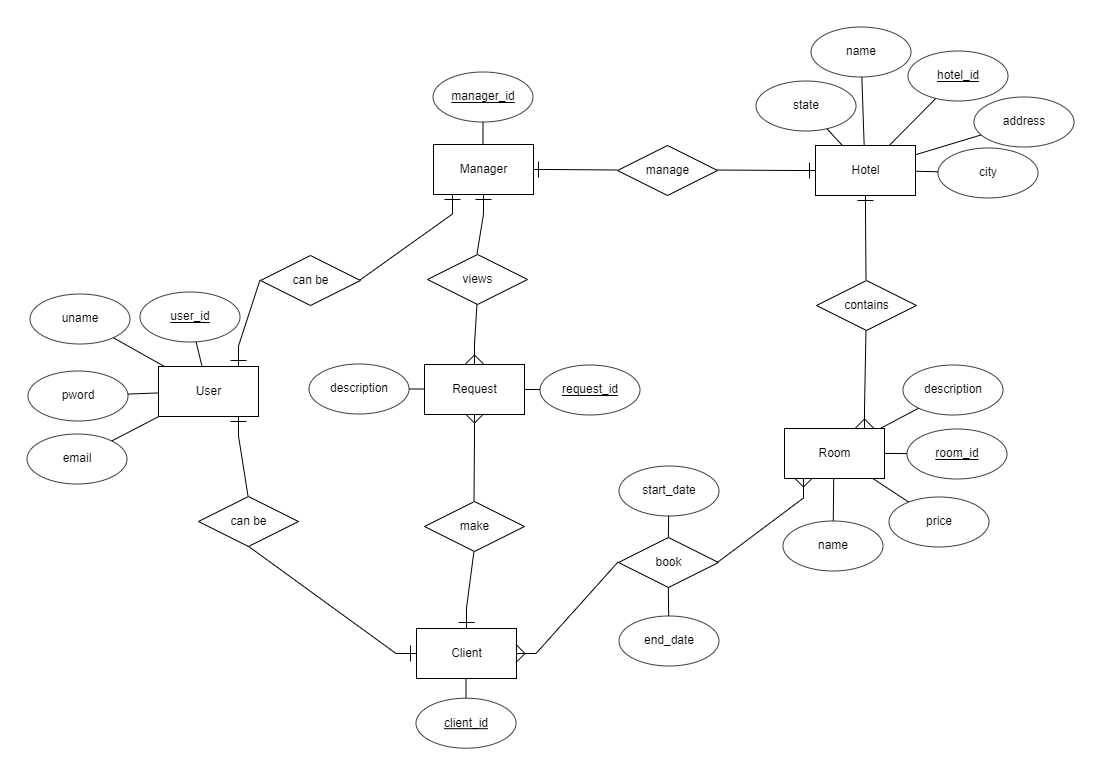
We are doing a desktop application for the moment. The desktop application will update in real time via the database, will make sure that the application is smooth running. The application can be incorporated into a mobile application down the road so that both users and employee can follow up on the go.

* Sub-Systems Communication (Diagram and Description)
  + Controls
  + I/O
  + DataFlow

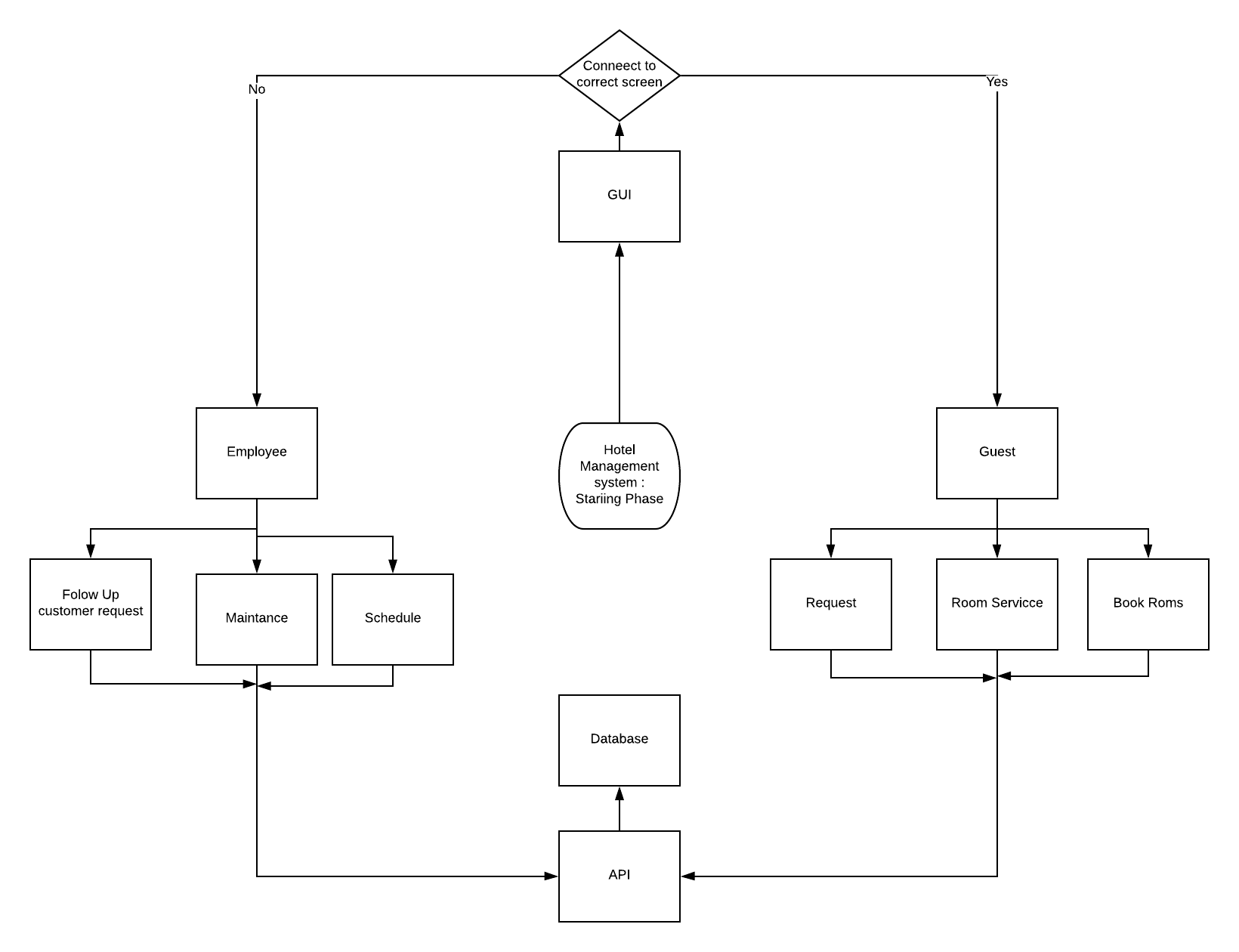


This diagram shows an example of how the data flows throughout the system, where a client searches for hotels with specific attributes.

* Entity Relationship Model (E-R Model)



* Overall operation - System Model
  + Simplified Sub-system to System interaction



**5. System – Analysis Perspective** – *Group responsibility*

* Identify subsystems – analysis point of view
  + GUI -

The GUI will be based of the login credentials when first logging in. Once logged in, the menu will display the appropriate menus according to who logged in and what all the options they have. Not all employees will have the same options since there will be different levels of employees and the same will go with the guest, they will have different menus according to their hotel plan.

* + Database

The database must be designed in such a way that it does not slow down the rest of the program. To do this in mySQL, indexes will be used. The database itself will be stored on Google’s Cloud SQL platform to allow quick access from multiple users at the same time. This allows both customers and managers to access valuable data as quickly as possible.

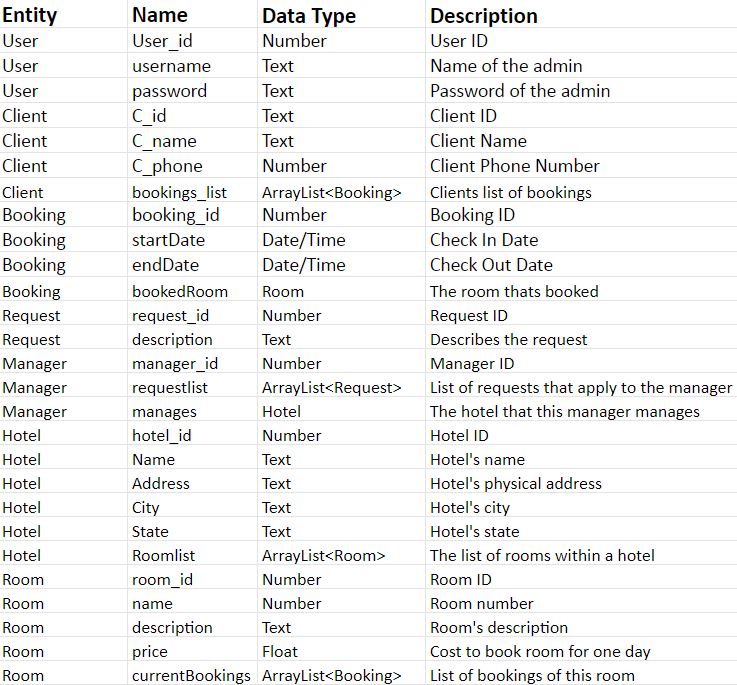
* + Customer

Customer will be sending requests to the database via GUI. The program will allow the clients to choose the hotel, dates, and provide optional feedback. The process should be quick-working and intuitive.

* + Manager

The manager will mostly be a series of setters and getters that refer to the database. This will be used by the manager of a given hotel, and be used by them to adjust prices and add rooms to the database.

* System (Tables and Description)
  + Data analysis
    - Data dictionary



* Algorithm Analysis
  + Big - O analysis of overall System and Subsystems
    - GUI

The big O analysis for the GUI is O(1), since it is constant, once the user logs in, it will direct them to either the employee or the customer subsystems

* + - Database

The time analysis of querying data in the database is O(log(n)).

* + - Customer

Big O Analysis: O(log(n))

* + - Manager

The big O analysis of the manager algorithms will be O(log(n)) since it will rely on the way the database is constructed.

**6. Project Scrum Report -** *Group Responsibility*

* Product Backlog (Table / Diagram)
* Sprint Backlog (Table / Diagram)
* Burndown Chart

**7. Subsystems**

**7.1 Subsystem 1** – Name 1 - GUI

Initial design and model

* + Illustrate with class, use-case, UML, sequence ..... diagrams
  + Design choices
* Data dictionary
* If refined (changed over the course of project)
  + Reason for refinement (Pro versus Con)
  + Changes from initial model
  + Refined model analysis
  + Refined design (Diagram and Description)
* Scrum Backlog (Product and Sprint - Link to Section 6)
* Coding
  + Approach (Functional, OOP)
  + Language
* User training
  + Training / User manual (needed for final report)
* Testing

**7.2 Subsystem 2** – Name 2 - *Individual responsibility*

* Initial design and model
  + Illustrate with class, use-case, UML, sequence ..... diagrams
  + Design choices
* Data dictionary
* If refined (changed over the course of project)
  + Reason for refinement (Pro versus Con)
  + Changes from initial model
  + Refined model analysis
  + Refined design (Diagram and Description)
* Scrum Backlog (Product and Sprint - Link to Section 6)
* Coding
  + Approach (Functional, OOP)
  + Language
* User training
  + Training / User manual (needed for final report)
* Testing

**7.3 Subsystem 3** – Name 3 - *Individual responsibility*

* Initial design and model
  + Illustrate with class, use-case, UML, sequence ..... diagrams
  + Design choices
* Data dictionary
* If refined (changed over the course of project)
  + Reason for refinement (Pro versus Con)
  + Changes from initial model
  + Refined model analysis
  + Refined design (Diagram and Description)
* Scrum Backlog (Product and Sprint - Link to Section 6)
* Coding
  + Approach (Functional, OOP)
  + Language
* User training
  + Training / User manual (needed for final report)
* Testing

**7.4 Subsystem 4** – Name 4 - *Individual responsibility*

* Initial design and model
  + Illustrate with class, use-case, UML, sequence ..... diagrams
  + Design choices
* Data dictionary
* If refined (changed over the course of project)
  + Reason for refinement (Pro versus Con)
  + Changes from initial model
  + Refined model analysis
  + Refined design (Diagram and Description)
* Scrum Backlog (Product and Sprint - Link to Section 6)
* Coding
  + Approach (Functional, OOP)
  + Language
* User training
  + Training / User manual (needed for final report)
* Testing

**8. Complete System** – *Group responsibility*

* Final software/hardware product
* Source code and user manual – screenshots as needed - Technical report
  + Github Link
* Evaluation by client and instructor
* Team Member Descriptions

***This is just a guide, and use it to create/improve your report. Feel free to add sections. You are responsible for your own subsystem/s, not other members. You have to contribute to the team’s goals and objectives, and develop your subsystem/s, write your documents and slides.***