1. INTRODUCTION

1.1 PURPOSE

The purpose of this document is to build an online system to manage peer groups for online learning in any course.

1.2 DOCUMENT CONVENTIONS

This document uses the following conventions.

DB	Database
DDB	Distributed Database
ER	Entity Relationship

1.3 INTENDED AUDIENCE AND READING SUGGESTIONS

This project is a prototype for the peer review system. This project is useful for the online course instructor and as well as the students undertaking the course.

1.4 PROJECT SCOPE

The purpose of the peer review system is to help students clear their doubts or facilitate better communication and peer learning and provide transparency to course instructor on a group level.

1.5 REFERENCES

- <link of app when live>

2. OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE

A distributed peer group database system stores the following information.

- **Admin details:**

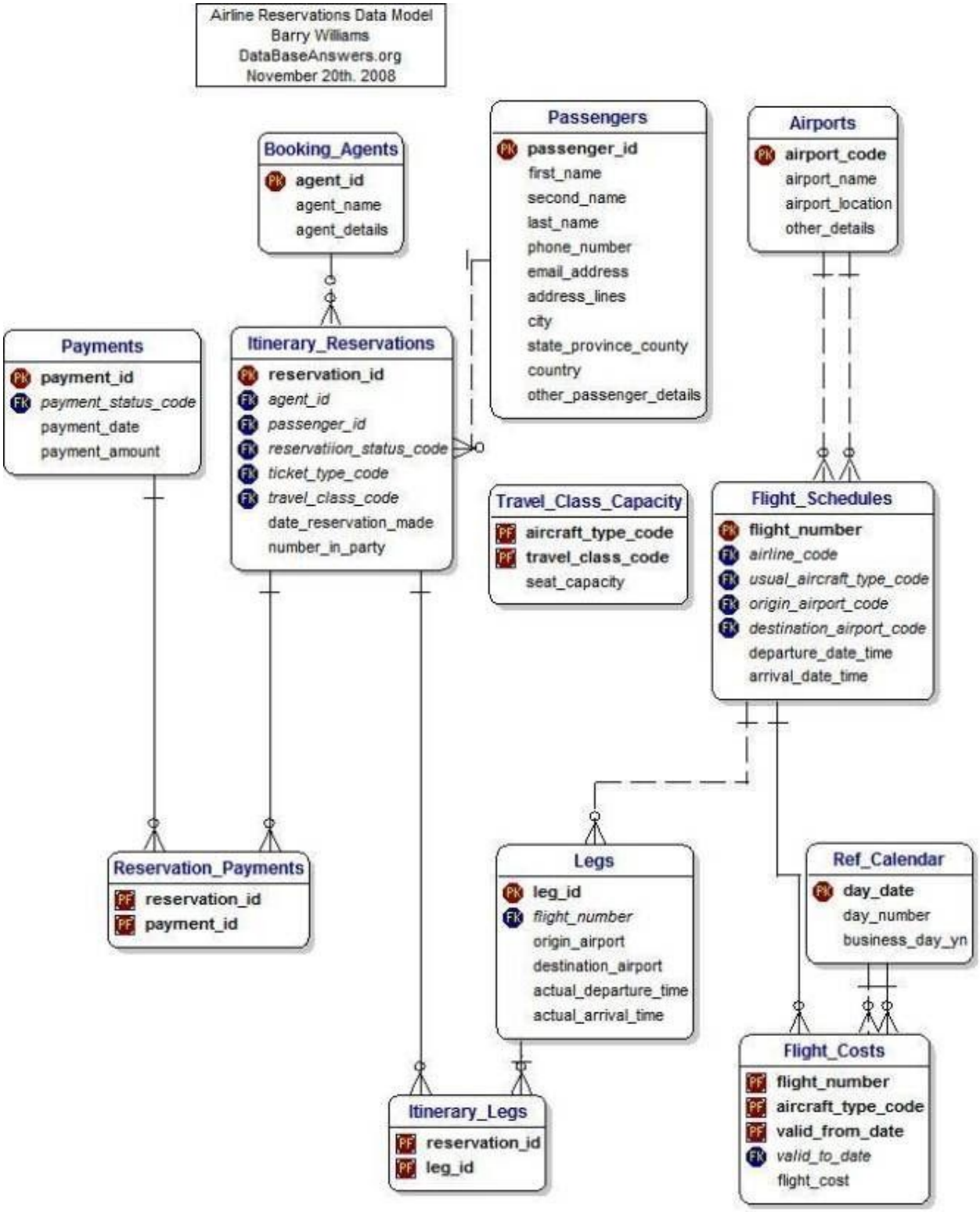
  It includes the students and group lists, and forms needed to track the group progress.
- **Group details:**

  It includes group members (with a group leader). This information may be used for keeping the records of the group members for any emergency or for any other kind of information.
- **User details:**

  It includes user details, form responses, progress percentage, and basic details, name, groupId, studentId.

2.2 PRODUCT FEATURES

The major features of airline database system as shown in below **entity–relationship model** (**ER model**)



The diagram shows the layout of airline database system – entity–relationship model

2.3 USER CLASS and CHARACTERISTICS

Users of the system should be able to retrieve tasks and peer review forms from the database and check their progress and group details. The system will support two types of user privileges, Admin(Course Instructor), and Students. Users will have access to user functions, and the admins will have access to both user and peer group management functions. The student should be able to do the following functions:

- Fill up checklist for to-do problems, availability form before meetings, peer review forms after peer meetings, as well as doubt forms.
- Check his/her group details and their performance

The Admin should have following management functionalities:

- Admin FUNCTIONS.

  • Share forms for assignment checklist, meeting availability time, doubt clearance and peer group review and share with each group.

  • Get all filled-up form responses from each group.

  • View group progress.

  • Send mail to students with pending backlogs.

  • Regroup to maintain uniform balance in terms of performance at group level.

ADMINISTRATIVE

• Add/Delete a group

• Add a new student

• Update groups.

• Add a new group.

• Update meeting schedules

• Move/Rearrange students amongst groups for balanced performance

Each group has a limited number of students.

2.4 OPERATING ENVIRONMENT

Operating environment for the airline management system is as listed below.

- distributed database
- client/server system
- Operating system: Ubuntu.
- database: sql database
- platform: ReactJS/Django

2.5 DESIGN and IMPLEMENTATION CONSTRAINTS

1. The global schema, fragmentation schema, and allocation schema.
2. SQL commands for above queries/applications
3. How the response for application 1 and 2 will be generated. Assuming these are global queries. Explain how various fragments will be combined to do so.

4. Implement the database at least using a centralized database management system.

2.6 ASSUMPTION DEPENDENCIES

Let us assume that this is a distributed peer group system and it is used in the following application:

- A request for booking/cancellation of a flight from any source to any destination, giving connected flights in case no direct flight between the specified Source-Destination pair exist.
- Calculation of high fliers (most frequent fliers) and calculating appropriate reward points for these fliers.

Assuming both the transactions are single transactions, we have designed a distributed database that is geographically dispersed at four cities Delhi, Mumbai, Chennai, and Kolkatta as shown in fig. below.

3. SYSTEM FEATURES

- **DESCRIPTION and PRIORITY**

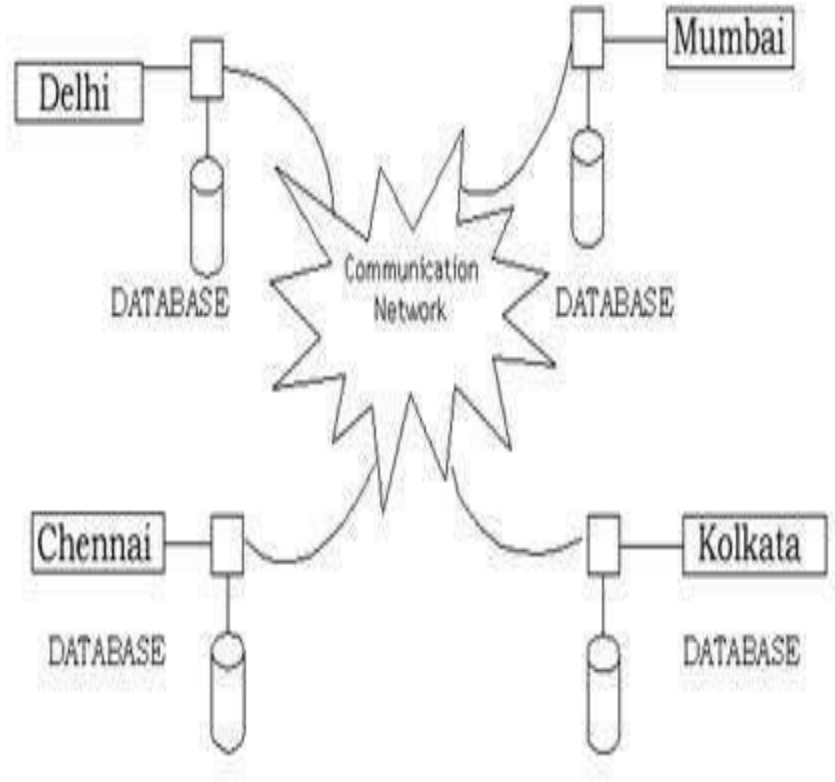
The peer group system maintains information on students individual as well as group progress, information about students having pending backlogs who need to be notified or regrouped in a way that the backlogs get cleared in time.

- **STIMULUS/RESPONSE SEQUENCES**
    - Search for students or a particular group.
    - Displays a detailed list of students/group and options to edit group, or delete student from that group.
    - Cancel an existing group, or re-group
- **FUNCTIONAL REQUIREMENTS**

Other system features include:

DISTRIBUTED DATABASE:

Distributed database implies that a single application should be able to operate transparently on data that is spread across a variety of different databases and connected by a communication network as shown in below figure.



Distributed database located in four different cities

CLIENT/SERVER SYSTEM

The term client/server refers primarily to an architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the database along with backend django app (also known as the back-end).

A client/server system is a distributed system in which,

- Some sites are client sites and others are server sites.
- All the data resides at the server sites.
- All applications execute at the client sites.

4. EXTERNAL INTERFACE REQUIREMENTS

4.1 USER INTERFACES

- Front-end software: ReactJS
- Back-end software: Django

4.2 HARDWARE INTERFACES

- Ubuntu.
- Any modern browser

4.3 SOFTWARE INTERFACES

Following are the software used for the flight management online application. <<*Include the software details as per your project* >>

Software used	**Description**
Operating system	We have chosen Ubuntu operating system for its best support and user-friendliness.
Database	To save the flight records, passengers records we have chosen SQL database.

Django and React	To implement the project we have chosen django framework as backend and ReactJS for frontend.

4.4 COMMUNICATION INTERFACES

This project supports all types of web browsers. We are using simple electronic forms for the to-do/checklist forms, peer review forms etc.

5. NONFUNCTIONAL REQUIREMENTS (only replace flight with our app name)

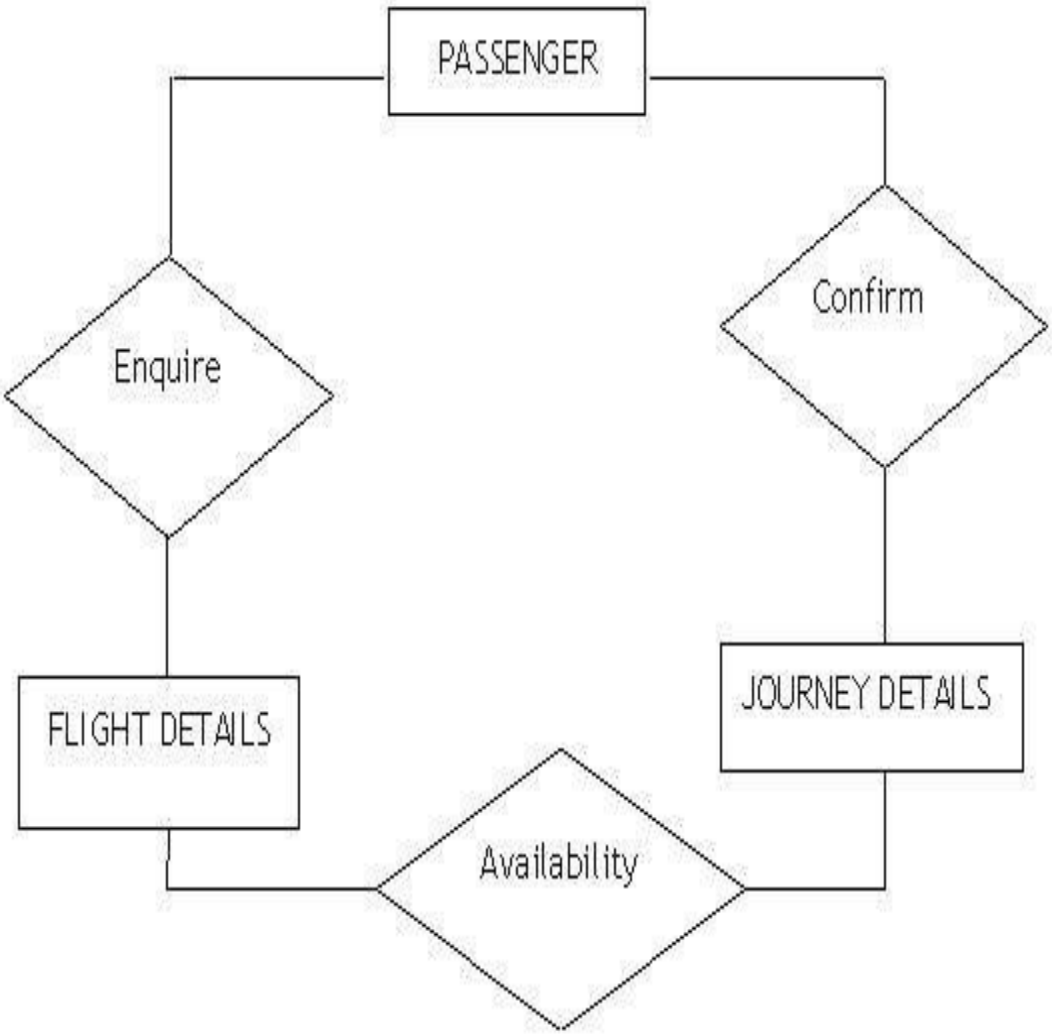
5.1 PERFORMANCE REQUIREMENTS

The steps involved to perform the implementation of airline database are as listed below.

A) E-R DIAGRAM

The E-R Diagram constitutes a technique for representing the logical structure of a database in a pictorial manner. This analysis is then used to organize data as a relation, normalizing relation and finally obtaining a relation database.

- **ENTITIES:** Which specify distinct real-world items in an application.
- **PROPERTIES/ATTRIBUTES:** Which specify properties of an entity and relationships.
- **RELATIONSHIPS:** Which connect entities and represent meaningful dependencies between them.



the diagram shows the ER diagram of airline database

B) NORMALIZATION:

The basic objective of normalization is to reduce redundancy which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of the data stored.

If a database is not properly designed it can give rise to modification anomalies. Modification anomalies arise when data is added to, changed or deleted from a database table. Similarly, in traditional databases as well as improperly designed

relational databases, data redundancy can be a problem. These can be eliminated by normalizing a database.

Normalization is the process of breaking down a table into smaller tables. So that each table deals with a single theme. There are three different kinds of modifications of anomalies and formulated the first, second and third normal forms (3NF) is considered sufficient for most practical purposes. It should be considered only after a thorough analysis and complete understanding of its implications.

5.2 SAFETY REQUIREMENTS

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

5.3 SECURITY REQUIREMENTS

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

5.4 SOFTWARE QUALITY ATTRIBUTES

- **AVAILABILITY:** The student should be available on the specified date and specified time for peer meeting as all students are doing advance reservations.
- **CORRECTNESS:** The progress for each individual as well as group should be accurate and consistent across all devices.

- **MAINTAINABILITY:** The peer group system ensures to maintain correct schedules of meetings as well as update groups and progress on the go.
- **USABILITY:** The meeting schedules are aimed to clear all the pending backlogs and all doubts from doubts form should be addressed and cleared for all students in the course.