# Dialdeal system Report

# GroupNo.-3





# **Table of Contents:**

**Contents**

[Dialdeal system Report 1](#_Toc130932468)

[GroupNo.-3 1](#_Toc130932469)

[Table of Contents: 2](#_Toc130932470)

[Executive Summary: 3](#_Toc130932471)

[Design Analysis Process: 4](#_Toc130932472)

[User Stories: 4](#_Toc130932473)

[Use-Case Diagram: 5](#_Toc130932474)

[Usecases: 6](#_Toc130932475)

[Functional Requirements: 7](#_Toc130932476)

[Non-Functional Requirements: 8](#_Toc130932477)

[Entity-Relationship Diagram: 9](#_Toc130932478)

[Logical ERD: 9](#_Toc130932479)

[Physical ERD: 10](#_Toc130932480)

[Data-Dictionary: 11](#_Toc130932481)

[Git Repository: 16](#_Toc130932482)

[Contributions: 16](#_Toc130932483)

# **Executive Summary:**

Dialdeal is the system which can be used by Users and Businesses. In this system, Businesses register and give all information about them such as Name, Address, Contact No., Website link and email. Users also register in the system and give their information. Businesses provide different types of services such as Food & drinks, Entertainment, Beauty and Accommodation. User can see the list of services, offers which are provided businesses. Users can see the information of businesses such as Name, Address, Mobile No., Email. Users can call to businesses to get services information and to fix the service. Users can give the Reviews and Rating on Services which are provided by particular Businesses. Businesses get reviews for their services so, according to reviews & rating they can make change on their services.

# **Design Analysis Process:**

## **User Stories:**

As a user, I want to be able to insert, update, and delete my profile information (Name, Address, Contact No., Email) so that I can keep my information up to date.

As a user, I want to be able to see the services provided by business owners so that I can find the service I need.

As a user, I want to be able to get the contact information of business owners so that I can communicate with them regarding their services.

As a user, I want to be able to see and compare the offers for services provided by different business owners so that I can choose the best offer.

As a user, I want to be able to add, update and delete reviews for the services provided by business owners so that I can share my experience with others.

As a user, I want to be able to delete my account from the system if I no longer wish to use the platform.

As a user, I want to be able to view the service information provided by a particular business owner so that I can decide whether to avail their services.

As a business owner, I want to be able to insert, update, and delete my profile information (Name, Address, Contact No., Email) so that my customers can find my services easily.

As a business owner, I want to be able to delete my account from the system if I no longer wish to use the platform.

As a business owner, I want to receive reviews and ratings for my services so that I can improve the quality of my services.

As a business owner, I want to be able to view the number of reviews/ratings based on a criteria (such as time frame or service category) so that I can get an overview of my service quality.

## **Use-Case Diagram:**

## **Usecases:**

1. Users can insert, update and delete the profile information (Name, Address, Contact No., Email)
2. Users see services provided by Business owners
3. Users get contact information of Business owners
4. Users can see and compare the offers for services
5. Users can add, update and delete reviews for the service which are provided by Business owner
6. Users are able to delete the account from system.
7. Users should be able to view service information provided by a particular business.
8. Business owner can insert, update and delete the profile information (Name, Address, Contact No., Email)
9. Business owner are able to delete the account from system.

10. Business owner will get reviews and rating by their services

11. Business owners should be able to view the number of reviews/ratings

Based on a criteria.

## **Functional Requirements:**

**User Profile Management:**

The system should allow users to create and manage their profile, including their name, address, contact number, and email address.

Users should be able to update and delete their profile information at any time.

**Service Management:**

The system should allow business owners to create and manage their service offerings.

Users should be able to view the services provided by business owners.

The system should allow users to view the service information provided by a particular business owner.

**Contact Information Management:**

The system should allow users to get the contact information of business owners to communicate with them regarding their services.

**Offer Comparison:**

The system should allow users to compare the offers for services provided by different business owners.

**Review Management:**

The system should allow users to add, update, and delete reviews for the services provided by business owners.

Business owners should be able to receive reviews and ratings for their services to improve the quality of their services.

**Account Management:**

The system should allow users to delete their account from the system if they no longer wish to use the platform.

Business owners should also be able to delete their account from the system.

## **Non-Functional Requirements:**

**Performance:**

The system should be fast and responsive, with quick load times and minimal delays.

The system should be able to handle a large number of users and transactions without slowing down or crashing.

**Security:**

The system should be secure, with appropriate measures in place to protect user data.

User authentication and authorization mechanisms should be implemented to ensure that only authorized users can access the system.

**Usability:**

The system should be easy to use and navigate, with a simple and intuitive user interface.

The system should be accessible to users with disabilities, following accessibility guidelines.

**Reliability:**

The system should be reliable, with minimal downtime and high availability.

Regular backups should be performed to ensure that user data is not lost.

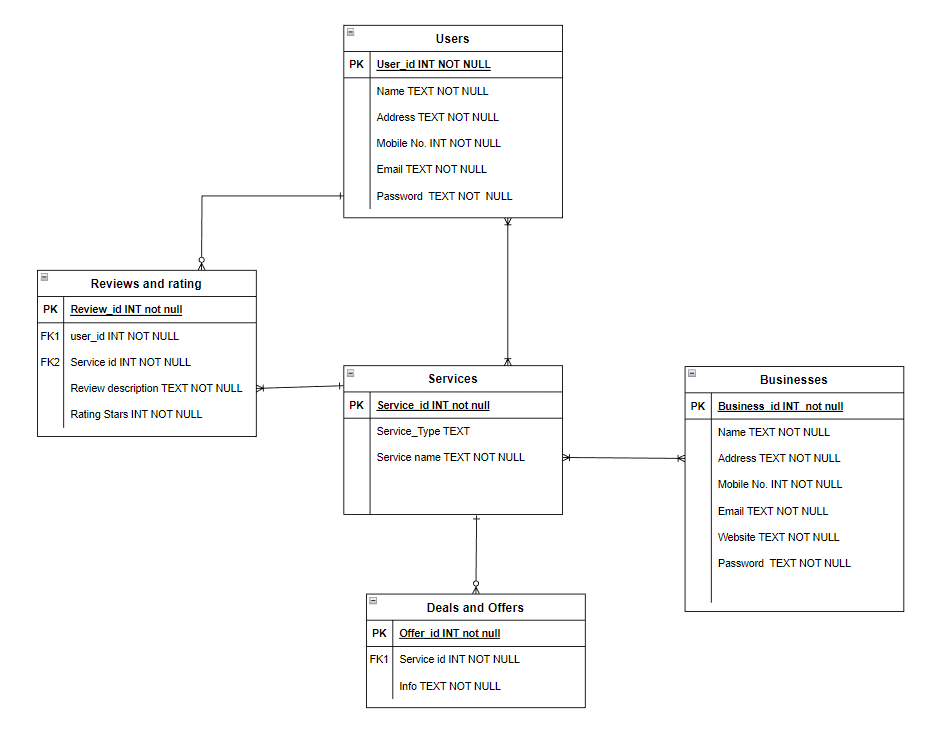
**Scalability:**

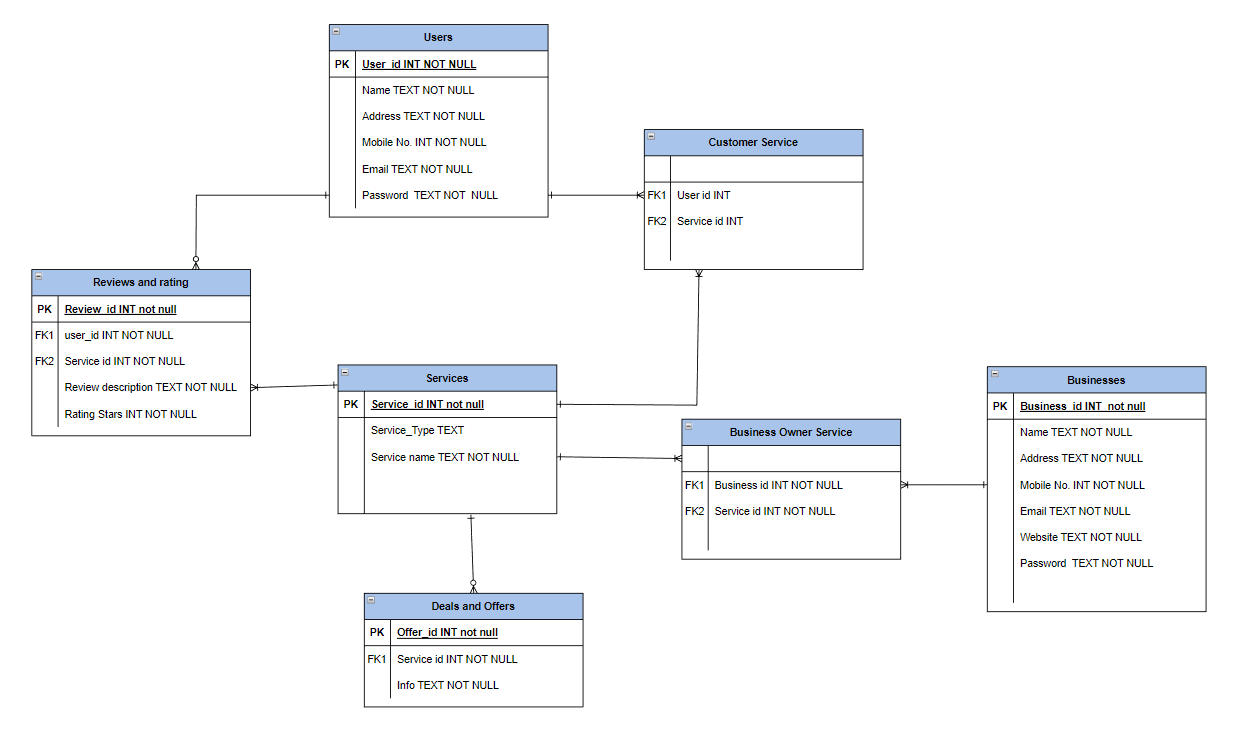
The system should be designed to be scalable, with the ability to handle increasing traffic and user demand over time.

The system architecture should be flexible and modular to accommodate changes and updates as needed.

**Entity-Relationship Diagram:**

## **Logical ERD:**



******Physical ERD:**

**Assumptions for ERD:**

* We allow many users or businesses to join our system. Each user is identified by key User\_id or Business\_id.
* One Business can provide many services each service is identified by key Service\_id.
* Users can access many services .
* Users can give reviews for services provided by particular business. Each review is identified by key Review\_id.
* Services can have many offers. Each offer is identified by Offer\_id.

# **Data-Dictionary:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entity Name | Entity Description | | | | |
| Users | Users register in the system and use the services and and give reviews. This entity which represents one to many relationship with Services and Reviews and Rating. | | | | |
| Field Name | Description | Datatype | Key | Constraints | Example |
| User\_ID | Unique ID for each user | Autonumber | PK | Unique, not null | 001 |
| Name | Name of User | Text | - | Not Null | Vaidik |
| Address | Address of User | Text | - | Not Null | 350, Queen Street, Auckland CBD |
| Mobile No. | Mobile No./ Contact No. of User | Number | - | Not Null | 022 3434398 |
| Email | Email address of User | Text | - | Not Null | abcd@gmail.com |
| Password | Password of User | Text | - | Not Null | abcd@123 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entity Name | Entity Description | | | | |
| Reviews and Rating | Users register in the system and use the services and and give reviews. This table is entity which represents relationship between Users and Businesses. | | | | |
| Field Name | Description | Datatype | Key | Constraints | Example |
| Review\_ID | Unique ID for each review | Autonumber | PK | Unique, not null | 001 |
| User\_ID | ID of User | Number | FK | Not Null | 045 |
| Service\_ID | ID of Business | Number | FK | Not Null | 020 |
| Review Description | Review given by user | Text | - | - | Good service. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entity Name | Entity Description | | | | |
| Customer Service | This entity represents the associative entity between user and services. | | | | |
| Field Name | Description | Datatype | Key | Constraints | Example |
| User\_ID | ID of users | Number | FK | Not null | 001 |
| Service\_ID | ID of Business | Number | FK | Not Null | 020 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entity Name | Entity Description | | | | |
| Business Owner Services | This entity represents associative entity between business and services. | | | | |
| Field Name | Description | Datatype | Key | Constraints | Example |
| Business ID | ID of Business | Number | FK | Not null | 001 |
| Service\_ID | ID of Business | Number | FK | Not Null | 020 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entity Name | Entity Description | | | | |
| Deals and Offers | Business provides offers and deals for users. This table is entity which represents many to one relationship with businesses. | | | | |
| Field Name | Description | Datatype | Key | Constraints | Example |
| Offer\_ID | Unique ID for each offer | Autonumber | PK | Unique, not null | 001 |
| Service\_ID | ID of Business | Number | FK | Not Null | 020 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entity Name | Entity Description | | | | |
| Services | Business provides provide for users. This table is entity which represents many to one relationship with businesses. | | | | |
| Field Name | Description | Datatype | Key | Constraints | Example |
| Service\_ID | Unique ID for each service | Autonumber | PK | Unique, not null | 001 |
| Service Type | Name of Service | Text | - | Not Null | Food & Drinks |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entity Name | Entity Description | | | | |
| Businesses | Businesses register in the system and create the services and and get reviews. This table is entity which represents relationship between Services , Reviews&Rating and adviertisement & promotion. | | | | |
| Field Name | Description | Datatype | Key | Constraints | Example |
| Business\_ID | Unique ID for each Business | Autonumber | PK | Unique, not null | 001 |
| Name | Name of Business | Text | - | Not Null | ABC Enterprise |
| Address | Address of Business | Text | - | Not Null | 350, Queen Street, Auckland CBD |
| Mobile No. | Mobile No./ Contact No. of Business | INT | - | Not Null | 022 3434398 |
| Email | Email address of Business | Text | - | Not Null | abcd@gmail.com |
| Website | Website link of Business | Text | - | Not Null | www.abcd.com |
| Password | Password of Business | Text | - | Not Null | abcd@123 |

# **Git Repository:**

https://github.com/vaidik100/Dialdeal.git

# **Contributions:**

**Members:**

Abhishek Sharma: ER diagram, Usescases, Database& Queries

Vaidik Kakadiya: ER diagram, Usescases, Database & Queries

Data-Dictionary, Usecase Diagram

Harikash: ER diagram, Usescases, Database & Queries

Lovepreet Singh: ER diagram, Usescases, Data-Dictionary, Database & Queries

Abhishek Choudhary: ER diagram, Usescases, Database & Queries

THANK YOU!