Restaurant Management System (RMS) Application for Hospitality

In this Case Study we are going to develop a customized “Restaurant Management System” application using Force.com platform.

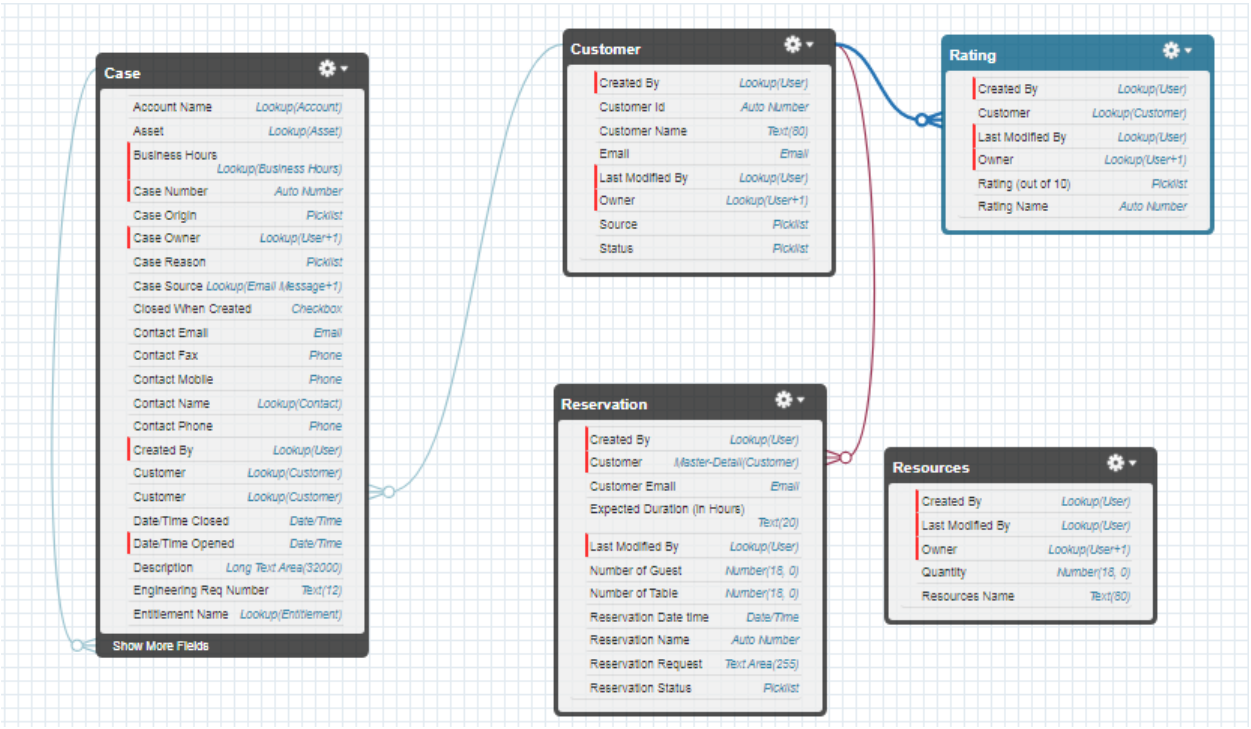
This application is supposed to provide solutions to automate complete process of a Restaurant Management System following processes to reduce manual efforts and time.

- **Different Entities would be**: Customer, Reservation, Cases, Ratings, Resources
- **Detailed Description regarding different entities/objects:** -
    - **Customer:** Customer who is visiting to Restaurant
    - **Reservation:** Customers can reserve in advance.
    - **Ratings:** Feedback about services.
    - **Resources:** To Track details about resources of Restaurant.
    - **Cases:** To Track any issue related to Customer/Restaurant.
- **Object Wise Fields are:** -
    - **Customer:** Name, Source, CustomerId, email, Status
    - **Reservation:** Reservation Name (Auto Number), Reservation Date &Time, Expected Duration, Reservation Requests (Text Area), Customer Email, Reservation Status (Draft/Approved/Rejected), Number of Guest (Number), Number of Table (Number), Customer (MD)
    - **Resource:** Resource Name (Text) e.g., Table (Setting arrangement), Quantity (Number)
        - ->The purpose of this object is to manage the sitting arrangement of the Restaurant.
    - **Ratings: Rating Name (Auto Number),** Rating (out of 10), Customer (LKP)
    - **Cases(standard):** Case Origin, Description, Customer (lookup on Customer)

Restaurant Management System (RMS) Application for Hospitality

Relationships among Objects as below:

ERD:



Assumptions/Details

- Possible values for **Sources**: **-**
    - **Marketing:** Social Media, Newspaper
    - **Non-Marketing:** Direct, Referral.
- **OWD would be Private by default unless stated for All Object.**
- Possible values for **Status** are: Active and Inactive
- Users using our application have Profile similar to 'Cloned System Admin Profile i.e., Helpdesk Profile' for Helpdesk User with below permissions: -

    Customer: Read, Create

Restaurant Management System (RMS) Application for Hospitality

- **Relationship between different combinations to be identified: -**
    - Customer & Reservation (One to Many)
    - Customer & Rating (One to Many)
    - Customer & Case (One to Many)

- **Following Fields should be visible for below: -**
    - **Marketing:** Name, Source, Customer Id, email, Status
    - **Non-Marketing:** Name, Source, Customer Id, email, Status

Business Rules/Scenarios: -

Validation Rules: -

1. User should not be able to enter new **Customer** record without Email Id. – Using Validation

2. **Reservation** time can't be in past.

3. Status for **Customer** cannot be inactive while creating new record.

Workflow Rules: -

4. If no status entered while **Customer** creation, Assign status as active for **Customer**.

5. Send email to '**Customer** email' on **Reservation** creation: -> we have to perform 2 actions.

    a. **Customer** Email field update on **Reservation** from **Customer** Email Field

    b. Email Alert

Process Builder:

6. Whenever a Reservation is Approved then Resource record must be changed accordingly.

    e.g., if Reservation approved then reduce the quantity in resource, otherwise it should remain as a its.

Note: if Resources are not available then please don't access the Reservation.

Restaurant Management System (RMS) Application for Hospitality

7. Create a Dummy **Reservation** whenever a new **Customer** is created.

Flows: -

8. Create a mechanism where user can submit feedback on scale 0-10. (using Screen flow)

9. If Status of **Customer** is Active set source as direct (using record triggered)

Sharing Rules/Manual Sharing

10. Share **Customer** record with Mr. Rajesh<<**Helpdesk user**>> if source is newspaper.

11. Share All **Customer** with Mr. Rajesh<<**Helpdesk user**>> which are created by admin

Field History Tracking

12. All Changes on **Customer** should be tracked.

Assignment Rules

13. Case assignment should be as follows: -

- If Priority is High->**High Priority Queue**
- IF Priority is low->**Low Priority Queue**

Escalation Rules

14. Cases should be dealt as per below: -

- High Priority Cases should be closed in 2 Hours otherwise assigned to Admin
- Low priority cases should be closed in 4 Hours otherwise assigned to Admin

Approval process (Done with auto submission)

15. All Reservations where number of guests > 15 should be sent for approval first and once approved then status of Reservation should be set as approved, if Rejected then Status will be Rejected.

Note: All Reservations where number of guests <=15 will be manually handled based on resource availability.

Restaurant Management System (RMS) Application for Hospitality

Reports and dashboards

16. Information Required by Higher Management (create reports accordingly): -

- Create a Dashboard which would give quick view about **Customer** as per different sources.
- Create a dashboard which would give quick view about restaurants ratings.

Permission Set

17. One user Mr. Rajesh<<**Helpdesk user**>> having same profile needs edit access on **Customer** object

Data Loader

18. Perform all operations on **Customer** object: -
    a. Export
    b. Export All
    c. Insert
    d. Update
    e. Delete
    f. Upsert

Apex Batch & Schedulable:

19. Create a mid-night daily job to delete all Rejected Reservation.

20. Create a min-night daily job which will reset the quantity of the Resource to original value.

Note 1: All Reservation will be handled same day only. No commitment, Reservation will be on same day based on the resource availability.

Note 2: The Original value will depend on the Restaurant.

Web Services

21. Create a Webservices where you have to create a new record for **Customer** Object (**Customer**Id, **Customer**Name) from third party application using Workbench

Restaurant Management System (RMS) Application for Hospitality

LWC (Lightning Web Component)

22. Create a LWC Component where you have to search **Customer** details either by Name or email and then populate searched result

    a. Searched result must include following fields:
        a. CustomerId, Name, email, Status