

MealManager

Team-2

Chaithra Lakshmi Sathyanarayana (Student ID: 012447565)

Thaijasa Badrinath Vijendranath (Student ID: 012470835)

Vineela Velicheti (Student ID: 012445186)

INTRODUCTION

- Interface between restaurants, dieticians and customers
- Customers:
 - People looking for meal subscriptions - breakfast, lunch and dinner
 - Option to enroll for a personalized diet plan planned by a affiliated dietician of their choosing
- Restaurants:
 - Orders details known in advance
 - Can offer food for lower rates and still profit
- Dieticians: Provide personalized diet plans
- Admin: Creates restaurant and dieticians users
- NoSQL MongoDB is used for storing contact form data

CUSTOMER

Major functionalities of a customer:

Customer without diet plan:

- Register to the application (Enter details and make payment)
- Login to the application with credentials
- Select the meals based on meal plan
- Place the order for selected meal and specify pick-up time
- Rate items
- Logout of the application

Customer with a diet plan:

- Register to the application (Enter details and make payment)
- Login to the application
- Select the meals based on meal plan and diet plan
- Place the order for selected meal and specify pick-up time
- View and update diet plan progress
- Rate items
- Rate dieticians
- Logout of the application

RESTAURANT

Major functionalities of a restaurant:

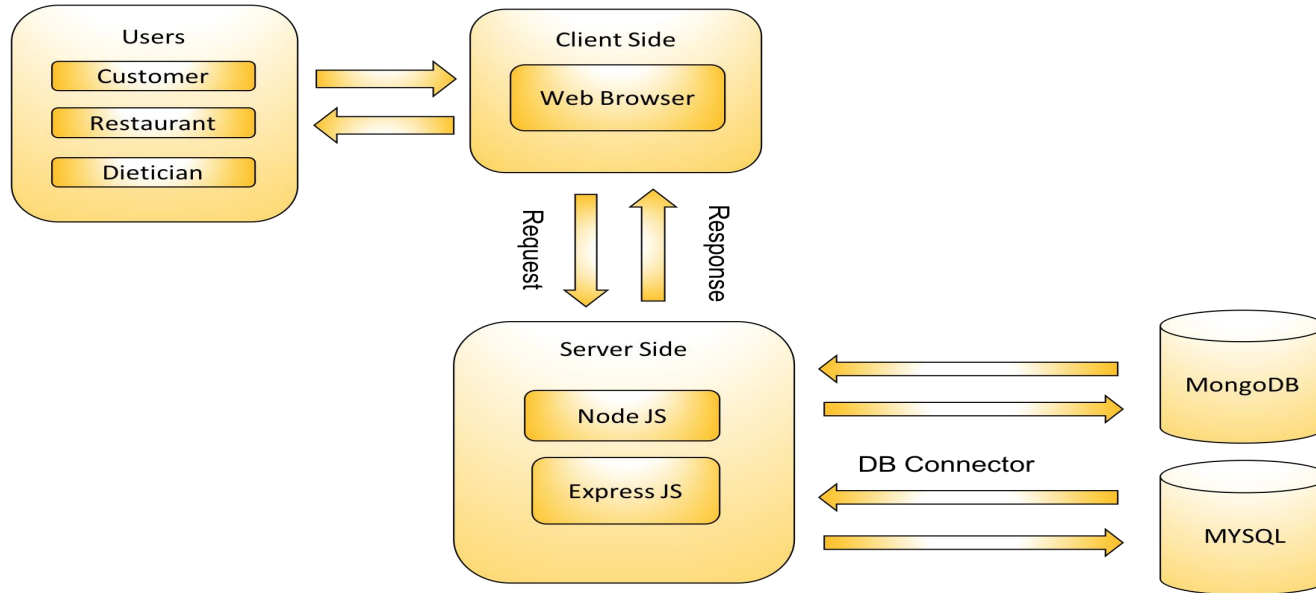
- Get registered to the application
- Login in to the application with credentials
- Add items with nutritional information to the menu
- Receive and view item orders from customers
- Update/delete items from menu
- Generate sales report for a specific time period
- Log out of the application

DIETICIAN

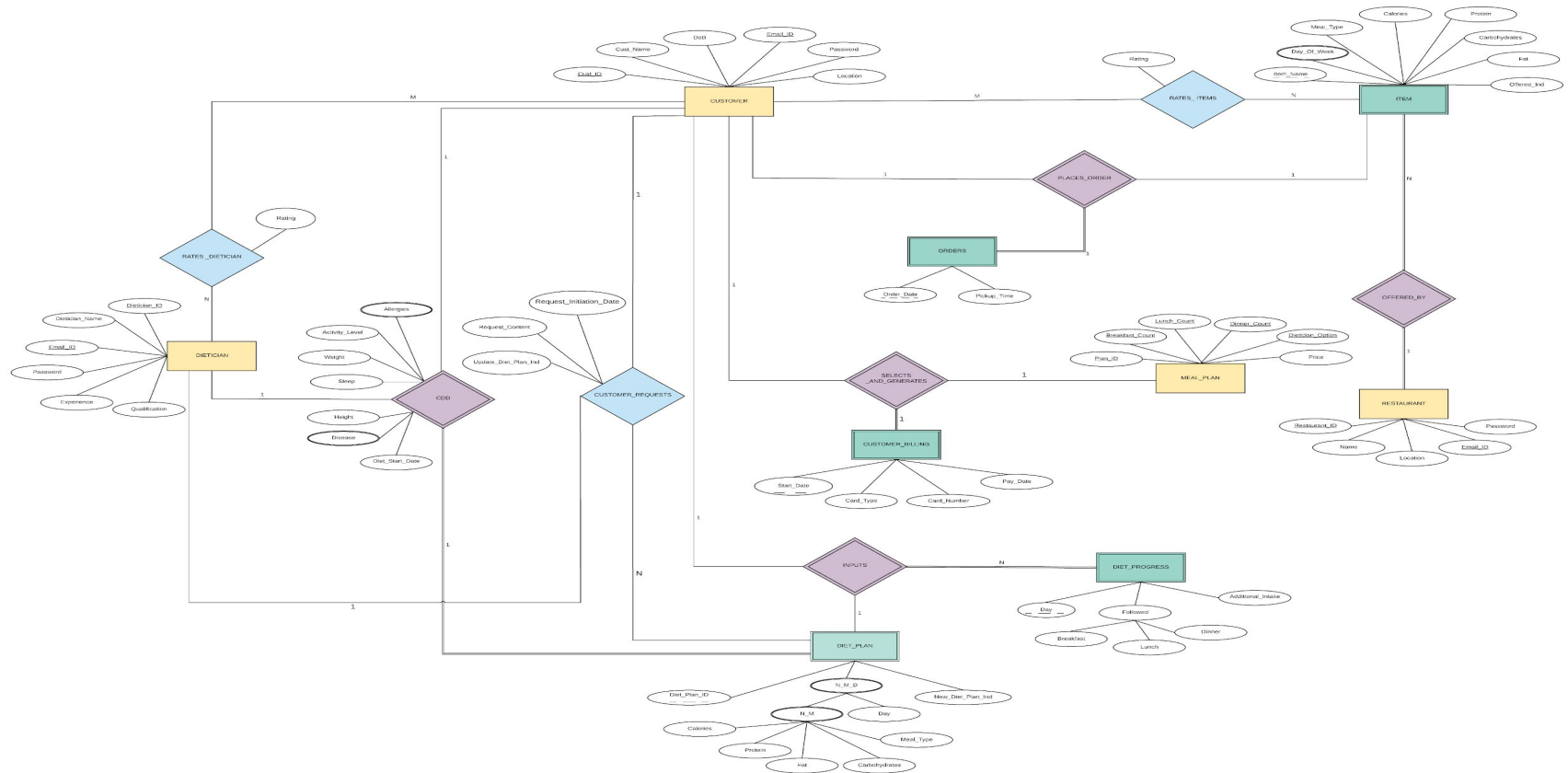
Major functionalities of a dietician:

- Get registered to the application
- Login in to the application with credentials
- Receive diet plan requests from customer
- Formulate diet plan for the requests and update them in the application
- View enrolled customer's diet progress
- Cater to change requests orders from the customer
- Logout of the application

ARCHITECTURE



ER DIAGRAM



TRIGGER

```
1
2 • DROP TRIGGER IF EXISTS update_offered_ind_for_insert;
3
4 DELIMITER $$
5 • CREATE TRIGGER update_offered_ind_for_insert
6 BEFORE INSERT
7 ON restaurant_item_day
8 FOR EACH ROW
9 BEGIN
10     DECLARE v_count INT;
11     SELECT COUNT(*) INTO v_count FROM restaurant_item_day
12     WHERE restaurant_id = NEW.restaurant_id AND item_name = NEW.item_name;
13     IF v_count > 0 THEN
14         UPDATE Item
15         SET Offered_Ind = 'Y'
16         WHERE restaurant_id = NEW.restaurant_id AND item_name = NEW.item_name;
17     ELSEIF v_count = 0 THEN
18         UPDATE Item
19         SET Offered_Ind = 'N'
20         WHERE restaurant_id = NEW.restaurant_id AND item_name = NEW.item_name;
21     END IF;
22 END;
23 $$
24 DELIMITER ;
25
```


STORED PROCEDURE

```
375
376 DELIMITER $$
377 • CREATE PROCEDURE c_update_item_rating
378   (IN p_cust_id int, IN p_restaurant_id int, IN p_item_name varchar(30), IN p_rating int, OUT v_chk int )
379 BEGIN
380   DECLARE `_rollback` BOOL DEFAULT 0;
381   DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET `_rollback` = 1;
382   START TRANSACTION;
383   SELECT COUNT(*) INTO v_chk
384   FROM item_ratings
385   WHERE cust_id = p_cust_id AND restaurant_id = p_restaurant_id AND item_name= p_item_name;
386   IF v_chk != 1 THEN
387     INSERT INTO item_ratings VALUES(p_cust_id,p_restaurant_id,p_item_name,p_rating);
388   ELSE
389     UPDATE item_ratings
390     SET rating = p_rating
391     WHERE cust_id = p_cust_id AND restaurant_id = p_restaurant_id AND item_name= p_item_name;
392   END IF;
393   IF `_rollback` THEN ROLLBACK;
394   ELSE COMMIT;
395   END IF;
396 END;
397 $$
398 DELIMITER ;
```

VIEW

```
1
2 • CREATE OR REPLACE VIEW d_customer_diet_plan_details_view AS
3 SELECT
4     dp.cust_id,
5     dp.dietician_id,
6     dp.diet_start_date,
7     mt.day,
8     DATE_ADD(dp.diet_start_date,
9     INTERVAL mt.day DAY) AS calendar_date,
10    dp.new_diet_plan_ind,
11    mt.meal_type,
12    dpd.calories,
13    dpd.proteins,
14    dpd.carbohydrates,
15    dpd.fat
16 FROM
17     diet_plan dp
18     JOIN
19     diet_plan_day_meal_type_view mt
20     LEFT JOIN
21     diet_plan_details dpd ON dp.cust_id = dpd.cust_id
22     AND dpd.diet_plan_id = dp.diet_plan_id
23     AND dpd.dietician_id = dp.dietician_id
24     AND mt.day = dpd.day
25     AND mt.meal_type = dpd.meal_type
26 ;
```

DB OBJECTS

DB Object	Total Count
Table	18
Index	1
Trigger	2
Views	20
Stored Procedures	20

NoSQL: MONGODB for Contact Form

```
> db.contactform.find().pretty();
{
  "_id" : ObjectId("5ccfb7dda87c6fcdaccf2283"),
  "name" : "mongo1@gmail.com",
  "message" : [
    [
      "This is message1!",
      "5/5/2019, 9:28:13 PM"
    ],
    [
      "This is message2",
      "5/5/2019, 9:28:31 PM"
    ],
    [
      "Please contact me on 98789373 on sundays between 9am and 10am.",
      "5/5/2019, 9:28:49 PM"
    ]
  ],
  "update_timestamp" : "5/5/2019, 9:28:49 PM"
}
{
  "_id" : ObjectId("5ccfb839a87c6fcdaccf2324"),
  "name" : "mongo2@gmail.com",
  "message" : [
    [
      "Hello, \r\nI would like to discuss business opportunities. Please call me on monday.",
      "5/5/2019, 9:29:45 PM"
    ]
  ],
  "update_timestamp" : "5/5/2019, 9:29:45 PM"
}
> |
```

```
3335
3336
3337 function func_updatecontactform(request,result)
3338 {
3339     var contact_email_id = request.body.email_id;
3340     var contact_message = request.body.contact_message;
3341     var MongoClient = require('mongodb').MongoClient;
3342     var url = "mongodb://localhost:27017/";
3343     MongoClient.connect(url, { useNewUrlParser: true }, function(err, db) {
3344         if (err) throw err;
3345         var dbo = db.db("mealmanager");
3346         var contact_message_time=new Date().toLocaleString();
3347         console.log(contact_message_time);
3348         dbo.collection("contactform").updateOne({name:contact_email_id},
3349 { $addToSet: {message:[ contact_message,contact_message_time]} , $set: { update_timestamp: contact_message_time} },
3350 { upsert: true }, function(err, res) {
3351     if (err) {
3352         throw err;
3353     }
3354     else{
3355         log.info('Sending success message for contact form');
3356         result.render('contactform',{cust_message:"Your message has been conveyed. You can add more messages with the same contact id."});
3357     }
3358     //console.log("Number of documents inserted: " + res.insertedCount);
3359 });
3360 });
3361
3362
3363
3364
3365
3366 module.exports.update_contactform = function(request, result)
3367 {
3368     log.info('Recieved request to update contact form');
3369     func_updatecontactform(request,result);
3370     console.log(request.body);
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
4000
```

RETROSPECTIVE

What we did well?

- Implementation and integration from frontend to backend of the application
- Team work and communication

What we learned new?

- Understand forming complex SQL stored procedures
- Using session concepts in application and generating logs during application run

What we can improve on?

- Gathering and understanding business requirements

Demo Time!

Thank You!