

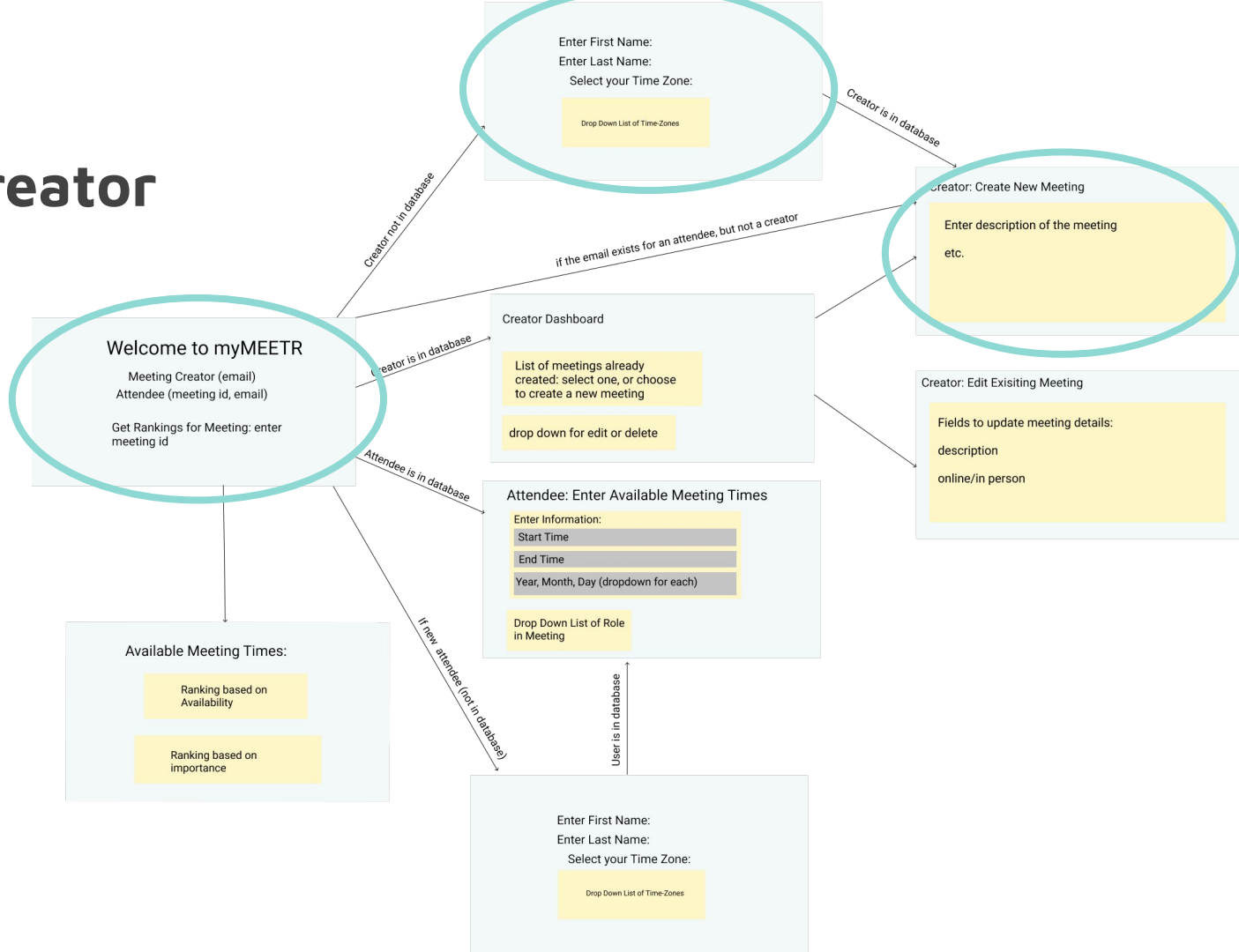
# myMEETR

Eesha Deepak, Kristy Huang, Sweta  
Saravanan, Rashmi Ananth



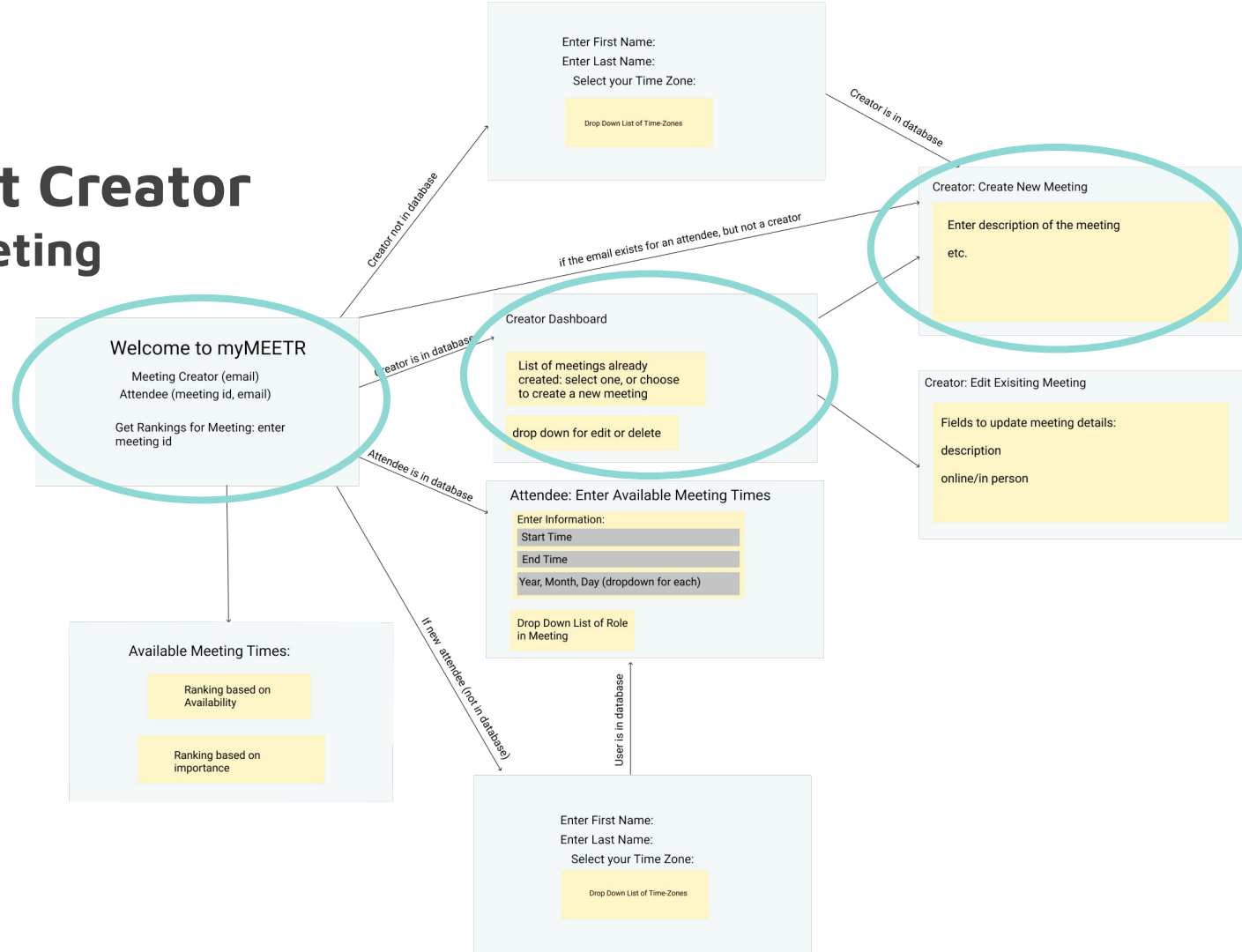


# New Creator

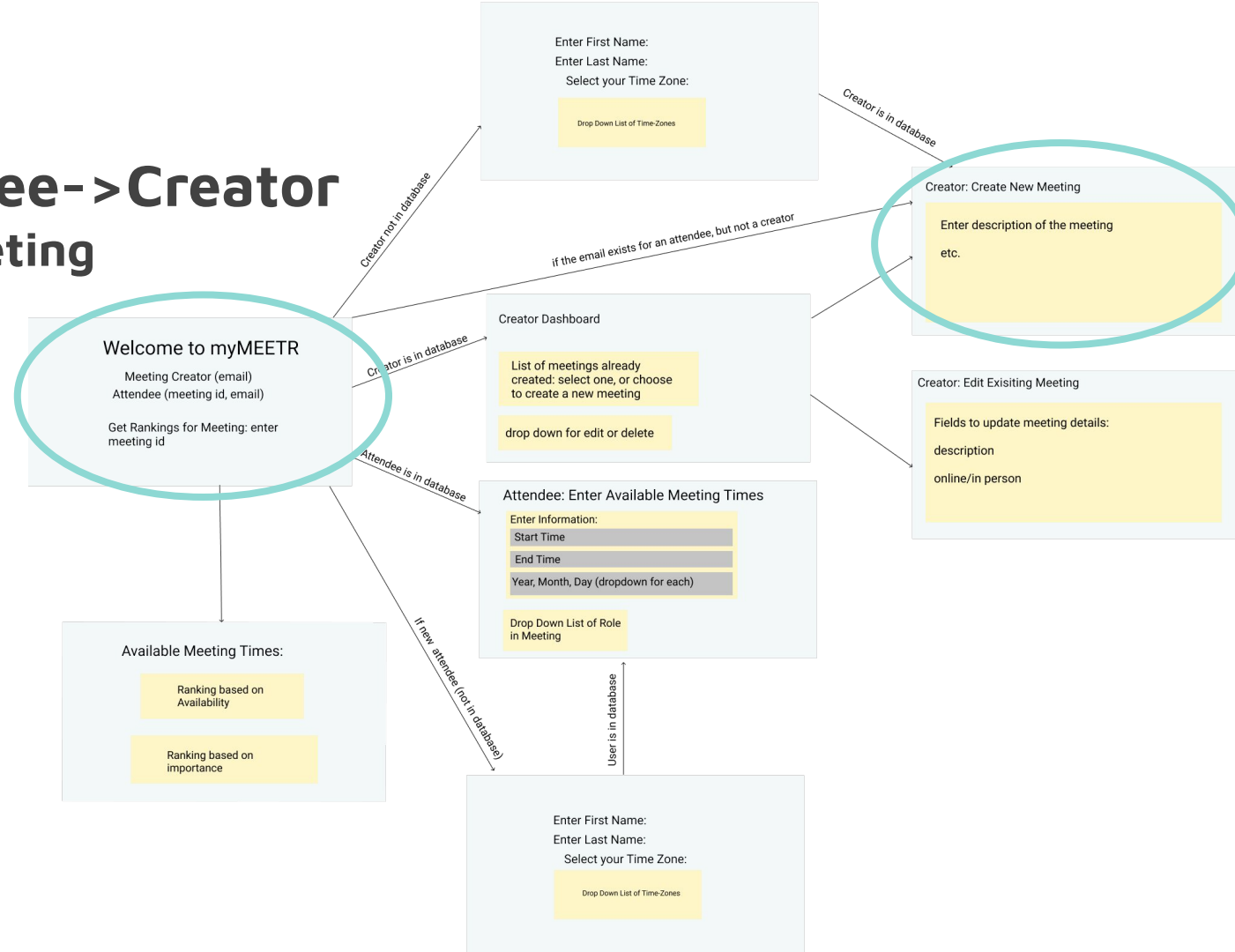


# Current Creator

## New Meeting

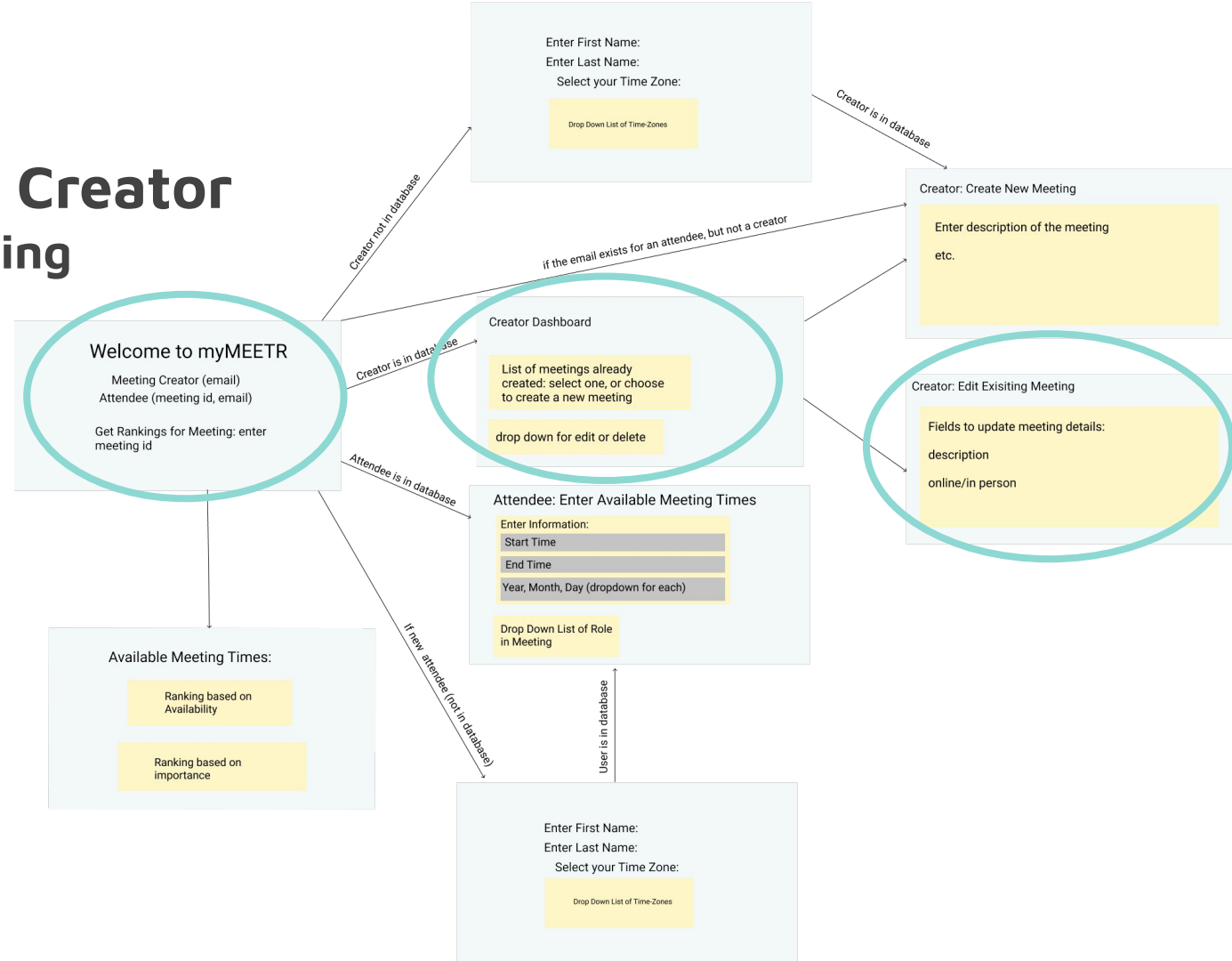


# Attendee->Creator New Meeting

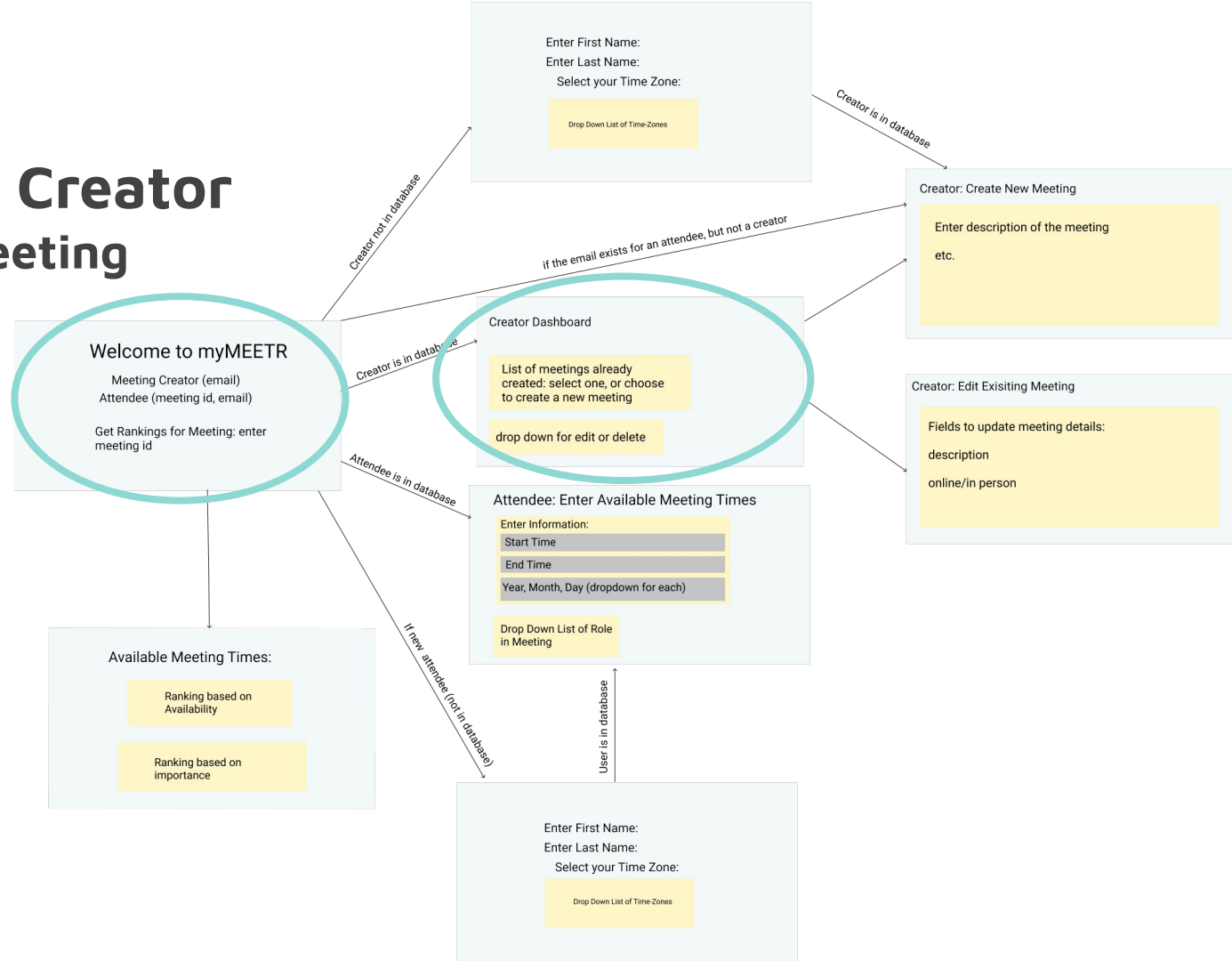


# Current Creator

## Edit Meeting

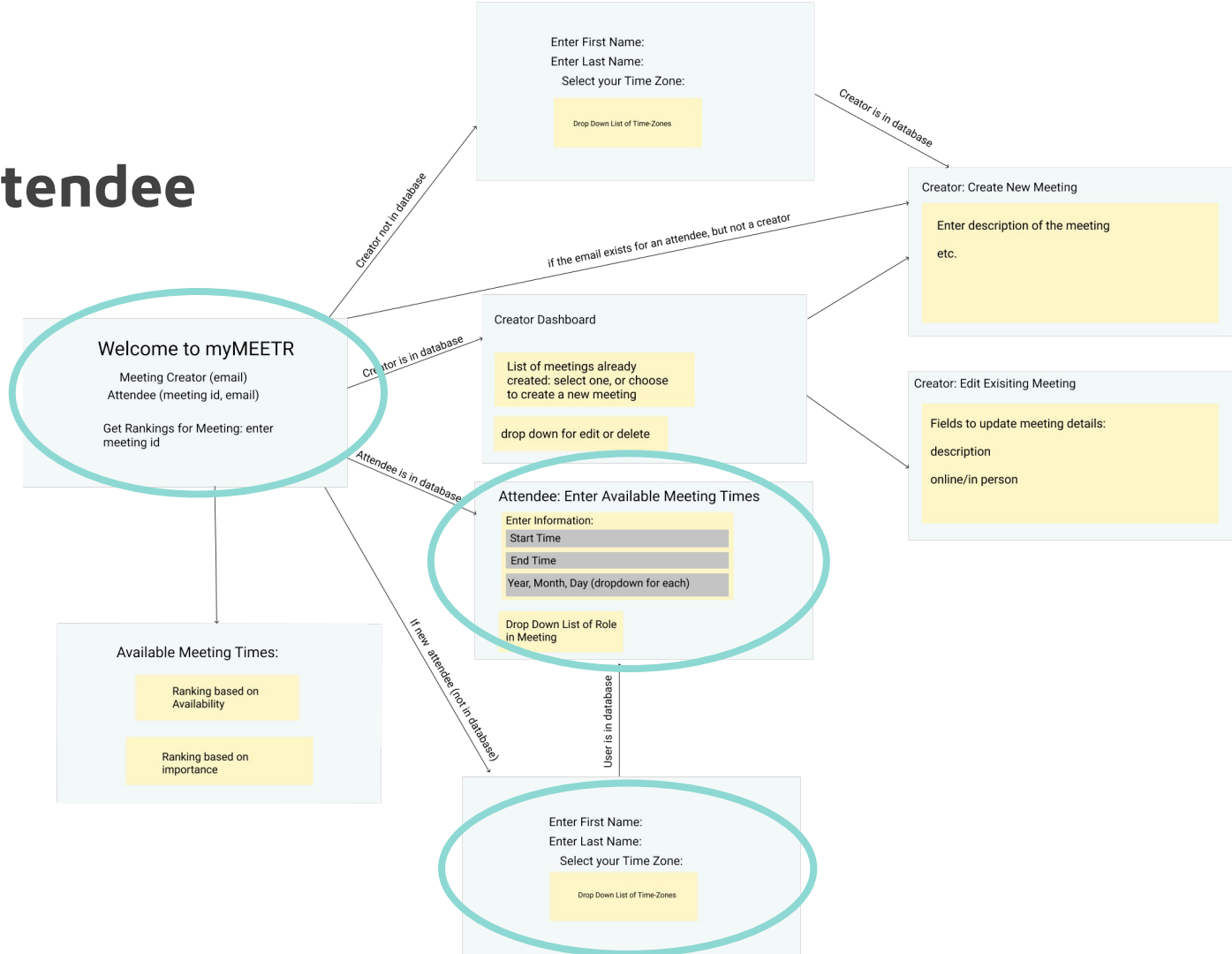


# Current Creator Delete Meeting



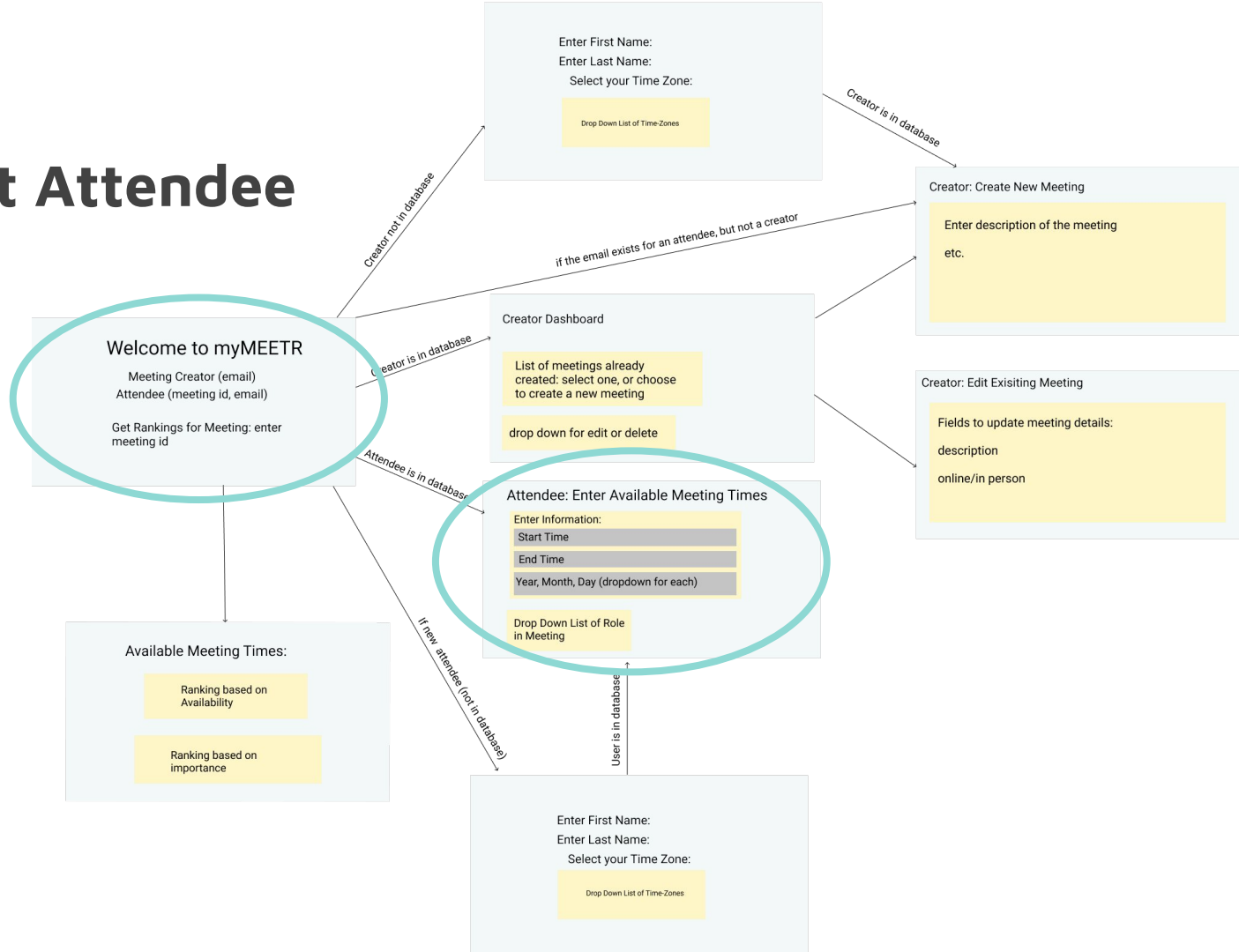


# New Attendee





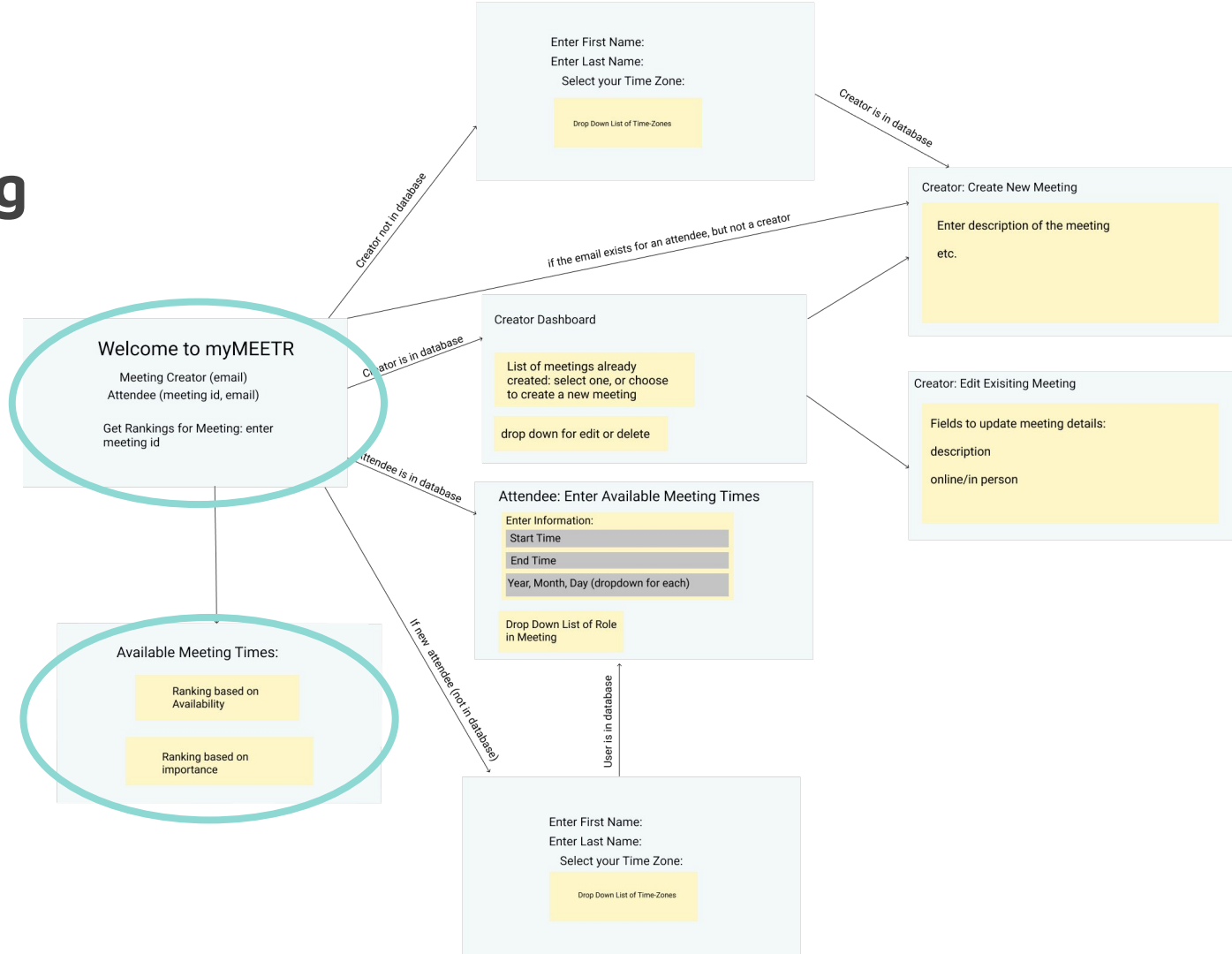
# Current Attendee







# Ranking





# Ranking - Complex Queries (1) + Prepared Statements

```
536 cnx = mysql.connector.connect(user=config.user, password=config.password, host=config.host, database=config.db)
537 cursor = cnx.cursor(prepared=True)
538 try:
539     query = """
540         select date, start_time, end_time, count(person_id) as people_available
541         from
542             (select link_meeting.availability_id, person.person_id, date, start_time, end_time from availability_info, person, link_meeting
543              where link_meeting.person_id = person.person_id and link_meeting.availability_id = availability_info.availability_id and
544                link_meeting.meeting_id = %s)
545             as table_times
546         group by availability_id
547         order by people_available desc;"""
548     cursor.execute(query, (R_meeting_id, ))
549 except mysql.connector.Error as err:
550     print(err)
551
552 data = cursor.fetchall()
553 cursor.close()
554 cnx.close()
```

# Ranking - Complex Queries (2) + Prepared Statements

```
554 cnx2 = mysql.connector.connect(user=config.user, password=config.password, host=config.host, database=config.db)
555 cursor2 = cnx2.cursor(prepared=True)
556 try:
557     query2 = """
558         select availability_id, level_1, level_2, level_3, level_4, level_5, (level_1 + level_2 + level_3 + level_4 + level_5) as total, date,
559         start_time, end_time
560         from
561             (select availability_id, sum(importance_level = 1) as level_1, sum(importance_level = 2) as level_2, sum(importance_level = 3) as
562             level_3, sum(importance_level = 4) as level_4, sum(importance_level = 5) as level_5, date, start_time, end_time
563             from
564                 (select link_meeting.meeting_id, link_meeting.availability_id, person.person_id, importance.importance_level, date, start_time,
565                 end_time
566                 from availability_info, link_meeting, importance, attendee_info, person
567                 where link_meeting.person_id = person.person_id
568                 and link_meeting.availability_id = availability_info.availability_id
569                 and link_meeting.person_id = attendee_info.person_id
570                 and attendee_info.meeting_id = link_meeting.meeting_id
571                 and link_meeting.meeting_id = %s
572                 and importance.meeting_role = attendee_info.meeting_role) as tables
573             group by availability_id
574             order by level_1 desc, level_2 desc, level_3 desc, level_4 desc, level_5 desc) as level_times;"""
575     cursor2.execute(query2, (R_meeting_id, ))
576 except mysql.connector.Error as err:
577     print(err)
578
579 data2 = cursor2.fetchall()
580 cursor2.close()
581 cnx2.close()
```



# Welcome Page - Prepared Statements

Prepared Statements: (153)

```
cnx = mysql.connector.connect(user=config.user, password=config.password, host=config.host, database=config.db)
cursor = cnx.cursor(prepared=True)
query = """
    SELECT p.email
    FROM person p
    JOIN meeting_details m ON p.person_id = m.creator_id
    WHERE p.email = %s
    LIMIT 1;"""
cursor.execute(query, (M_email, ))
mexists = cursor.fetchall()
cursor.close()
cnx.close()
```



# Welcome Page - ORM

ORM: (132, 135, 166, 172, 178, 192)

```
#check if attendee email exists
```

```
pexists = db.session.query(db.exists().where(person.email == A_email)).scalar()
```



# New Attendee Page - ORM

ORM: (221)

```
pers = person(request.form['first_name'], request.form['last_name'], request.form['time_zone_name'], request.form['email'])  
  
db.session.add(pers)  
db.session.commit()
```



# Availability Page - Prepared Statements

Prepared Statements: (257, 277, 339, 399, 462, 485, 517)

```
cnx2 = mysql.connector.connect(user=config.user, password=config.password, host=config.host, database=config.db)
cursor2 = cnx2.cursor(prepared=True)
query2 = """
    select meeting_role
    from attendee_info, person
    where person.email = %s
    and attendee_info.person_id = person.person_id
    and attendee_info.meeting_id = %s;"""
cursor2.execute(query2, (A_email, meeting_id, ))
data2 = cursor2.fetchall()
cursor2.close()
cnx2.close()
```



## Availability Page - ORM

ORM: (478, 499, 505)

```
if len(data4)==0:
    ai = availability_info(full_date, start_time, end_time)
    db.session.add(ai)
    db.session.commit()
    av_id = ai.id
```





# New Creator Page - ORM

ORM: (236)

```
pers = person(request.form['first_name'], request.form['last_name'], request.form['time_zone_name'], request.form['email'])  
  
db.session.add(pers)  
db.session.commit()
```



# Creator Dashboard Page - Prepared Statements

Prepared Statements: (642, 657, 670, 693)

```
cnx9 = mysql.connector.connect(user=config.user, password=config.password, host=config.host, database=config.db)
cursor9 = cnx9.cursor(prepared=True)
query9 = """
    select person_id
    from meeting_details, person
    where person.email = %s
    and person.person_id = meeting_details.creator_id;
    """
cursor9.execute(query9, (M_email,))
data9 = cursor9.fetchall()
cursor9.close()
cnx9.close()
```



# Create Meeting Page - Prepared Statements

Prepared Statements: (713)

```
cnx9 = mysql.connector.connect(user=config.user, password=config.password, host=config.host, database=config.db)
cursor9 = cnx9.cursor(prepared=True)
query9 = """
    select person_id from person order by person_id desc LIMIT 1;
    """
cursor9.execute(query9)
data9 = cursor9.fetchall()
cursor9.close()
cnx9.close()
```



# Create Meeting Page - ORM

ORM: (753)

```
newMeeting = meeting_details(inperson, online, start_day, end_day, request.form['length'], request.form['meeting_description'],  
w_creator_id)  
db.session.add(newMeeting)  
db.session.commit()
```



# Edit Existing Meeting Page - ORM

ORM: (584, 597, 613, 624)

```
db.session.query(meeting_details).filter(meeting_details.meeting_id == creator_meeting_id).update(  
    {'in_person':in_person, 'online':online, 'description':description, 'creator_id':w_creator_id})  
db.session.commit()
```



# Transactions (1)

```
516 # make sure to not offer the role of coordinator
517 cnx6 = mysql.connector.connect(user=config.user, password=config.password, host=config.host, database=config.db)
518 cnx6.start_transaction(consistent_snapshot=bool,
519 | | | | isolation_level='SERIALIZABLE',
520 | | | | readonly=True)
521 cursor6 = cnx6.cursor(prepared=True)
522 query6 = """
523 | | | | select meeting_role from importance where importance_level != 1;"""
524 result = cursor6.execute(query6)
525 data6 = cursor6.fetchall()
526 cursor6.close()
527 cnx6.close()
```



## Transactions (2)

```
338 # get the person's id and the time_zone offset
339 cnx = mysql.connector.connect(user=config.user, password=config.password, host=config.host, database=config.db)
340 cnx.start_transaction(consistent_snapshot=bool,
341                       isolation_level='READ COMMITTED',
342                       readonly=True)
343 cursor = cnx.cursor(prepared=True)
344 query = """
345     select person.person_id, time_zone.offset
346     from person
347     JOIN time_zone on person.time_zone_name = time_zone.name
348     where person.email = %s;"""
349 cursor.execute(query, (A_email, ))
350
351 data = cursor.fetchall()
352 cursor.close()
353 cnx.close()
```



# Indices (1)

```
mysql> explain analyze select meeting_role from importance where importance_level != 1;
```

```
+-----+
| EXPLAIN
+-----+
|
+-----+
| -> Filter: (importance.importance_level <> 1)  (cost=0.75 rows=4) (actual time=0.023..0.026 rows=4 loops=1)
|   -> Table scan on importance  (cost=0.75 rows=5) (actual time=0.021..0.023 rows=5 loops=1)
|
+-----+
1 row in set (0.03 sec)
```

```
mysql> create index I3 on importance(importance_level);
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> explain analyze select meeting_role from importance where importance_level != 1;
```

```
+-----+
| EXPLAIN
+-----+
|
+-----+
| -> Filter: (importance.importance_level <> 1)  (cost=0.75 rows=5) (actual time=0.018..0.021 rows=4 loops=1)
|   -> Index scan on importance using I3  (cost=0.75 rows=5) (actual time=0.015..0.018 rows=5 loops=1)
|
+-----+
1 row in set (0.02 sec)
```





## Indices (2, 3)

```
mysql> create index I4 on meeting_details(creator_id);
```

```
Query OK, 0 rows affected (0.10 sec)
```

```
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> explain analyze select meeting_id, start_day, end_day, length_hr, description from meeting_details where creator_id = 6;
```

```
+-----+
| EXPLAIN                                                                                                     |
+-----+
| -> Index lookup on meeting_details using I4 (creator_id=6)  (cost=0.35 rows=1) (actual time=0.018..0.019 rows=1 loops=1)
|
+-----+
1 row in set (0.03 sec)
```

```
mysql> create index I6 on person(email);
```

```
Query OK, 0 rows affected (0.08 sec)
```

```
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> explain analyze SELECT p.email
```

```
->      FROM person p
->      JOIN meeting_details m ON p.person_id = m.creator_id
->      WHERE p.email = "jbaker@purdue.edu"
->      LIMIT 1;
+-----+
| EXPLAIN                                                                                                     |
+-----+
| -> Limit: 1 row(s)  (actual time=0.021..0.021 rows=1 loops=1)
|   -> Nested loop inner join  (cost=0.71 rows=1) (actual time=0.020..0.020 rows=1 loops=1)
|     -> Index lookup on p using I6 (email='jbaker@purdue.edu')  (cost=0.35 rows=1) (actual time=0.014..0.014 rows=1 loops=1)
|     -> Index lookup on m using I4 (creator_id=p.person_id)  (cost=0.36 rows=1) (actual time=0.005..0.005 rows=1 loops=1)
|
+-----+
1 row in set (0.03 sec)
```



# Lessons learned + how we can improve

- Have a thought out conceptual design
- Importance of accurate SQL file (made further steps a lot easier)
- Ensure that any additions made to our implementations work with the previously implemented code
  - Occasionally ran up on errors on code that previously worked
- Compartmentalizing code and putting code in separate files makes it easier to collaborate and reduces merge conflicts

**Thank You!**