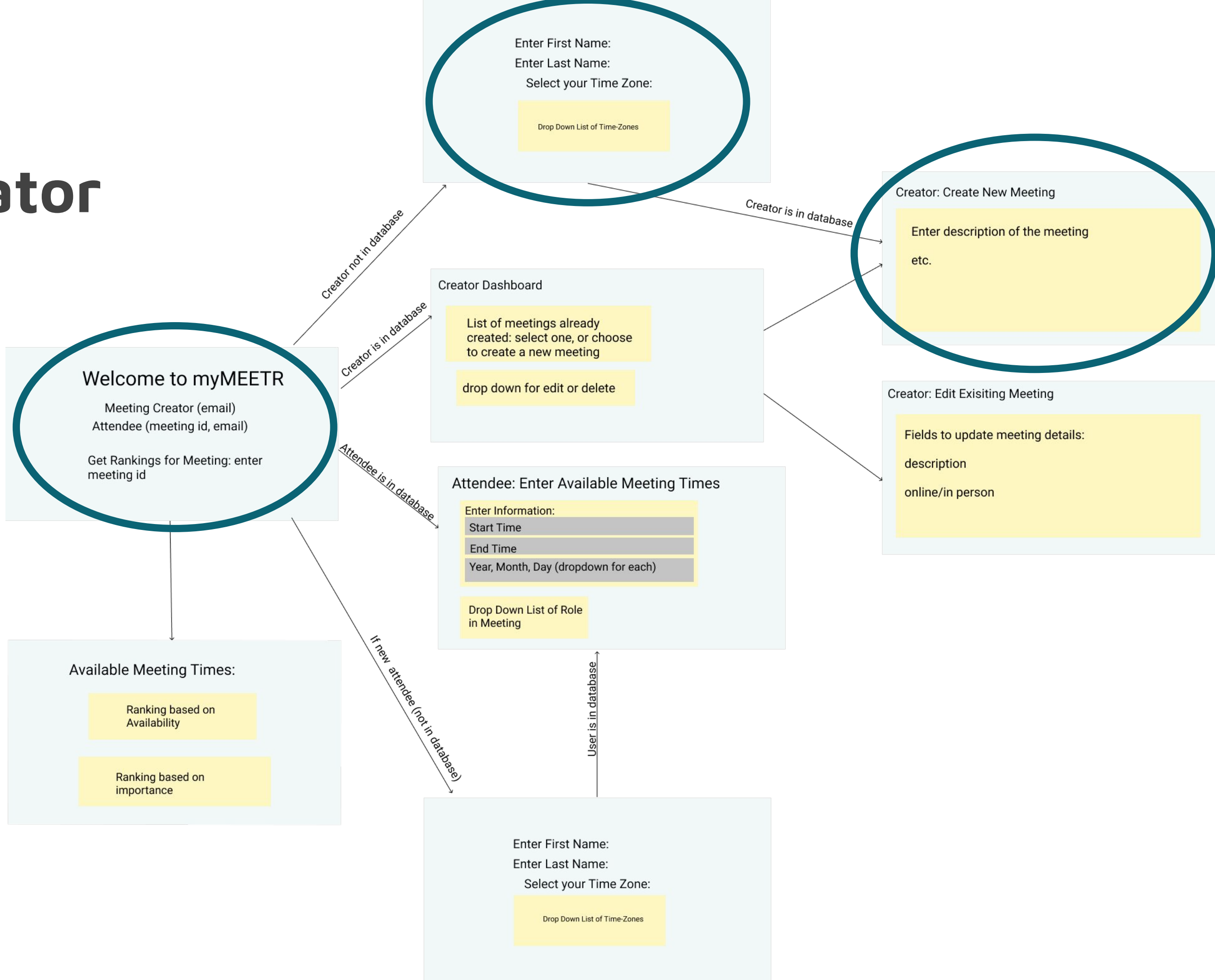
myMEETR

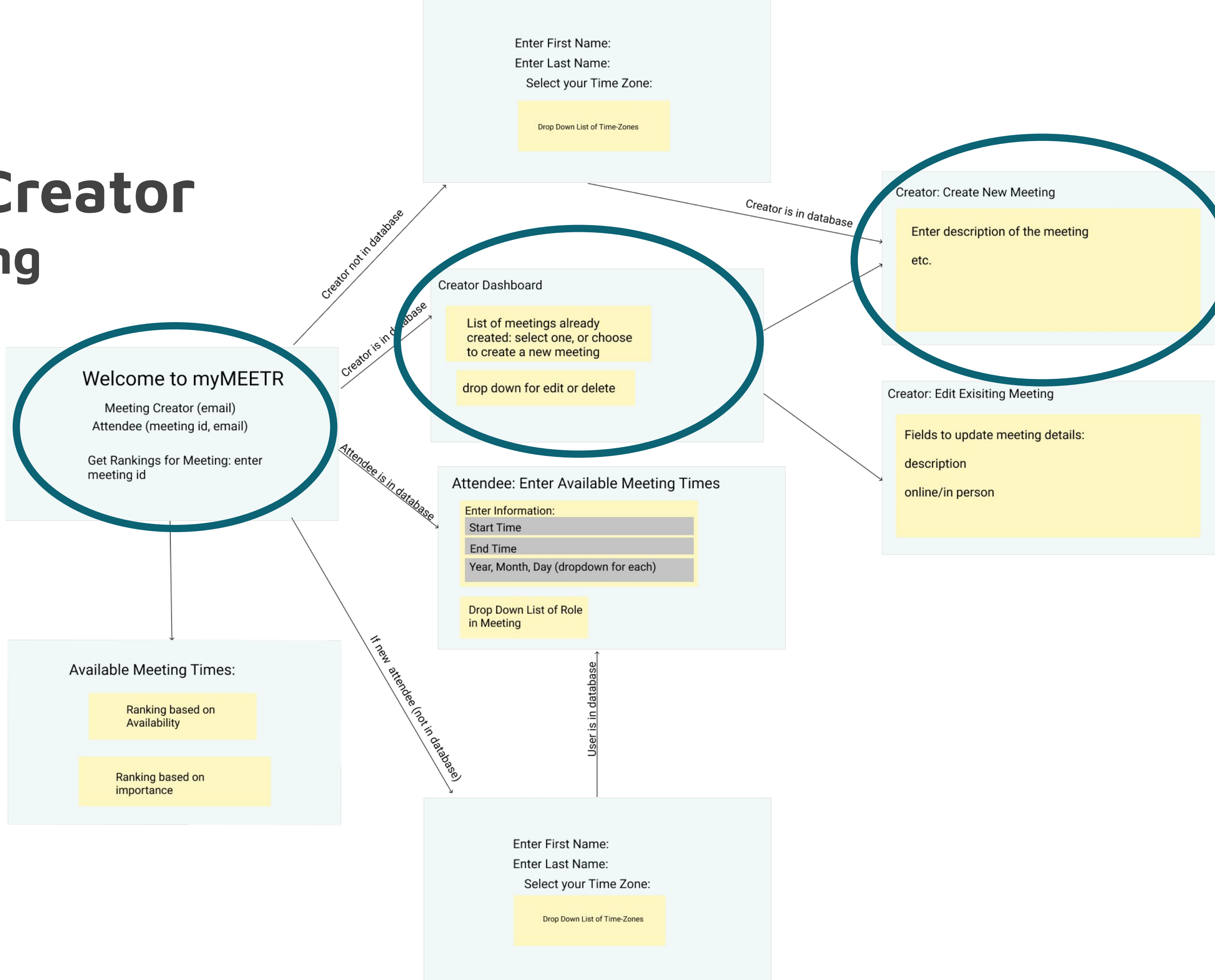
Eesha Deepak, Kristy Huang, Sweta Saravanan, Rashmi Ananth

New Creator



Welcome to myMEETR
Meeting Creator (email)
Attendee (meeting id, email)
Get Rankings for Meeting: enter meeting id
Creator not in database
Enter First Name:
Enter Last Name:
Select your Time Zone:
Drop Down List of Time-Zones
Creator is in database
Creator: Create New Meeting
Enter description of the meeting
etc.
Creator is in database
Creator Dashboard
List of meetings already created: select one, or choose to create a new meeting
drop down for edit or delete
Creator: Edit Exisiting Meeting
Fields to update meeting details:
description
online/in person
Attendee is in database
Attendee: Enter Available Meeting Times
Enter Information:
Start Time
End Time
Year, Month, Day (dropdown for each)
Drop Down List of Role in Meeting
User is in database
If new attendee (not in database)
Enter First Name:
Enter Last Name:
Select your Time Zone:
Drop Down List of Time-Zones
Available Meeting Times:
Ranking based on Availability
Ranking based on importance

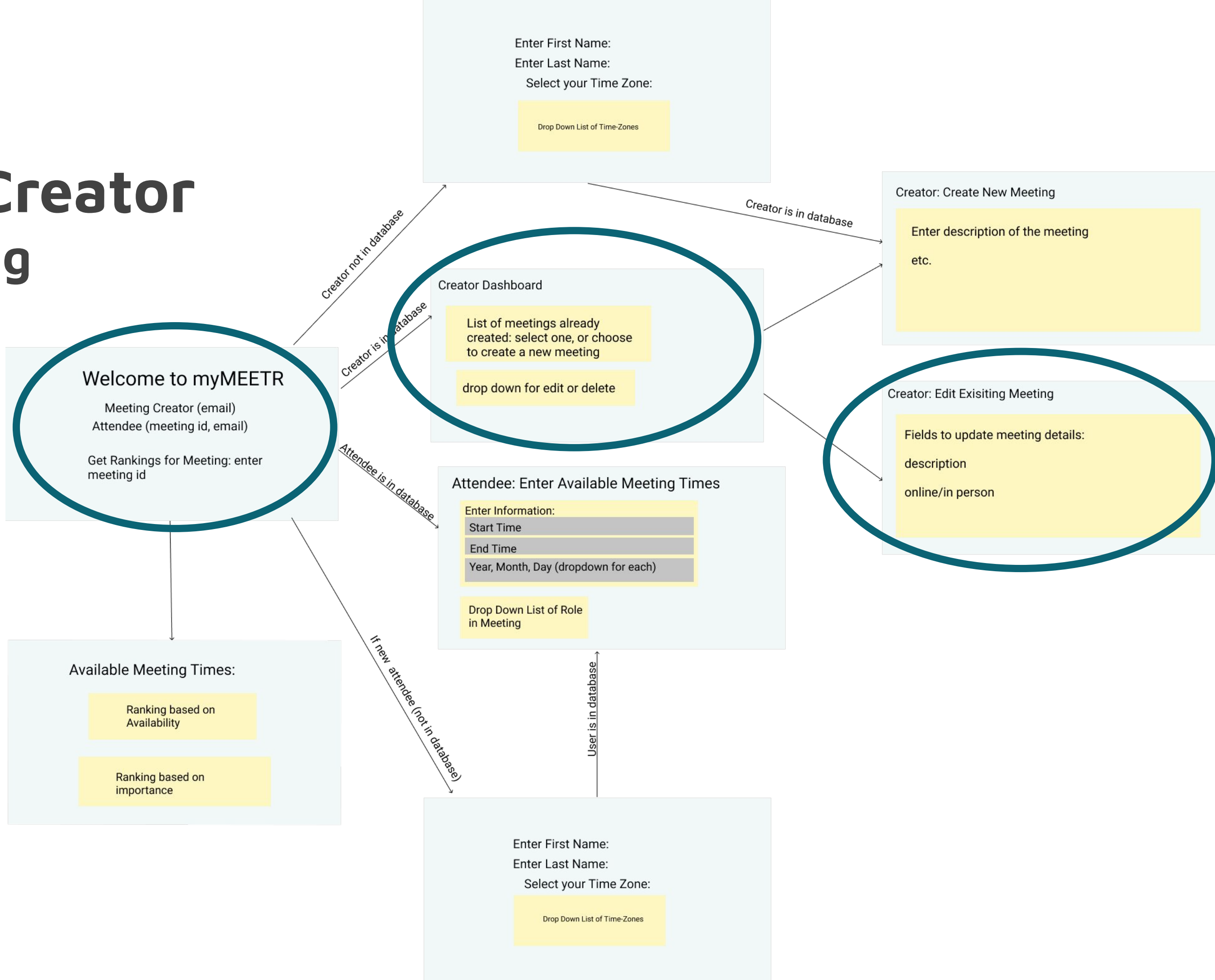
Current Creator
New Meeting



Enter First Name:
Enter Last Name:
Select your Time Zone:
Drop Down List of Time-Zones
Creator not in database
Creator is in database
Creator is in database
Creator: Create New Meeting
Enter description of the meeting
etc.
Creator Dashboard
List of meetings already created: select one, or choose to create a new meeting
drop down for edit or delete
Welcome to myMEETR
Meeting Creator (email)
Attendee (meeting id, email)
Get Rankings for Meeting: enter meeting id
Creator: Edit Exisiting Meeting
Fields to update meeting details:
description
online/in person
Attendee is in database
Attendee: Enter Available Meeting Times
Enter Information:
Start Time
End Time
Year, Month, Day (dropdown for each)
Drop Down List of Role in Meeting
Available Meeting Times:
Ranking based on Availability
Ranking based on importance
If new attendee (not in database)
User is in database
Enter First Name:
Enter Last Name:
Select your Time Zone:
Drop Down List of Time-Zones

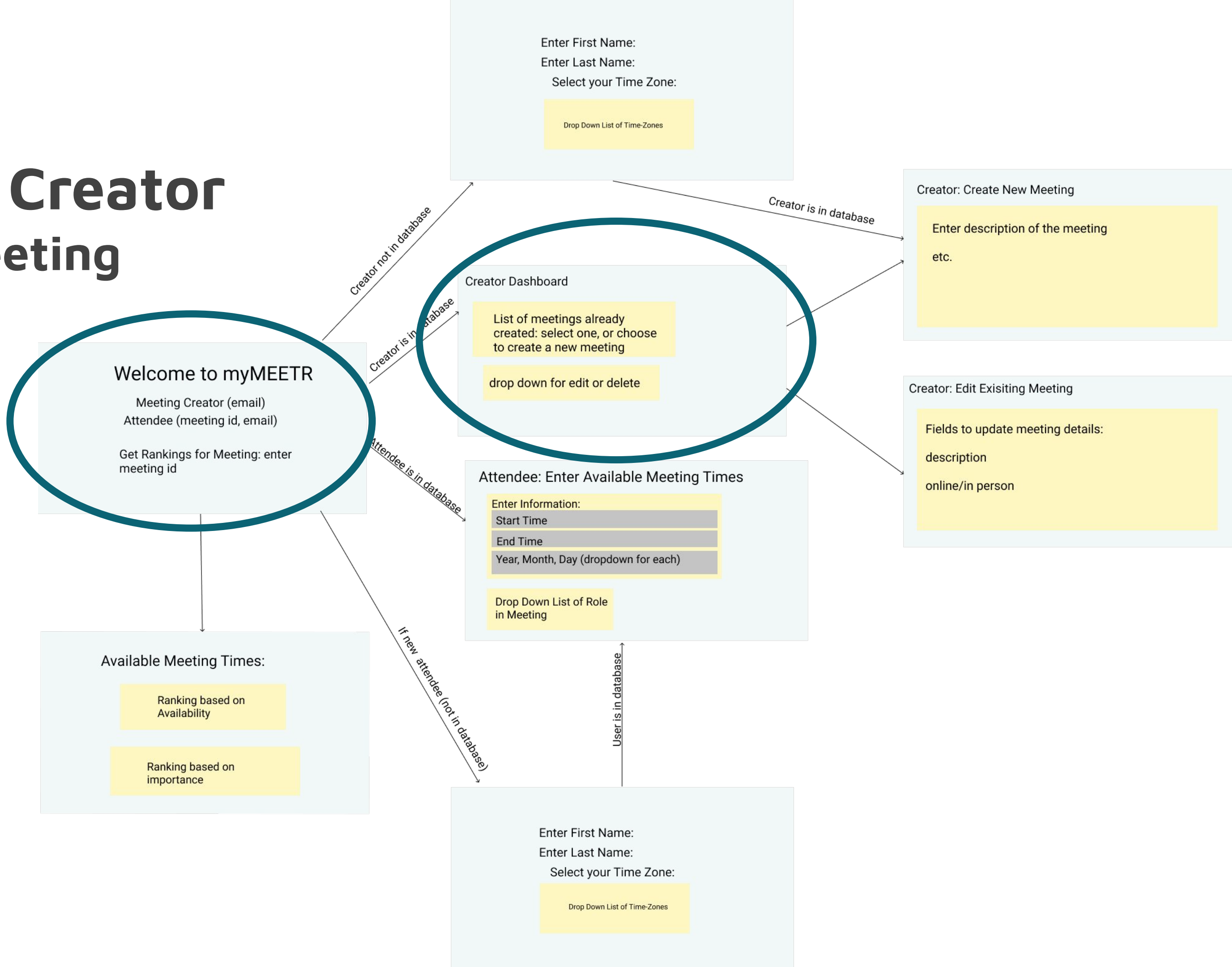
Current Creator

Edit Meeting

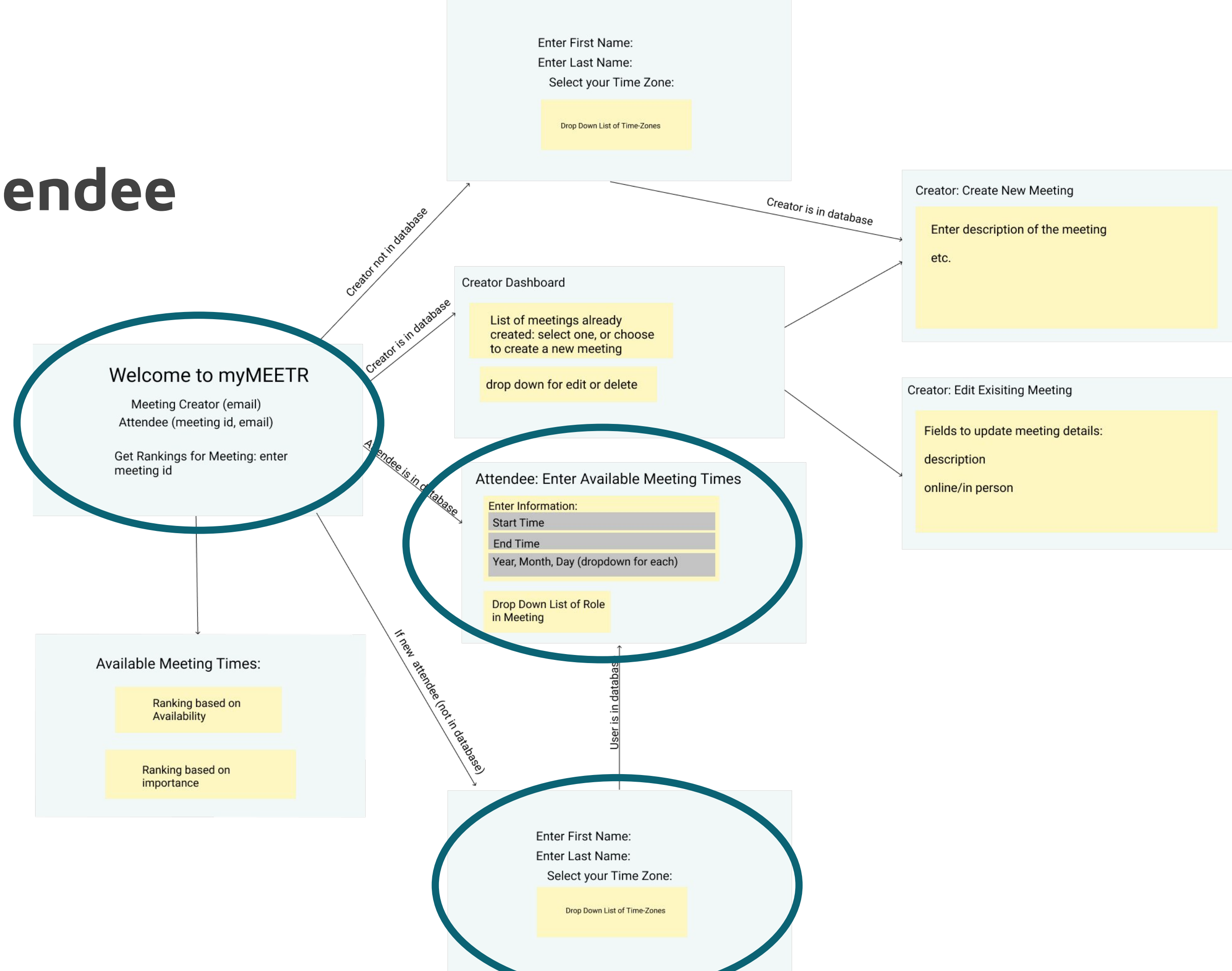


Enter First Name:
Enter Last Name:
Select your Time Zone:
Drop Down List of Time-Zones
Creator not in database
Creator is in database
Creator: Create New Meeting
Enter description of the meeting
etc.
Creator Dashboard
List of meetings already created: select one, or choose to create a new meeting
drop down for edit or delete
Creator is in database
Welcome to myMEETR
Meeting Creator (email)
Attendee (meeting id, email)
Get Rankings for Meeting: enter meeting id
Creator: Edit Exisiting Meeting
Fields to update meeting details:
description
online/in person
Attendee is in database
Attendee: Enter Available Meeting Times
Enter Information:
Start Time
End Time
Year, Month, Day (dropdown for each)
Drop Down List of Role in Meeting
Available Meeting Times:
Ranking based on Availability
Ranking based on importance
If new attendee (not in database)
User is in database
Enter First Name:
Enter Last Name:
Select your Time Zone:
Drop Down List of Time-Zones

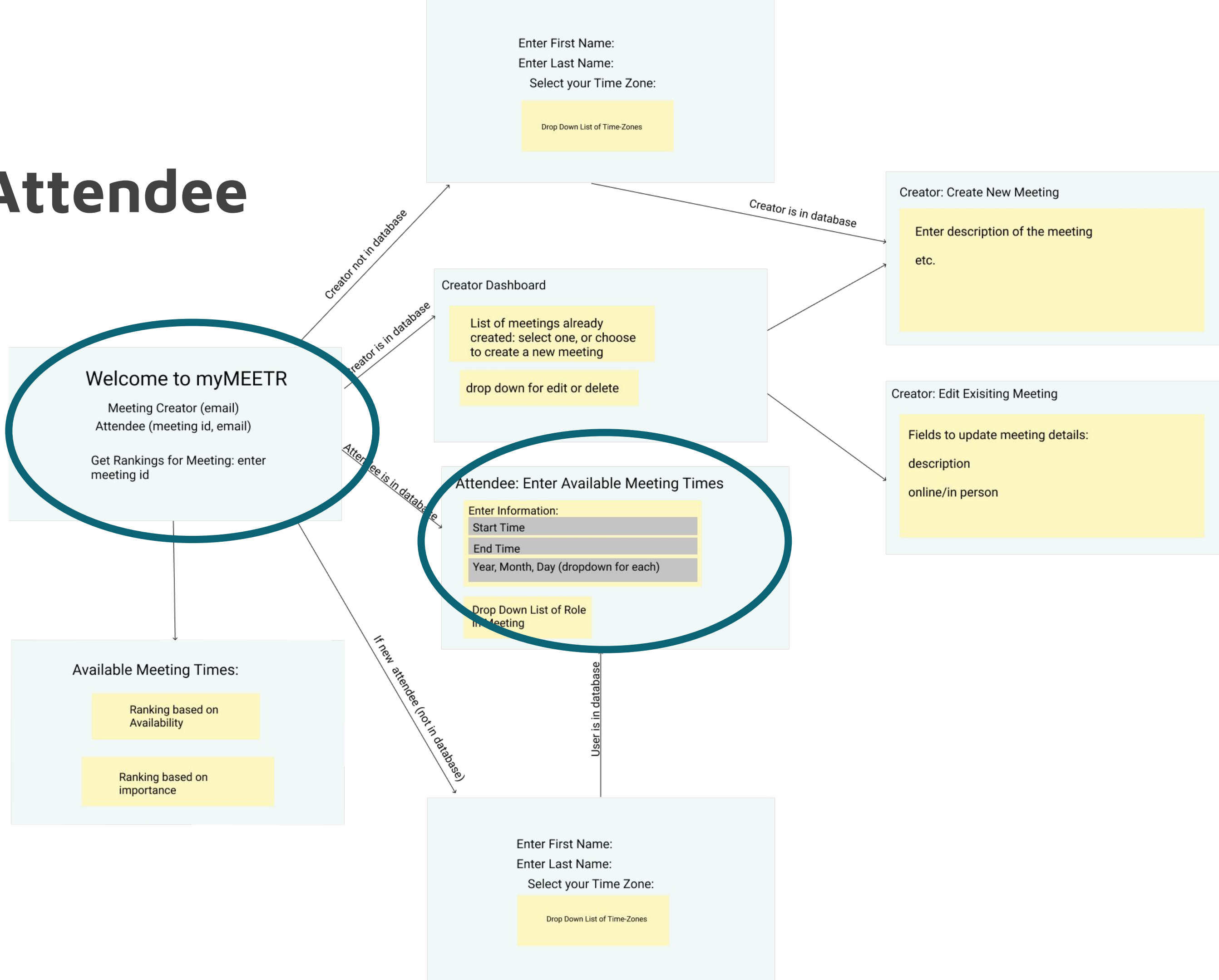
Current Creator
Delete Meeting


Enter First Name:
Enter Last Name:
Select your Time Zone:
Drop Down List of Time-Zones
Creator not in database
Creator is in database
Creator is in database
Creator: Create New Meeting
Enter description of the meeting
etc.
Creator Dashboard
List of meetings already created: select one, or choose to create a new meeting
drop down for edit or delete
Creator: Edit Exisiting Meeting
Fields to update meeting details:
description
online/in person
Welcome to myMEETR
Meeting Creator (email)
Attendee (meeting id, email)
Get Rankings for Meeting: enter meeting id
Attendee is in database
Attendee: Enter Available Meeting Times
Enter Information:
Start Time
End Time
Year, Month, Day (dropdown for each)
Drop Down List of Role in Meeting
Available Meeting Times:
Ranking based on Availability
Ranking based on importance
If new attendee (not in database)
User is in database
Enter First Name:
Enter Last Name:
Select your Time Zone:
Drop Down List of Time-Zones

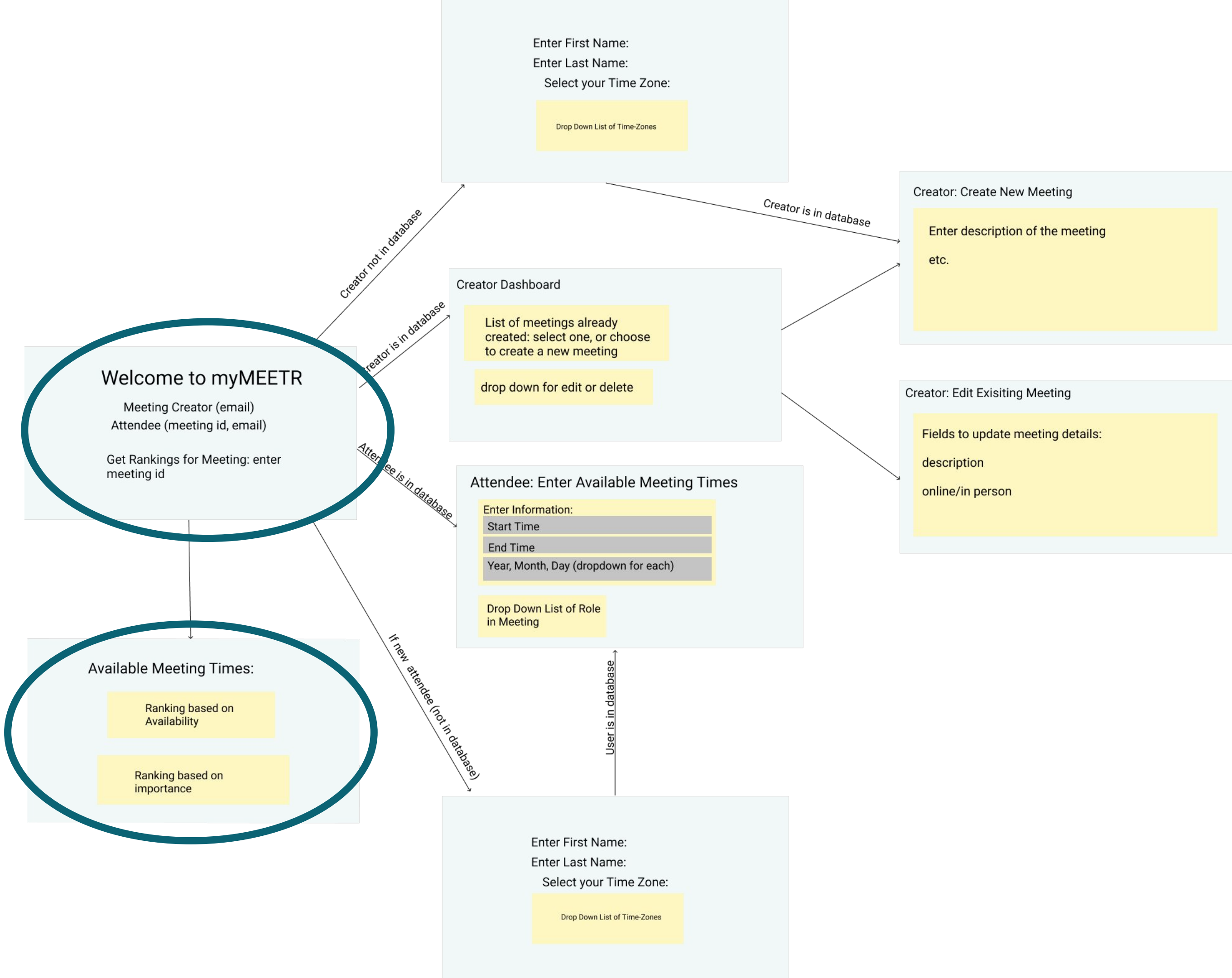
New Attendee


Enter First Name:
Enter Last Name:
Select your Time Zone:
Drop Down List of Time-Zones
Creator not in database
Creator is in database
Creator: Create New Meeting
Enter description of the meeting
etc.
Creator is in database
Creator Dashboard
List of meetings already created: select one, or choose to create a new meeting
drop down for edit or delete
Creator: Edit Exisiting Meeting
Fields to update meeting details:
description
online/in person
Welcome to myMEETR
Meeting Creator (email)
Attendee (meeting id, email)
Get Rankings for Meeting: enter meeting id
Attendee is in database
Attendee: Enter Available Meeting Times
Enter Information:
Start Time
End Time
Year, Month, Day (dropdown for each)
Drop Down List of Role in Meeting
Available Meeting Times:
Ranking based on Availability
Ranking based on importance
If new attendee (not in database)
User is in databas
Enter First Name:
Enter Last Name:
Select your Time Zone:
Drop Down List of Time-Zones

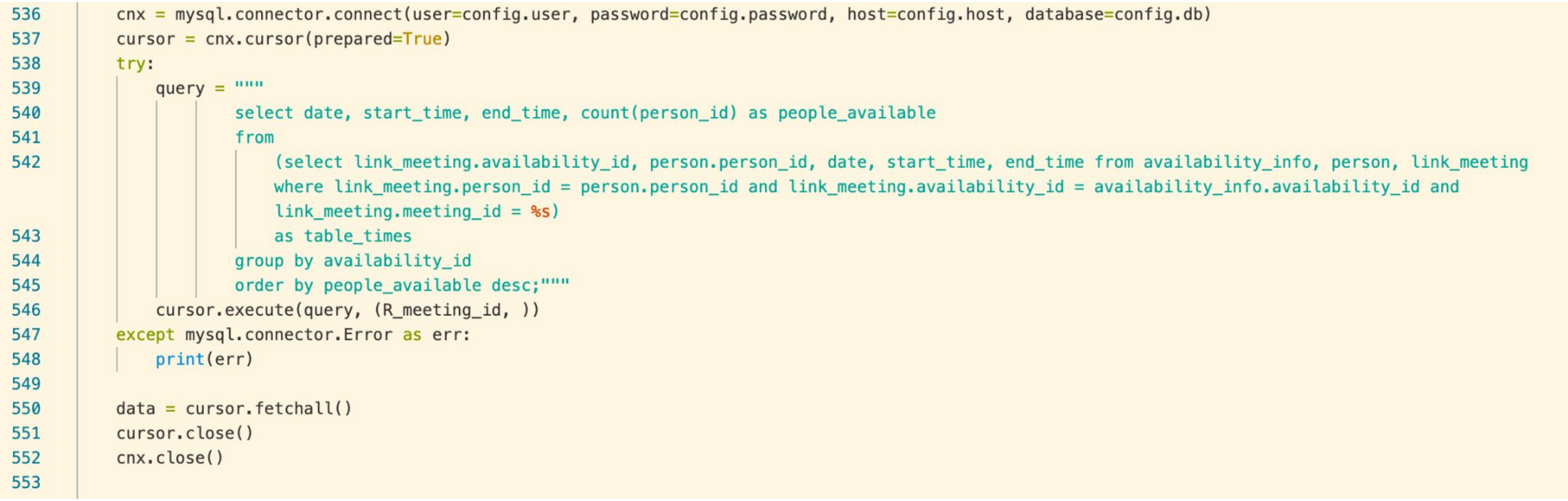
Current Attendee


Enter First Name:
Enter Last Name:
Select your Time Zone:
Drop Down List of Time-Zones
Creator not in database
Creator is in database
Creator: Create New Meeting
Enter description of the meeting
etc.
Creator is in database
Creator Dashboard
List of meetings already created: select one, or choose to create a new meeting
drop down for edit or delete
Creator: Edit Exisiting Meeting
Fields to update meeting details:
description
online/in person
Welcome to myMEETR
Meeting Creator (email)
Attendee (meeting id, email)
Get Rankings for Meeting: enter meeting id
Attendee is in database
Attendee: Enter Available Meeting Times
Enter Information:
Start Time
End Time
Year, Month, Day (dropdown for each)
Drop Down List of Role in Meeting
Available Meeting Times:
Ranking based on Availability
Ranking based on importance
If new attendee (not in database)
User is in database
Enter First Name:
Enter Last Name:
Select your Time Zone:
Drop Down List of Time-Zones

Ranking

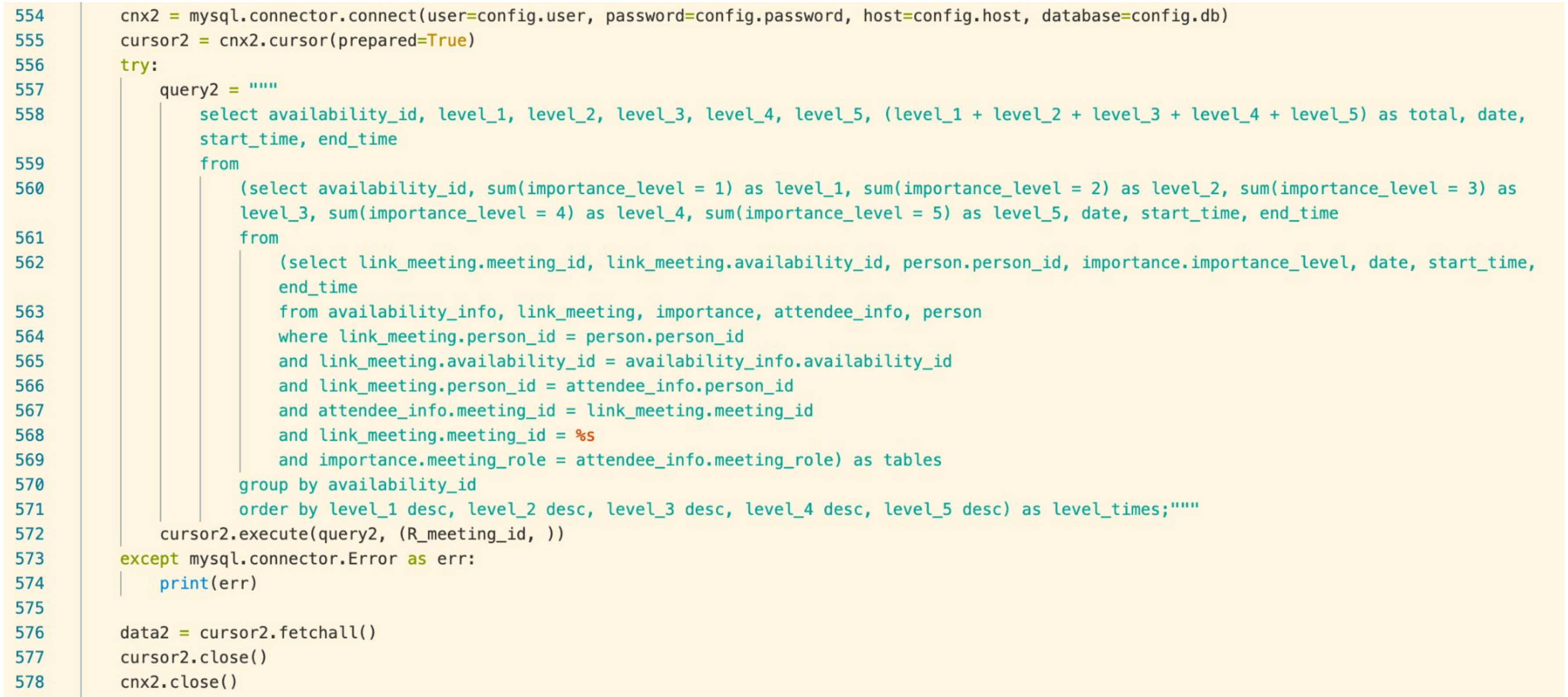

Enter First Name:
Enter Last Name:
Select your Time Zone:
Drop Down List of Time-Zones
Creator not in database
Creator is in database
Creator: Create New Meeting
Enter description of the meeting
etc.
Creator Dashboard
List of meetings already created: select one, or choose to create a new meeting
drop down for edit or delete
reator is in database
Creator: Edit Exisiting Meeting
Fields to update meeting details:
description
online/in person
Welcome to myMEETR
Meeting Creator (email)
Attendee (meeting id, email)
Get Rankings for Meeting: enter meeting id
Attendee is in database
Attendee: Enter Available Meeting Times
Enter Information:
Start Time
End Time
Year, Month, Day (dropdown for each)
Drop Down List of Role in Meeting
Available Meeting Times:
Ranking based on Availability
Ranking based on importance
If new attendee (not in database)
User is in database
Enter First Name:
Enter Last Name:
Select your Time Zone:
Drop Down List of Time-Zones

Ranking - Complex Queries (1) + Prepared Statements



```
    cnx = mysql.connector.connect(user=config.user, password=config.password, host=config.host, database=config.db)
    cursor = cnx.cursor(prepared=True)
    try:
        query = """
                select date, start_time, end_time, count(person_id) as people_available
                from
                    (select link_meeting.availability_id, person.person_id, date, start_time, end_time from availability_info, person, link_meeting where link_meeting.person_id = person.person_id and link_meeting.availability_id = availability_info.availability_id and link_meeting.meeting_id = %s)
                    as table_times
                group by availability_id
                order by people_available desc;"""
        cursor.execute(query, (R_meeting_id, ))
    except mysql.connector.Error as err:
        print(err)

    data = cursor.fetchall()
    cursor.close()
    cnx.close()

```

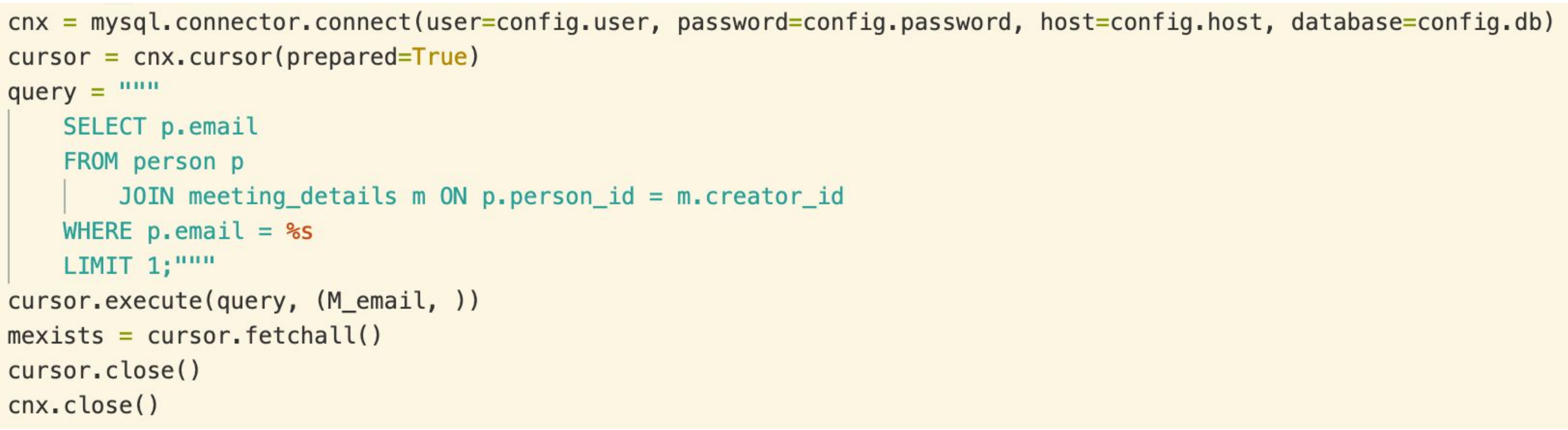
Ranking - Complex Queries (2) + Prepared Statements



```
    cnx2 = mysql.connector.connect(user=config.user, password=config.password, host=config.host, database=config.db)
    cursor2 = cnx2.cursor(prepared=True)
    try:
        query2 = """
            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, start_time, end_time
            from
                (select availability_id, sum(importance_level = 1) as level_1, sum(importance_level = 2) as level_2, sum(importance_level = 3) as level_3, sum(importance_level = 4) as level_4, sum(importance_level = 5) as level_5, date, start_time, end_time
                from
                    (select link_meeting.meeting_id, link_meeting.availability_id, person.person_id, importance.importance_level, date, start_time, end_time
                    from availability_info, link_meeting, importance, attendee_info, person
                    where link_meeting.person_id = person.person_id
                    and link_meeting.availability_id = availability_info.availability_id
                    and link_meeting.person_id = attendee_info.person_id
                    and attendee_info.meeting_id = link_meeting.meeting_id
                    and link_meeting.meeting_id = %s
                    and importance.meeting_role = attendee_info.meeting_role) as tables
                group by availability_id
                order by level_1 desc, level_2 desc, level_3 desc, level_4 desc, level_5 desc) as level_times;"""
        cursor2.execute(query2, (R_meeting_id, ))
    except mysql.connector.Error as err:
        print(err)

    data2 = cursor2.fetchall()
    cursor2.close()
    cnx2.close()
```

Welcome Page - Prepared Statements

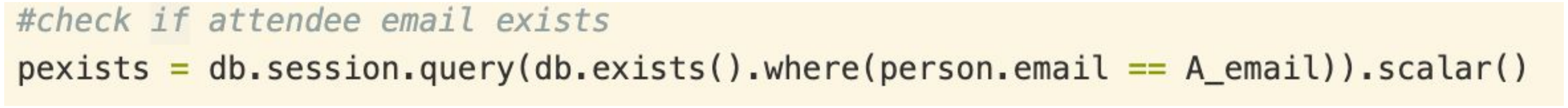
Prepared Statements: (153)



```
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

Prepared Statements: (132, 135, 166, 172, 178, 192)



```
#check if attendee email exists
pexists = db.session.query(db.exists().where(person.email == A_email)).scalar()
```

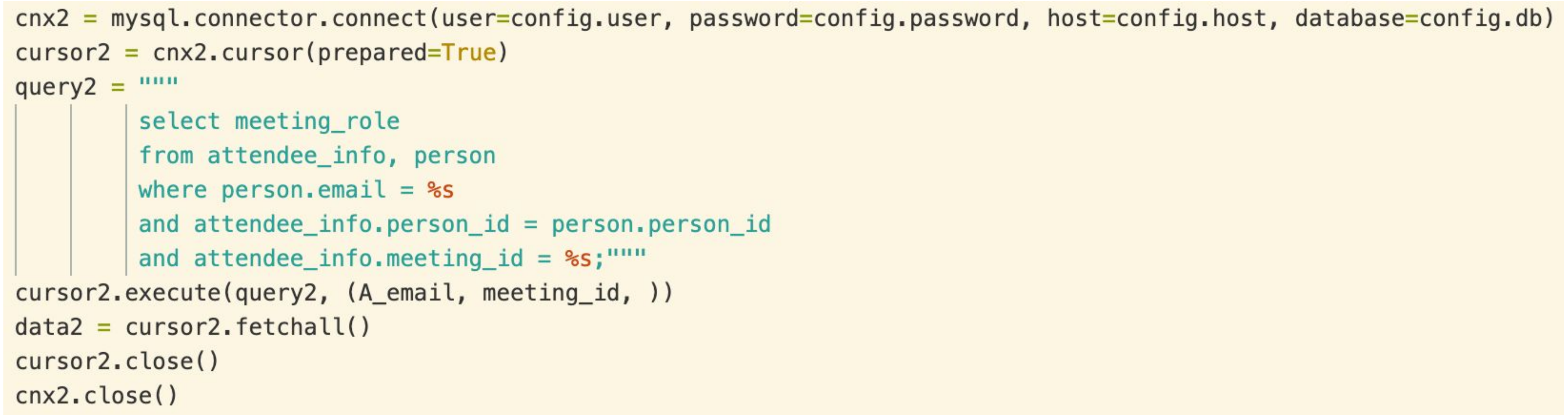
New Attendee Page - ORM

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

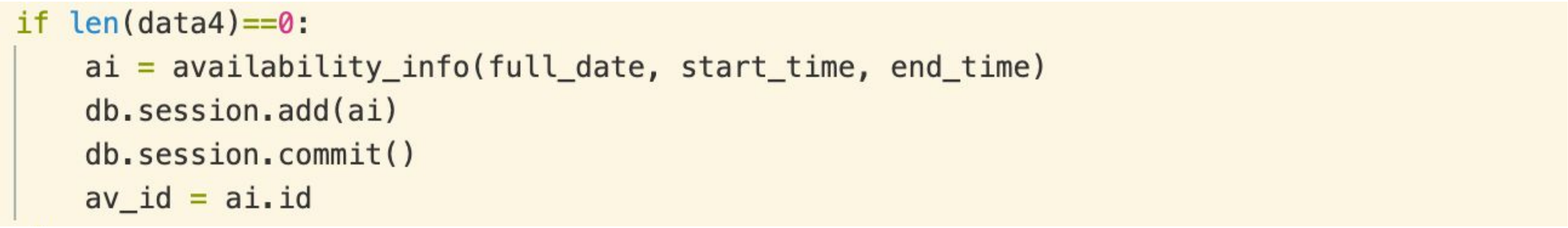
Prepared Statements: (257, 277, 339, 399, 462, 485, 517)



```
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
```

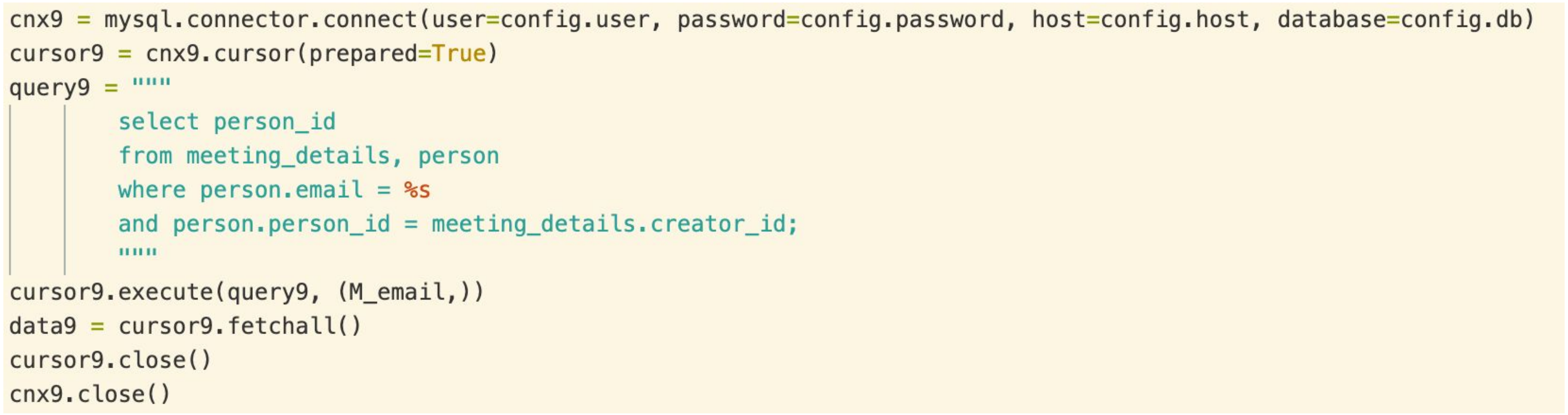
New Creator Page - ORM

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

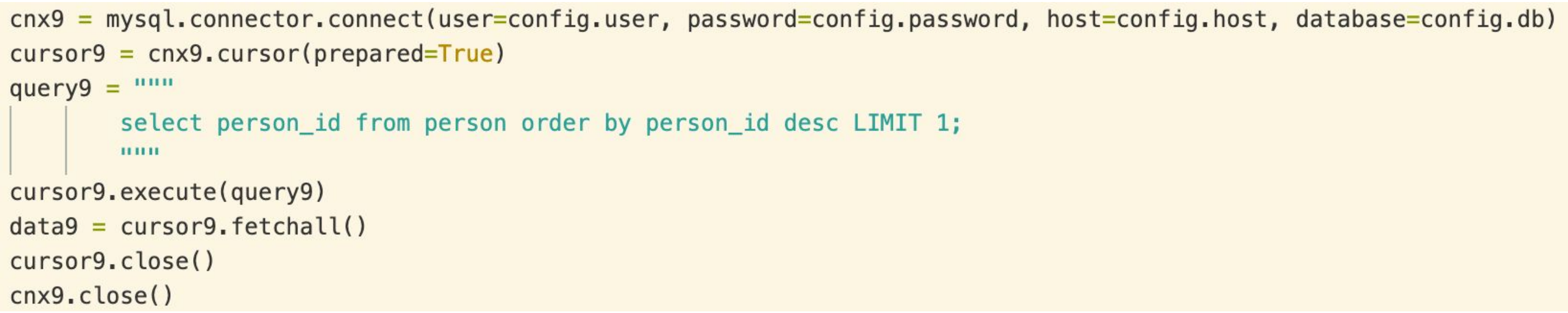
Prepared Statements: (642, 657, 670, 693)



```
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()
```

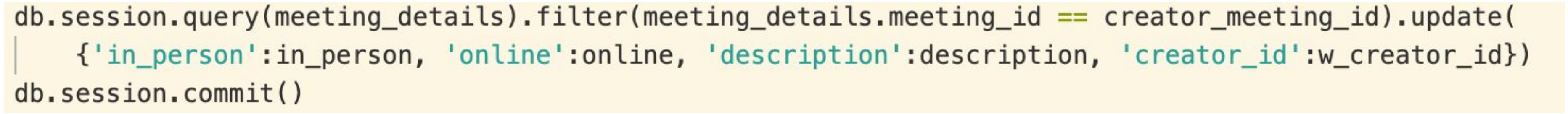
Create Meeting Page - ORM

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()
```

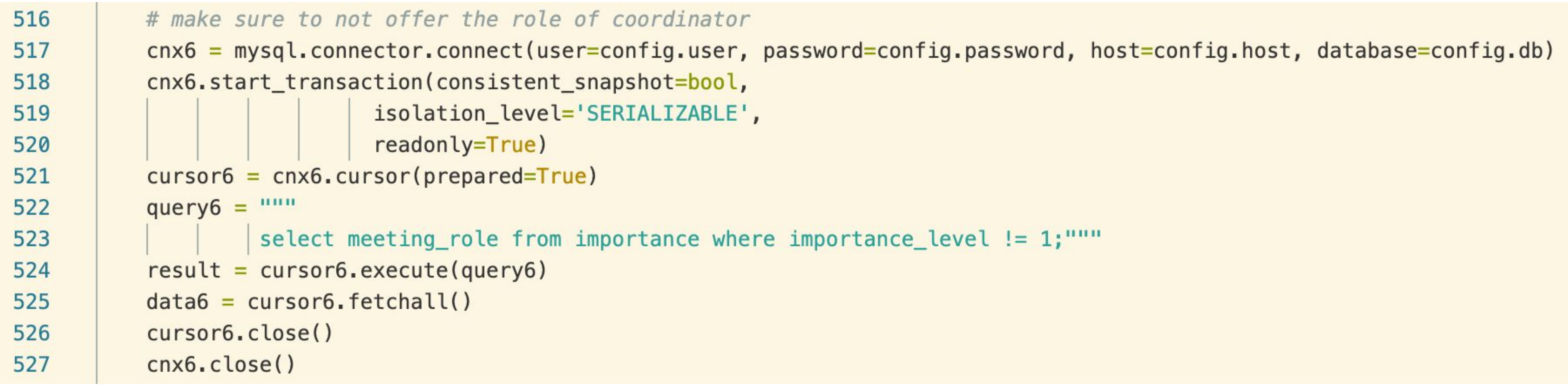
Edit Existing Meeting Page - ORM

Prepared Statements: (584, 597, 613, 624)



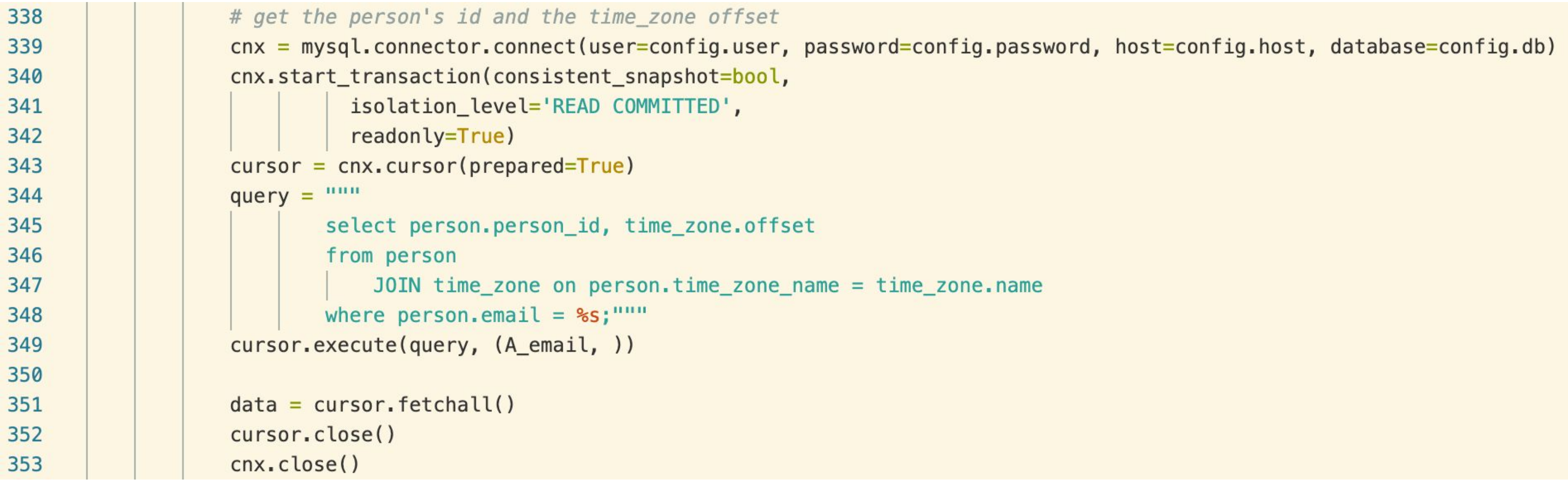
```
db.session.query(meeting_details).filter(meeting_details.meeting_id == creator_meeting_id).update(
    {'in_person':in_person, 'online':online, 'description':description, 'creator_id':w_creator_id})
db.session.commit()
```

Transactions (1)



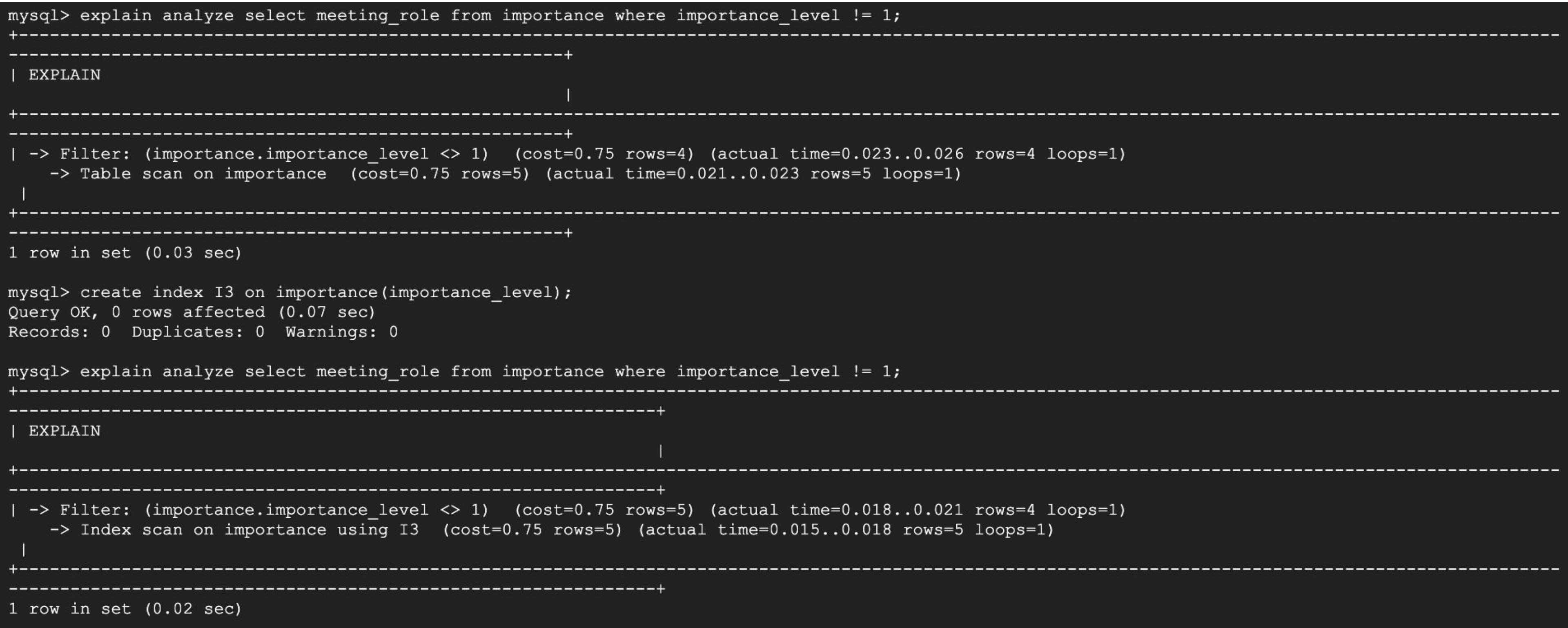
```python
    # make sure to not offer the role of coordinator
    cnx6 = mysql.connector.connect(user=config.user, password=config.password, host=config.host, database=config.db)
    cnx6.start_transaction(consistent_snapshot=bool,
                      isolation_level='SERIALIZABLE',
                      readonly=True)
    cursor6 = cnx6.cursor(prepared=True)
    query6 = """
             select meeting_role from importance where importance_level != 1;"""
    result = cursor6.execute(query6)
    data6 = cursor6.fetchall()
    cursor6.close()
    cnx6.close()
```

Transactions (2)



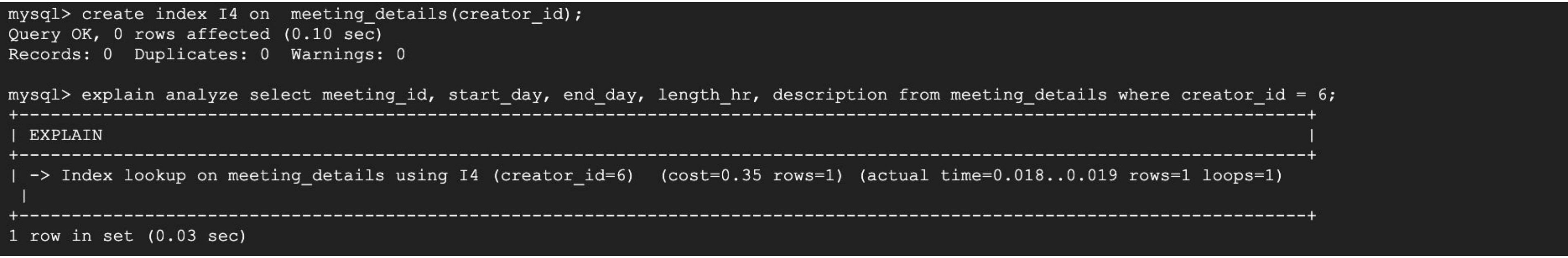
```python
# get the person's id and the time_zone offset
cnx = mysql.connector.connect(user=config.user, password=config.password, host=config.host, database=config.db)
cnx.start_transaction(consistent_snapshot=bool,
          isolation_level='READ COMMITTED',
          readonly=True)
cursor = cnx.cursor(prepared=True)
query = """
        select person.person_id, time_zone.offset
        from person
            JOIN time_zone on person.time_zone_name = time_zone.name
        where person.email = %s;"""
cursor.execute(query, (A_email, ))

data = cursor.fetchall()
cursor.close()
cnx.close()
```

Indices (1)

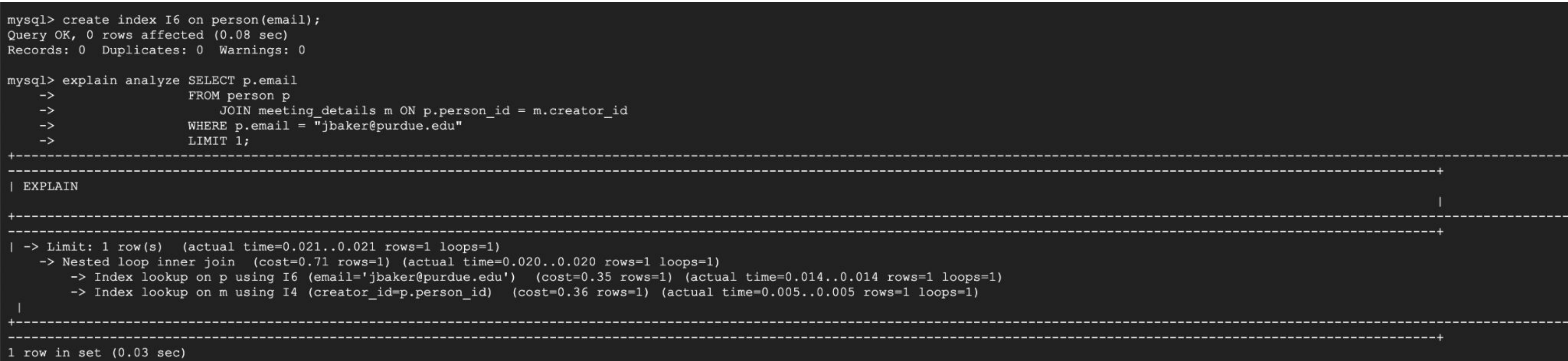


```
mysql> explain analyze select meeting_role from importance where importance_level != 1;
+-----------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------+
|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