Yes ✅, your UserAccess model is correctly set up to **give view access to specific users**, and you **can absolutely test this in the Django shell**. Below, I’ll walk you through:

**✅ 1. Confirm your UserAccess model works**

You have:

class UserAccess(models.Model):

user = models.OneToOneField(User, on\_delete=models.CASCADE)

can\_view\_requirements = models.BooleanField(default=False)

def \_\_str\_\_(self):

return f"{self.user.username} access: {self.can\_view\_requirements}"

This setup:

* Gives each user a single UserAccess object.
* Allows toggling access for specific users.

✅ Perfect for "selected users" and scalable.

**✅ 2. Make Migrations and Migrate**

Run in terminal:

python manage.py makemigrations

python manage.py migrate

**✅ 3. Use Django Shell to Demo**

Launch the shell:

python manage.py shell

Then run the following step-by-step:

**✅ Create users (if needed)**

from django.contrib.auth.models import User

from chat.models import UserAccess

# Create a user (if needed)

user1 = User.objects.create\_user(username="normaluser1", password="testpass")

user2 = User.objects.create\_user(username="normaluser2", password="testpass")

**✅ Give view access to selected users**

# Give access to user1 only

access1 = UserAccess.objects.create(user=user1, can\_view\_requirements=True)

# user2 should NOT have access

access2 = UserAccess.objects.create(user=user2, can\_view\_requirements=False)

**✅ Check access for any user**

# Check if user1 can view requirements

UserAccess.objects.get(user=user1).can\_view\_requirements # True

# Check for user2

UserAccess.objects.get(user=user2).can\_view\_requirements # False

**✅ Check if a user has access in Python logic**

def has\_access(user):

try:

return UserAccess.objects.get(user=user).can\_view\_requirements

except UserAccess.DoesNotExist:

return False

Then:

has\_access(user1) # True

has\_access(user2) # False

**✅ Can You Give Access to "All Users"?**

Yes – simply loop through all users:

from django.contrib.auth.models import User

from chat.models import UserAccess

for user in User.objects.all():

access, created = UserAccess.objects.get\_or\_create(user=user)

access.can\_view\_requirements = True

access.save()

This gives **all users** view access. You can set it to False too to revoke.

**✅ Summary**

| **Action** | **How** |
| --- | --- |
| Give access to selected user | UserAccess.objects.create(user=..., can\_view\_requirements=True) |
| Give access to all | Loop over all User and set field |
| Check in consumer/view | Use UserAccess.objects.get(user=user).can\_view\_requirements |
| Test in shell | ✅ Fully supported |