



THE ESCAPE ROOM

TØNSBERG

Wenxiao Zhang

Zhuoni Yang

Design Rationale

We are proposed to design an online escape room game which allows multi-players to join. In each level, players may search every possible corner of the room to get hints of the way to escape. In certain occasions, player may also need collaborate and exchange information they gained to get more hints.



Outlines

- Util now
- Static web pages (login/out, index page)
- Level one front-end demo
- `.model.py`
- Live demo
- `.Next steps`

Until now

Front: Zhuoni Yang

The basic user interface of the website

Room model for the game

Player model

Level 1 - Player's interaction with the room

Until now

Backend: Wenxiao Zhang

Register

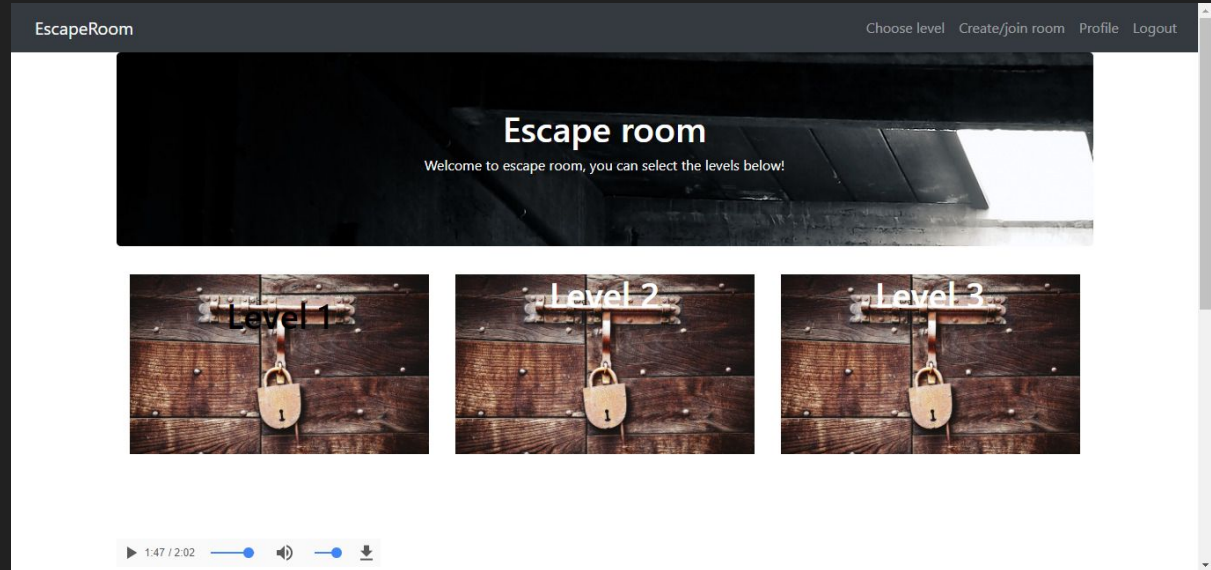
Login

Model.py

Log in/ Join games

First, players log in and get into the main page, which shows the current ongoing games and all the levels.

Players can also see others' profiles and send messages to them.

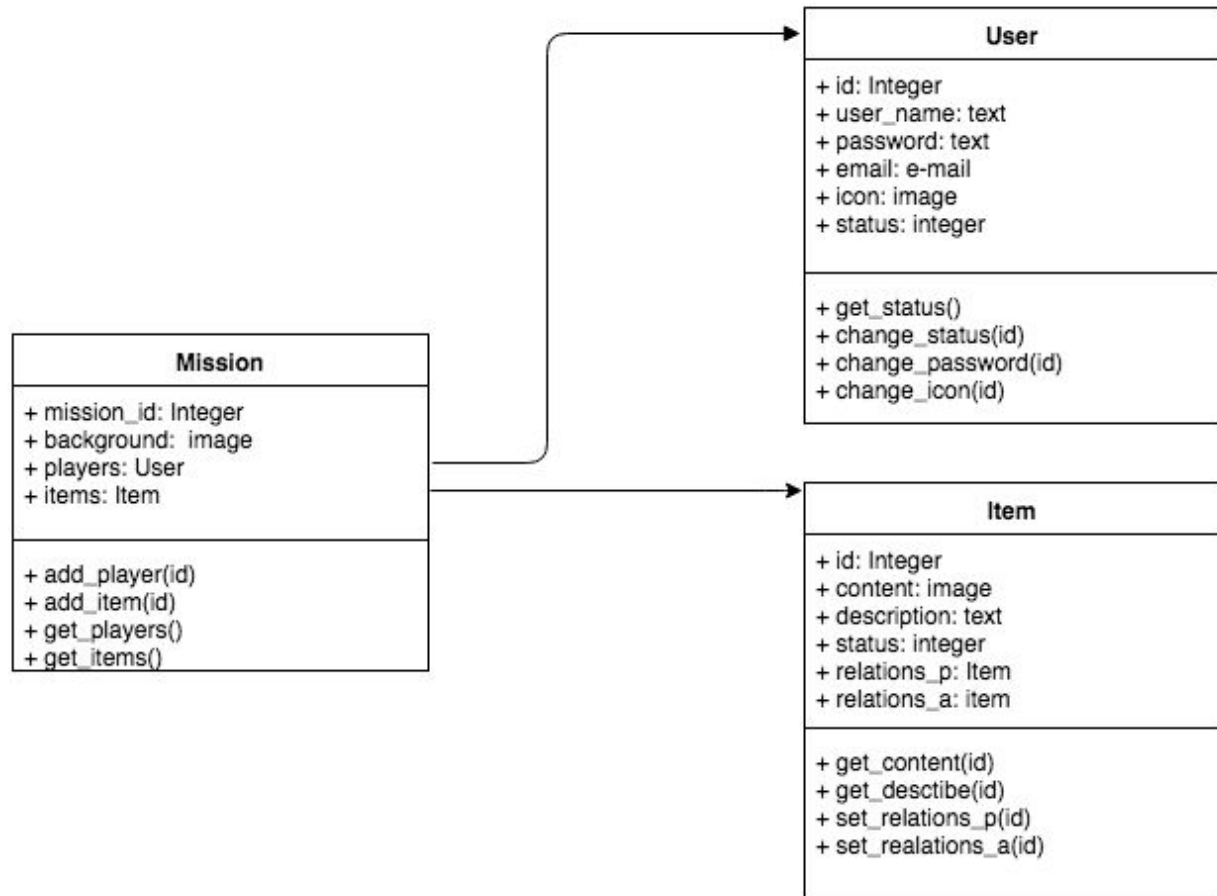


Front-End Demo

EscapeRoom Level 1
Use arrows or "W""S""A""D" to control characters, mouse for camera



model.py



model.py

```
class UserData(models.Model):
    user = models.ForeignKey(User, default=None)
    image = models.ImageField(upload_to="image", default="image/default.jpeg")
    creation_time = models.DateTimeField(default=datetime.now, blank=True)
    bio = models.CharField(max_length=430, blank=True)
    status = models.IntegerField(default=0, blank=True)
    # this to record which level the players is in
    levels = models.IntegerField(default=0, blank=True)
    positionx = models.FloatField(default=0, blank=True)
    positionz = models.FloatField(default=0, blank=True)
```

model.py

```
class Item(models.Model):
    item_id = models.IntegerField()
    content = models.ImageField()
    status = models.IntegerField(default=0, blank=True)
    description = models.TextField(max_length=100, blank=True)
    relations_p = models.ForeignKey('self', blank=True, null=True, related_name="relations_prev")
    relations_a = models.ForeignKey('self', blank=True, null=True, related_name="relations_after")

class Mission(models.Model):
    level_id = models.IntegerField()
    background = models.ImageField()
    items = models.ManyToManyField(Item, related_name="items")
    players = models.ManyToManyField(UserData, related_name="players")
```

Next steps

1. Design more levels
2. View a item: Interaction when a player find some items
3. Optional mini games: Design puzzle games for key or password
4. Multiplayers:
 - a. Send player's location change and events to backend
 - b. Handle multiple players