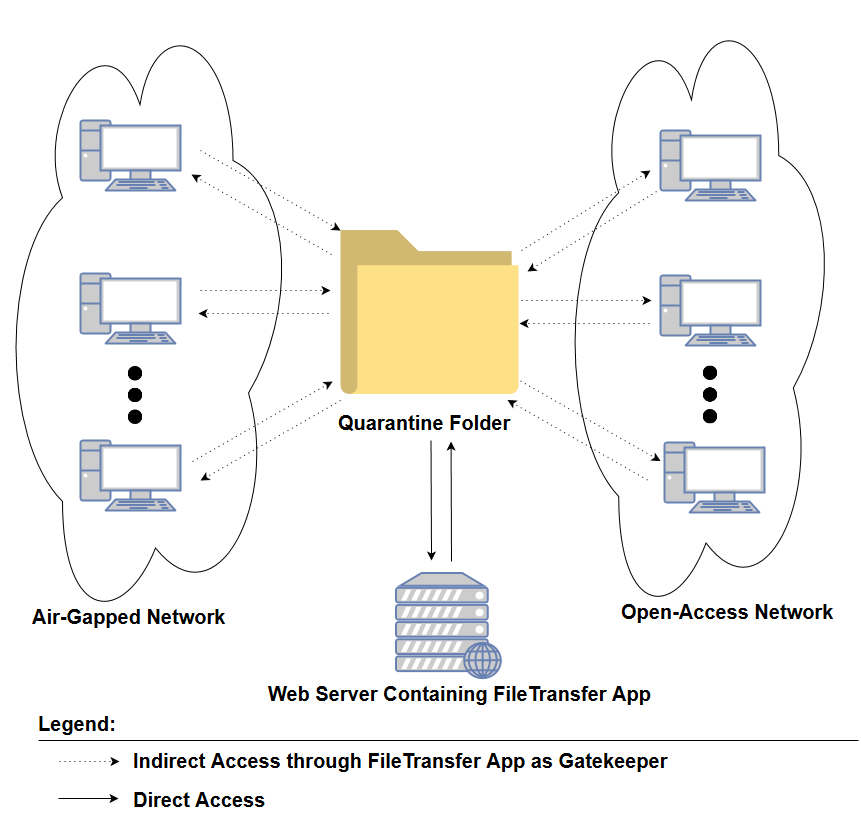
# هدف:

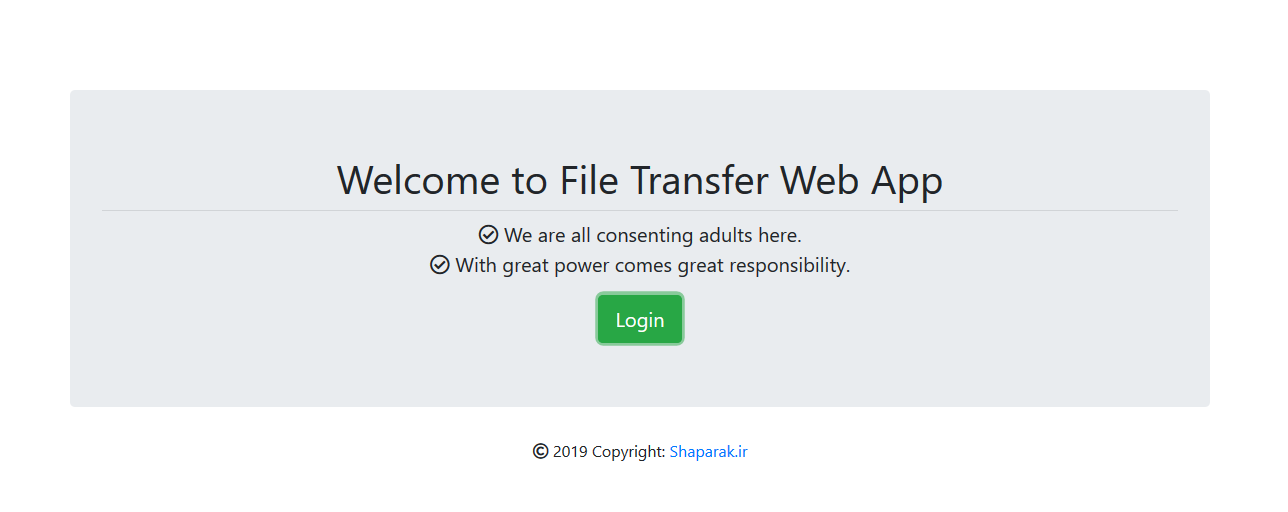
انتقال فایل از شبکه بسته (ایزوله شده)[[1]](#footnote-1) به شبکه با دسترسی باز (فلش دیسک، اینترنت). (شکل *1*)



شکل 1: مدل مفهومی انتقال فایل بین دو شبکه

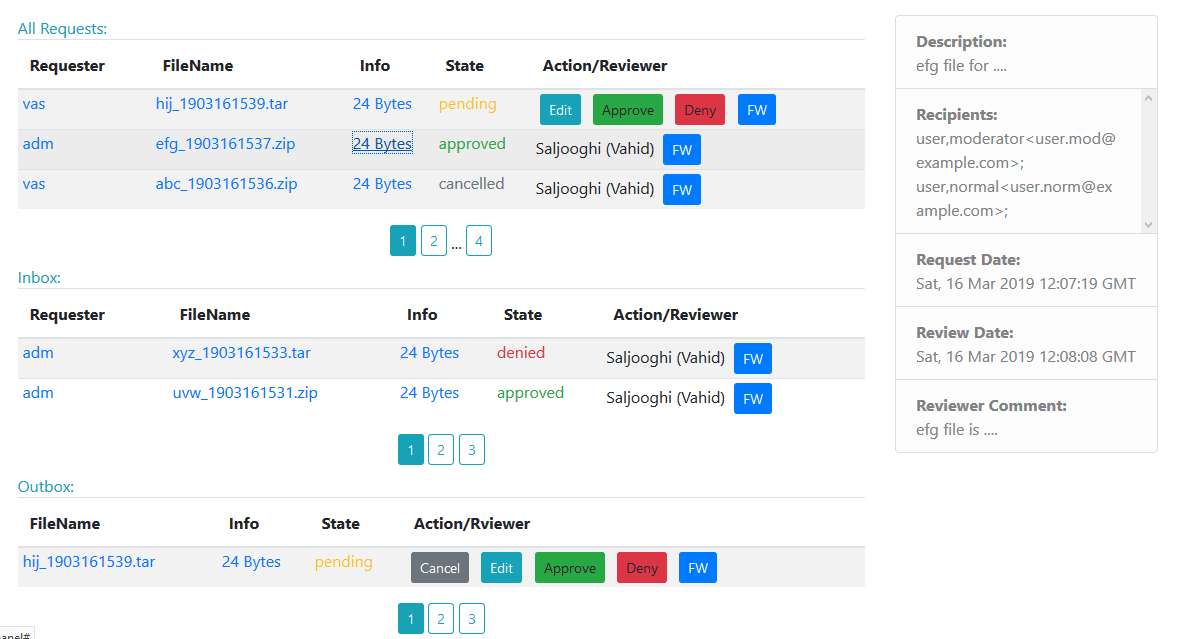
# نحوه عملکرد:

وب اپلیکیشن انتقال فایل[[2]](#footnote-2) (شکل 2) بر روی وب سروری با آدرس سوکت[[3]](#footnote-3) خاص قرار می گیرد. به این ترتیب کاربر می تواند با اتصال به سرور (از شبکه هایی که سرور قابل دسترسی است)، درخواست خود را در پنل مختص خود ثبت کند یا وضعیت درخواست های از قبل ثبت شده خود را مشاهده نماید. ثبت درخواست همراه با بالاگذاری[[4]](#footnote-4) فایل به صورت فشرده شده می باشد.



شکل 2: صفحه خانه جهت خوش آمدگویی

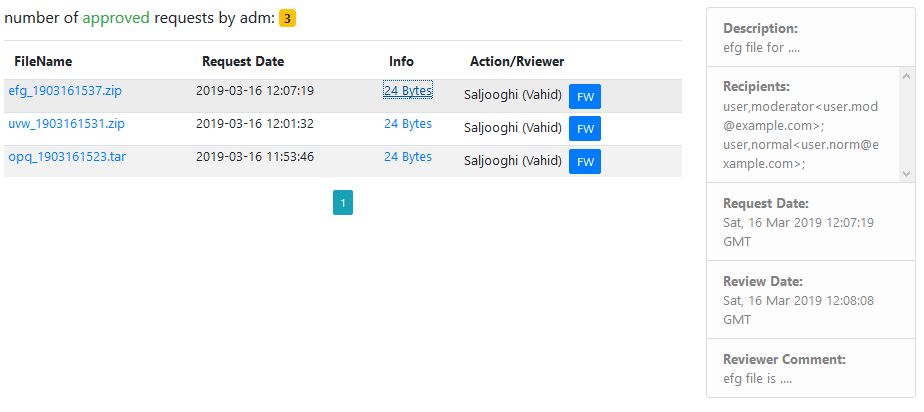
پس از ارزیابی درخواست ثبت شده توسط Moderator و در صورت تایید، پایین گذاری[[5]](#footnote-5) فایل بر روی پنل کاربر درخواست دهنده (در شبکه با دسترسی باز) امکان پذیر می باشد. علاوه بر آن اسامی افرادی که در لیست دریافت کنندگان[[6]](#footnote-6) فایل قراردارند نیز (در صورت تایید) قادر به دانلود فایل در پنل مختص خود به صورت دسته بندی شده (All Requests, Inbox, Outbox) هستند.



شکل 3: پنل درخواست ها

# ویژگی ها:

* وب اپلیکیشن قابلیت مدیریت حساب های کاربری[[7]](#footnote-7) (ایجاد[[8]](#footnote-8)، حذف[[9]](#footnote-9)، ویرایش[[10]](#footnote-10) و مسدود کردن[[11]](#footnote-11)) و همچنین مدیریت درخواست های[[12]](#footnote-12) انتقال فایل را می دهد (لغو[[13]](#footnote-13)، ویرایش، پذیرفتن[[14]](#footnote-14)، رد[[15]](#footnote-15) و ارسال بیشتر[[16]](#footnote-16) کردن درخواست های انتقال فایل). همچنین امکان جداسازی و نمایش درخواست ها براساس وضعیت (لغو شده، منتظر بررسی[[17]](#footnote-17)، رد شده و پذیرفته شده) یا براساس نام کاربری در صفحه ای مجزا می باشد (شکل 4).



شکل 4: لیست تمام درخواست های تایید شده کاربر خاص (adm)

* اطلاعات هر درخواست (شرح درخواست[[18]](#footnote-18)، دریافت کنندگان، تاریخ درخواست[[19]](#footnote-19)، تاریخ بررسی/ ارزیابی مادریتور[[20]](#footnote-20) و اضهارنظر مادریتور[[21]](#footnote-21)) در نوارجانبی[[22]](#footnote-22) پس از کلیک بر روی بخش Info نمایش داده می شود.
* برای وب اپلیکیشن 3 نوع کاربری با مشخصات زیر تعریف شده است:

1. کاربری عادی (Normal):

می تواند درخواست خودش رو لغو یا ویرایش کند.

می تواند درخواست انتقال فایل ثبت کند تا بعدا توسط یک Moderator بررسی شود.

نمی تواند کاربر جدید ایجاد کند یا کاربری را حذف یا ویرایش کند.

نمی تواند درخواست های دیگران را ارزیابی کند.

می تواند درخواست های تایید شده دیگران را که خود به عنوان دریافت کننده درخواست ذکر شده است پایین گذاری کند.

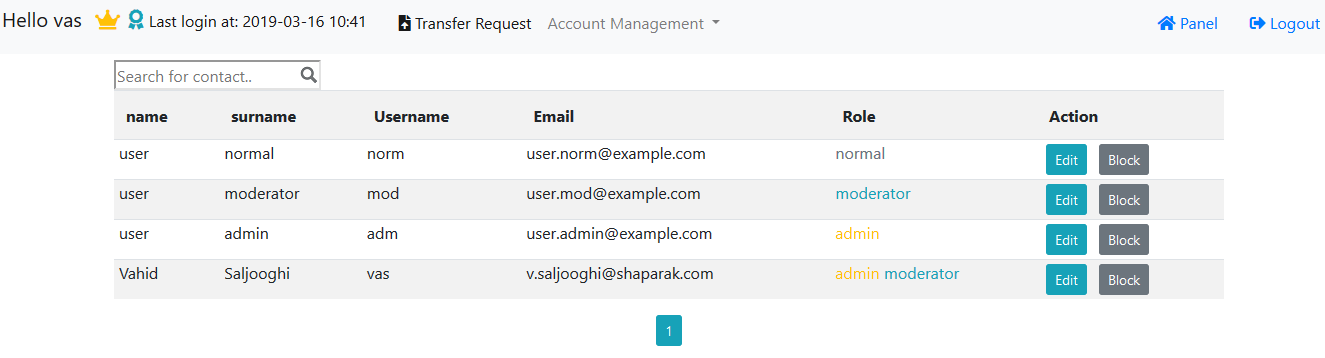
1. کاربری Admin:

می تواند درخواست خودش رو لغو یا ویرایش کند.

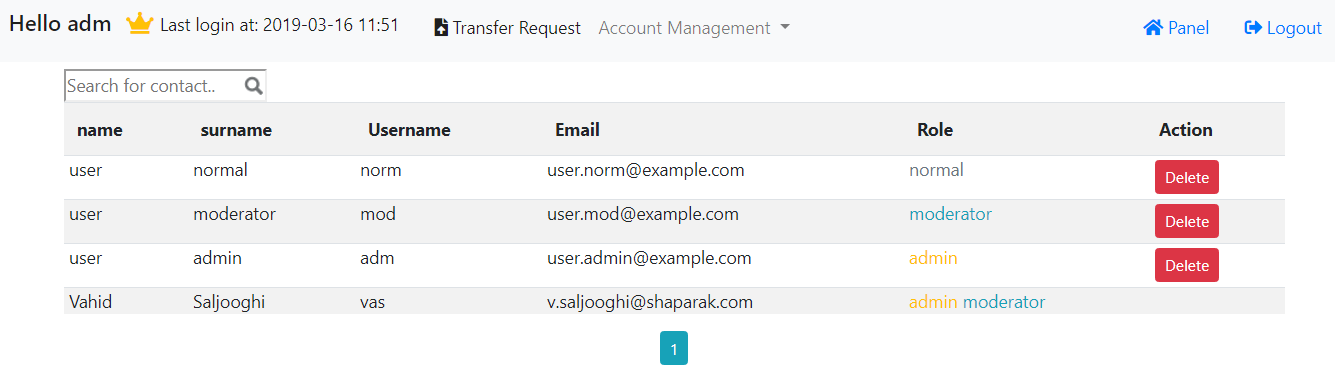
می تواند درخواست انتقال فایل ثبت کند تا بعدا توسط یک Moderator بررسی شود.

می تواند حساب کاربر با ویژگی ادمین یا عادی (و نه Moderator) ایجاد کند. می تواند هر نوع حساب کاربر به غیر از کاربر با ویژگی های ادمین و مادریتور(ادمین+مادریتور بالاترین سطح دسترسی را دارد) را حذف، مسدود یا ویرایش کند. (شکل 5 و شکل 6)

می تواند درخواست های تایید شده دیگران را که خود به عنوان دریافت کننده درخواست ذکر شده است پایین گذاری کند.



شکل 5: کاربر با هر دو دسترسی ادمین و مادریتور امکان مدیریت تمام حساب ها را دارد.



شکل 6: کاربر با دسترسی ادمین (با تاج طلایی مشخص شده است) امکان تغییر برای کاربر ادمین و مادریتور را ندارد.

1. کاربری Moderator:

می تواند درخواست خودش رو لغو، رد، تایید یا ویرایش کند.

می تواند درخواست های دیگران را تایید، رد یا ویرایش کند.

نمی تواند حساب کاربر ایجاد، حذف یا ویرایش کند.

کاربر وب اپلیکیشن می تواند یک یا چند نوع کاربری را همزمان دارا باشد. به این ترتیب کاربر دارای ویژگی های/کاربری های Admin و Moderator دارای بالاترین سطح دسترسی می باشد.

کاربر وب اپلیکیشن می تواند جهت جستجو در بین تماس های[[23]](#footnote-23) نمایش داده شده، با کلیک بر جعبه "جستجو" به نحوه دلخواه لیست تماس ها را فیلتر نماید.

* در پیاده سازی از زبان های برنامه نویسی جاوااسکریپت، پایتون و کتابخانه های JQuery، Bootstrap، Font Awesome و میکروفریمورک فلاسک (به همراه کتابخانه هایش ) استفاده شده است. ساختار درختی فایل ها در زیر آمده است.

FileTransferWebApp

│ run.py

│

└───FileTransferPackage

│ forms.py

│ models.py

│ routes.py

│ \_\_init\_\_.py

│

├───DB

│ FileTransfer.db

│

├───static

│ ├───css

│ │ all.min.css

│ │ bootstrap-grid.min.css

│ │ bootstrap-grid.min.css.map

│ │ bootstrap-reboot.min.css

│ │ bootstrap-reboot.min.css.map

│ │ bootstrap.min.css

│ │ bootstrap.min.css.map

│ │ main.css

│ │

│ ├───img

│ │ FileTransfer.png

│ │ searchicon.png

│ │

│ ├───js

│ │ AJAX2GetReqInfo.js

│ │ bootstrap.bundle.js

│ │ bootstrap.bundle.js.map

│ │ bootstrap.bundle.min.js

│ │ bootstrap.bundle.min.js.map

│ │ bootstrap.js

│ │ bootstrap.js.map

│ │ bootstrap.min.js

│ │ bootstrap.min.js.map

│ │ delUser.js

│ │ ImportContacts.js

│ │ jquery-3.3.1.js

│ │ reqAction.js

│ │ searchContacts.js

│ │ SelectContacts.js

│ │

│ └───webfonts

│ fa-brands-400.eot

│ fa-brands-400.svg

│ fa-brands-400.ttf

│ fa-brands-400.woff

│ fa-brands-400.woff2

│ fa-regular-400.eot

│ fa-regular-400.svg

│ fa-regular-400.ttf

│ fa-regular-400.woff

│ fa-regular-400.woff2

│ fa-solid-900.eot

│ fa-solid-900.svg

│ fa-solid-900.ttf

│ fa-solid-900.woff

│ fa-solid-900.woff2

│

├───templates

│ create\_account.html

│ delete\_accounts.html

│ edit\_account.html

│ edit\_accounts.html

│ edit\_req.html

│ fw\_req.html

│ home.html

│ layout.html

│ login.html

│ panel\_moderator.html

│ panel\_normal.html

│ reset\_request.html

│ reset\_token.html

│ trans\_req.html

│ x\_reqs.html

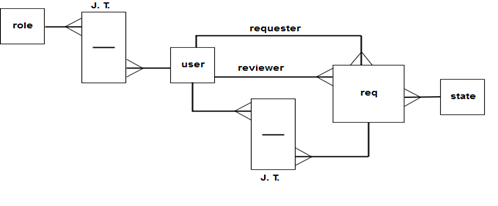
│

└───includes

\_colorize.html

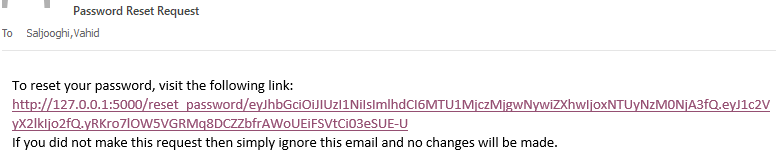
\_formhelpers.html

* از آنجا که از کتابخانه SQLAlchemy به عنوان یک Object-Relational Mapping استفاده شده است، امکان پشتیبانی از پایگاه داده های رایج می باشد. شکل 7 دیاگرام رابطه بین جداول پایگاه داده را نشان می دهد.



شکل 7: Entity Relationship Diagram

* پس از تعریف کاربر توسط ادمین، ایمیلی به صورت خودکار جهت تعیین/ انتخاب رمز ورود برای کاربر ارسال می شود و رمز انتخاب شده توسط کاربر به صورت رمزنگاری[[24]](#footnote-24) شده در پایگاه داده ذخیره می شود.



شکل 8: نمونه ایمیل ارسال شده جهت انتخاب رمز برای اولین بار یا ریست کردن رمز فراموش شده

* از روی محتوا و نام فایل Digest تهیه می شود و در پایگاه داده ذخیره می گردد. به این ترتیب امکان اینکه فایلی با نام و محتوا یکسان چندبار بارگذاری شود وجود ندارد.
* تمام فرم های HTML توسط WTForms قبل از ارسال[[25]](#footnote-25) ارزیابی[[26]](#footnote-26) می شود تا داده های نامعتبر مانند آدرس ایمیل با فرمت اشتباه در پایگاه داده ذخیره نشود و همچنین فرمت ها فایل ها محدود شده است یه چند فرمت فشرده سازی (.zip, .rar, .tar) تا از بارگذاری بدافزار در سمت سرور جلوگیری شود.

# برای ایجاد اولین کاربر در پایگاه داده به صورت دستی:

0- Go to python REPL inside application folder

1- from FileTransferPackage import db,bcrypt

2- from FileTransferPackage.models import \*

3- db.create\_all()

4- pass1=bcrypt.generate\_password\_hash("123456").decode('utf-8')

5- u1=User(name="Vahid", surname="Saljooghi", username="vas", email="vas@example.com", password=pass1, permission=6)

6- db.session.add(u1)

7- db.session.commit()

# دستورات مفید برای فرد نگهدارنده وب اپلیکیشن:

User.\_\_table\_\_.drop(db.engine)

User.\_\_table\_\_.create(db.engine)

UserRequest.drop(db.engine)

UserRequest.create(db.engine)

User.query.all()

User.query.first()

User.query.get(id)

db.drop\_all()

db.session.rollback()

# تغییرات برای بهبود در آینده:

* اضافه کردن پردازش موازی برای ارسال ایمیل تا زمان انتظار کاربر کاهش یابد.
* اضافه کردن Captcha یرای لاگین.

1. Air-gapped [↑](#footnote-ref-1)
2. File Transfer Web App [↑](#footnote-ref-2)
3. Socket Address [↑](#footnote-ref-3)
4. Upload [↑](#footnote-ref-4)
5. Download [↑](#footnote-ref-5)
6. Recipients [↑](#footnote-ref-6)
7. Account Management [↑](#footnote-ref-7)
8. Create [↑](#footnote-ref-8)
9. Delete [↑](#footnote-ref-9)
10. Edit [↑](#footnote-ref-10)
11. Block [↑](#footnote-ref-11)
12. Request Management [↑](#footnote-ref-12)
13. Cancel [↑](#footnote-ref-13)
14. Approve [↑](#footnote-ref-14)
15. Deny [↑](#footnote-ref-15)
16. Forward [↑](#footnote-ref-16)
17. Pending [↑](#footnote-ref-17)
18. Description [↑](#footnote-ref-18)
19. Request Date [↑](#footnote-ref-19)
20. Review Date [↑](#footnote-ref-20)
21. Reviewer Comment [↑](#footnote-ref-21)
22. Sidebar [↑](#footnote-ref-22)
23. Contacts [↑](#footnote-ref-23)
24. Encrypted [↑](#footnote-ref-24)
25. Submit [↑](#footnote-ref-25)
26. Validation [↑](#footnote-ref-26)