



دانشگاه صنعتی قوچان

پروژه کارشناسی پیوسته کامپیوتر گرایش فناوری اطلاعات

طراحی مدیریت کاربران اینترنت روترهای میکروتیک

نگارش: مسلم مجیدی

استاد راهنما:

جناب آقای دکتر نیما صالحی مقدمی

زمستان ۹۸



تقدیر و تشکر

با تشکر فراوان از استاد گرامی جناب آقای دکتر صالحی مقدمی که با آموزش مطالب ارزنده بنده را در گردآوری این مطالب و تنظیم این پروژه یاری نمودند.

چکیده

با توجه به استفاده روزانه و مداوم کاربران از اینترنت، کنترل و نظارت در استفاده از آن توسط مدیران شبکه در مجموعه های مختلف بسیار هائز اهمیت می باشد. با توجه به محدود بودن امکانات داخلی و همچنین سخت و پیچیده بودن نمونه های خارجی مدیریت کاربران اینترنت را بیش از پیش دشوارتر شده است. هدف از انجام این پروژه مدیریت و نظارت بر کاربران در استفاده از اینترنت در مجموعه هایی می باشد که دارای روتر های میکروتیک هستند. فرایند مدیریت و نظارت شامل آماده سازی و تخصیص بسته های اینترنت، مشاهده وضعیت استفاده کاربران از اینترنت بصورت کلی و جزئی و همچنین محدود سازی کاربران در استفاده از اینترنت می باشد .

استفاده از زبان فارسی در طراحی پنل، طراحی ظاهر ساده و بدون گزینه های پیچیده و همچنین استفاده از امکانات خود روتر میکروتیک که در نسخه اصلی وجود ندارد و همچنین طراحی قسمت هایی از قبیل کاربران مشکوک و یا کاربران پر مصرف، این نرم افزار را نسبت به نسخه اصلی و یا نمونه های دیگر متمایز ساخته است. در ساخت این پروژه از زبان برنامه نویسی پایتون، فریم ورک جنگو، API مخصوص میکروتیک به زبان پایتون، زبان جاوا اسکریپت، CSS و Bootstrap استفاده شده است.

فهرست مطالب

عنوان	صفحه
۱. فصل اول	۲
۱-۱. شرح کلی پروژه	۳
۱-۲. اهداف پروژه	۳
۱-۳. بررسی کارهای مشابه	۳
۱-۳-۱. نرم افزار اکانتینگ هات اسپات پلاس	۳
۱-۳-۲. نرم افزار اکانتینگ سیگما	۴
۱-۳-۳. نرم افزار اکانتینگ NTM	۵
۱-۴. سطوح دسترسی برنامه	۷
۱-۵. امکانات نهایی پروژه	۷
۱-۶. ابزار مورد استفاده	۷
۲. فصل دوم	۹
۲-۱. مقدمه ای بر مهندسی نرم افزار	۱۰
۲-۲. تعریف UML	۱۰
۲-۲-۱. نمودار کلاس (Class Diagram)	۱۰
۲-۲-۲. نمودار مورد کاربرد (Use Case Diagram)	۱۱
۲-۲-۳. نمودار توالی (Sequence Diagram)	۱۱
۲-۳. نمودار های پروژه	۱۱
۲-۳-۱. نمودار Use Case کاربر پذیرش	۱۱
۲-۳-۲. نمودار Use Case کاربر مدیر	۱۲
۲-۳-۳. نمودار sequence ورود به سایت	۱۳
۲-۳-۴. نمودار sequence ایجاد اکانت اینترنت	۱۳
۲-۳-۵. نمودار sequence ایجاد پروفایل اینترنت	۱۴
۲-۳-۶. نمودار sequence ایجاد محدودیت برای بسته های اینترنت	۱۴
۲-۳-۷. نمودار sequence اضافه کردن روتر	۱۵
۲-۳-۸. نمودار sequence حذف اطلاعات	۱۵
۲-۳-۸. نمودار Class سایت	۱۶
۳. فصل سوم	۱۷
۳-۱. پایگاه داده	۱۸
۳-۲. سیستم مدیریت پایگاه داده	۱۸

۱۸.....	۳-۲-۱. انواع سیستم های مدیریت پایگاه داده
۱۸.....	۳-۳. SQLite چیست؟.....
۱۹.....	۳-۴. جداول پایگاه داده پروژه
۱۹.....	۳-۴-۱. Username جدول
۱۹.....	۳-۴-۲. Active User جدول
۲۰.....	۳-۴-۳. Customer جدول
۲۰.....	۳-۴-۴. IP Binding جدول
۲۰.....	۳-۴-۵. Limitation جدول
۲۱.....	۳-۴-۶. Profile Limitation جدول
۲۱.....	۳-۴-۷. Profile جدول
۲۱.....	۳-۴-۸. Router جدول
۲۲.....	۳-۴-۹. Session جدول
۲۲.....	۳-۴-۱۰. Mac Address جدول
۲۳.....	۴. فصل چهارم
۲۴.....	۴-۱. صفحه ورود به سایت
۲۴.....	۴-۲. صفحه اصلی
۲۵.....	۴-۳. صفحه فهرست کاربران
۲۷.....	۴-۴. صفحه کاربران فعال
۲۷.....	۴-۵. صفحه کاربران پرمصرف
۲۸.....	۴-۶. صفحه کاربران مشکوک
۲۹.....	۴-۷. صفحه کاربران مسدود شده
۲۹.....	۴-۸. صفحه پروفایل
۳۱.....	۴-۹. صفحه لیست محدودیت ها
۳۲.....	۴-۱۰. صفحه جست و جو
۳۲.....	۴-۱۰-۱. جست و جو بر اساس نشست ها
۳۲.....	۴-۱۰-۲. جست و جو بر اساس کاربر
۳۳.....	۴-۱۱. صفحه روتر ها
۳۴.....	۴-۱۲. صفحه پنل پذیرش
۳۵.....	۴-۱۳. صفحه مدیریت پایگاه داده و عملیات کاربر
۳۵.....	۴-۱۳-۱. صفحه ورود
۳۶.....	۴-۱۳-۲. صفحه ادمین مدیریت پایگاه داده
۳۶.....	۴-۱۳-۳. صفحه کاربران

۳۶ صفحه ساخت کاربر
۳۸ فصل پنجم
۳۹ ۱-۵. خلاصه
۳۹ ۲-۵. جمع بندی
۴۰ ۳-۵. کار های آینده
۴۲ کد های صفحه ورود و قسمت خروج
۴۲ کد های صفحه اصلی پنل با دسترسی ادمین
۴۳ کد های صفحه فهرست کاربران
۴۶ کد های صفحه کاربران فعال
۴۷ کد های صفحه پروفایل
۴۸ کد های صفحه لیست محدودیت ها
۴۹ کد های صفحه جست و جو
۵۰ کد های صفحه روتر
۵۱ کد های صفحه کاربران پر مصرف
۵۳ کد های صفحه کاربران مشکوک
۵۴ کد صفحه لیست سیاه
۵۵ کد های صفحه نشست ها
۵۵ کد های صفحه کاربر پذیرش
۵۶ فهرست منابع

فهرست اشکال

شکل	صفحه
شکل ۱-۱: نرم افزار اکانتینگ هات اسپات پلاس	۴
شکل ۱-۲: نرم افزار اکانتینگ سیگما	۵
شکل ۱-۳: نرم افزار اکانتینگ NTM	۶
شکل ۲-۱: نمودار Use Case کاربر پذیرش	۱۲
شکل ۲-۲: نمودار Use Case کاربر مدیر	۱۲
شکل ۲-۳: نمودار Sequence ورود به سایت	۱۳
شکل ۲-۴: نمودار Sequence ایجاد اکانت اینترنت	۱۳
شکل ۲-۵: نمودار Sequence ایجاد پروفایل اینترنت	۱۴
شکل ۲-۶: نمودار Sequence ایجاد محدودیت برای بسته های اینترنت	۱۴
شکل ۲-۷: نمودار Sequence اضافه کردن روتر	۱۵
شکل ۲-۸: نمودار Sequence حذف اطلاعات	۱۶
شکل ۲-۹: نمودار Class سایت	۱۶
شکل ۴-۱: صفحه ورود به سایت	۲۴
شکل ۴-۲: صفحه اصلی	۲۵
شکل ۴-۳: صفحه فهرست کاربران	۲۶
شکل ۴-۴: فرم ایجاد کاربر	۲۶
شکل ۴-۵: صفحه کاربران فعال	۲۷
شکل ۴-۶: صفحه کاربران پر مصرف	۲۸
شکل ۴-۷: صفحه کاربران مشکوک	۲۸
شکل ۴-۸: صفحه کاربران مسدود شده	۲۹
شکل ۴-۹: صفحه پروفایل	۳۰
شکل ۴-۱۰: فرم ایجاد پروفایل	۳۰
شکل ۴-۱۱: صفحه لیست محدودیت ها	۳۱
شکل ۴-۱۲: فرم ایجاد محدودیت	۳۱
شکل ۴-۱۳: صفحه جست و جو بر اساس نشست ها	۳۲
شکل ۴-۱۴: صفحه جست و جو بر اساس کاربر	۳۳
شکل ۴-۱۵: صفحه روترها	۳۳
شکل ۴-۱۶: فرم افزودن روتر	۳۴
شکل ۴-۱۷: صفحه پنل پذیرش	۳۵

- شکل ۱۸-۴: صفحه ورود مدیریت پایگاه داده و کاربران ۳۵
- شکل ۱۹-۴: صفحه ادمین مدیریت پایگاه داده ۳۶
- شکل ۲۰-۴: صفحه کاربران ۳۶
- شکل ۲۱-۴: صفحه ساخت کاربران ۳۷

فهرست جداول

صفحه	جدول
۱۹.....	۳-۱: Username جدول
۱۹.....	۳-۲: Active User جدول
۲۰.....	۳-۳: Customer جدول
۲۰.....	۳-۴: IP Binding جدول
۲۰.....	۳-۵: Limitation جدول
۲۱.....	۳-۶: Profile Limitation جدول
۲۱.....	۳-۷: Profile جدول
۲۲.....	۳-۸: Router جدول
۲۲.....	۳-۹: Session جدول
۲۲.....	۳-۱۰: Mac Address جدول

مقدمه

با توجه به استفاده روزانه و دائمی مردم از شبکه اینترنت، کنترل و مدیریت مصرف اینترنت یکی از مهم ترین و اصلی ترین دغدغه های مدیریتی در هر سازمان و یا موسسه محسوب می شود. به همین دلیل استفاده از سیستم و روشی که بتواند نظارت و کنترلی بر این استفاده داشته باشد و همچنین تمامی جوانب آن را پوشش دهد همواره از دغدغه های مدیران بوده است. نرم افزار های اکانتینگ و مدیریت استفاده اینترنت به عنوان راه حلی برای رفع این مشکلات به وجود آمده اند. با استفاده از این سیستم های اکانتینگ مدیران قادر خواهند بود تا بصورت دائم و صحیح و کار آمد به نظارت و مدیریت مصرف اینترنت کاربران بپردازند.

در روتر های میکروتیک ابزاری به عنوان Usermanager یا مدیریت کاربران وجود دارد که با استفاده از این ابزار کاربر مدیر قادر خواهد بود تا به مدیریت کاربران اینترنت بپردازد. امکانات اصلی این ابزار شامل ساخت کاربر های اینترنت بصورت تکی و گروهی بصورت نامحدود، ایجاد بسته های اینترنت و محدودیت های حجم بسته و سرعت دانلود و آپلود، جست و جو پیشرفته، امکان اضافه کردن روتر می باشد. اما با توجه به پیچیده بودن این پنل و همچنین کاربری سخت و ظاهری نه چندان زیبا، باعث شده تا طراحی پنل های مجزا با امکانات بیشتر و با کاربری ساده تر در دست اقدام قرار گیرد.

اهداف پروژه و امکانات آن

۱-۱. شرح کلی پروژه

هدف از این پروژه طراحی وب سایتی کاربردی و قدرتمند در زمینه مدیریت کاربران اینترنت در روتر های میکروتیک می باشد که با استفاده از این برنامه می توان به راحتی مصرف اینترنت توسط کاربران و همچنین میزان استفاده از اکانت های مختلف را بدست آورد. این پروژه شامل دو پنل کاربر مدیر و کاربران عادی می باشد که در پنل مدیریت کاربر می تواند به لیست تمام اکانت ها و بسته های اینترنت ساخته شده دسترسی داشته باشد و همچنین بر حسب نیاز اکانت اینترنت ایجاد کند و یا بسته اینترنت بسازد. این پنل شامل جست و جو پیشرفته می باشد که کاربر بر اساس نام اکانت یا مک آدرس کاربر و ... می تواند ریز مصرف کاربر را مشاهده کند این پنل همچنین شامل کاربران پرمصرف یا کاربران مشکوک نیز می باشد که می توان در آنها می توان لیست کاربران پرمصرف یا کاربرانی که در بازه زمانی خاص بیش از یک اکانت استفاده کرده اند را مشاهده نمود. در پنل کاربران عادی کاربر با وارد کردن نام کاربری اکانت صادر شده میتواند میزان مصرف کاربر و تمام نشست هایی که بر روی این اکانت قرار گرفته را مشاهده نمایند.

۱-۲. اهداف پروژه

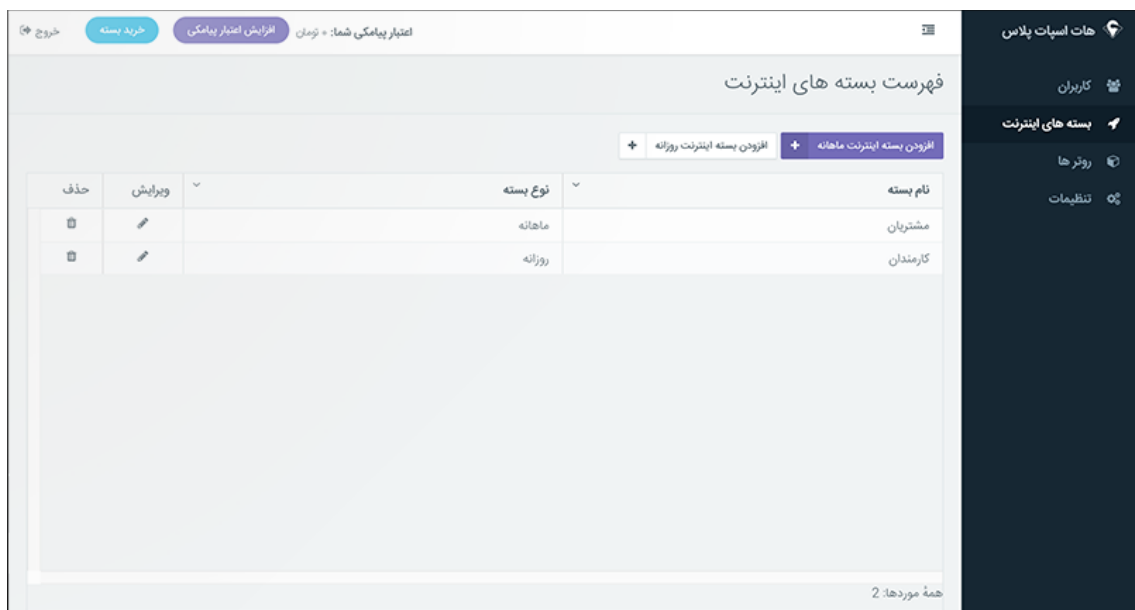
با توجه به اینکه کار با پنل اصلی مدیریت کاربران میکروتیک بسیار پیچیده می باشد و شاید کاربران به راحتی نتوانند از امکانات آن استفاده کنند و همچنین به دلیل کم بودن نمونه فارسی این پنل و با توجه با نیاز های مطرح شده از سمت کاربران و عدم وجود این نیاز ها در نسخه اصلی باعث شده تا ایده طراحی و پیاده سازی این پروژه ایجاد شده و پیاده سازی آن به پایان برسد.

۱-۳. بررسی کارهای مشابه

در ادامه به بررسی اجمالی سه پروژه مشابه در زمینه اکانتینگ و مدیریت مصرف کاربران اینترنت خواهیم پرداخت.

۱-۳-۱. نرم افزار اکانتینگ هات اسپات پلاس

پروژه هات اسپات پلاس یکی از نرم افزار های مدیریت کاربران اینترنت می باشد که به مشتریان خود این امکان را می دهد تا یک شبکه وای فای بصورت هات اسپات راه اندازی کرده و مصرف اینترنت کاربران را مدیریت کنند. با توجه به شکل ۱-۱ می توان امکانات کلی این نرم افزار را مشاهده نمود.



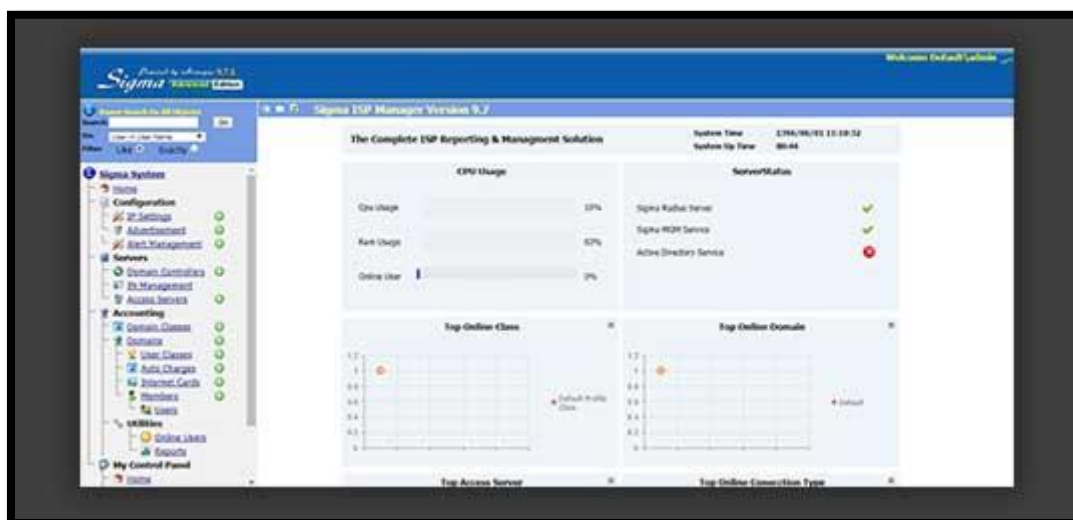
شکل ۱-۱: نرم افزار اکانتینگ هات اسپات پلاس

امکانات نرم افزار

- ✓ مدیریت مصرف اینترنت
- ✓ تایید هویت پیامکی
- ✓ عدم نیاز به سرور
- ✓ درگاه پرداخت اختصاصی
- ✓ لاگ بازدید سایت
- ✓ تغییر فرم عضویت
- ✓ ارسال کد تخفیف
- ✓ پنل اختصاصی کاربران

۲-۳-۱. نرم افزار اکانتینگ سیگما

سیگما، یک AAA Server قدرتمند است که وظیفه احراز هویت، مدیریت مجوزها و محدودیت ها و اکانتینگ اینترنت در در شبکه های مختلف را بر عهده دارد. با توجه به شکل ۱-۲ می توان امکانات وسیع این نرم افزار در حوزه مدیریت کاربران را مشاهده نمود.



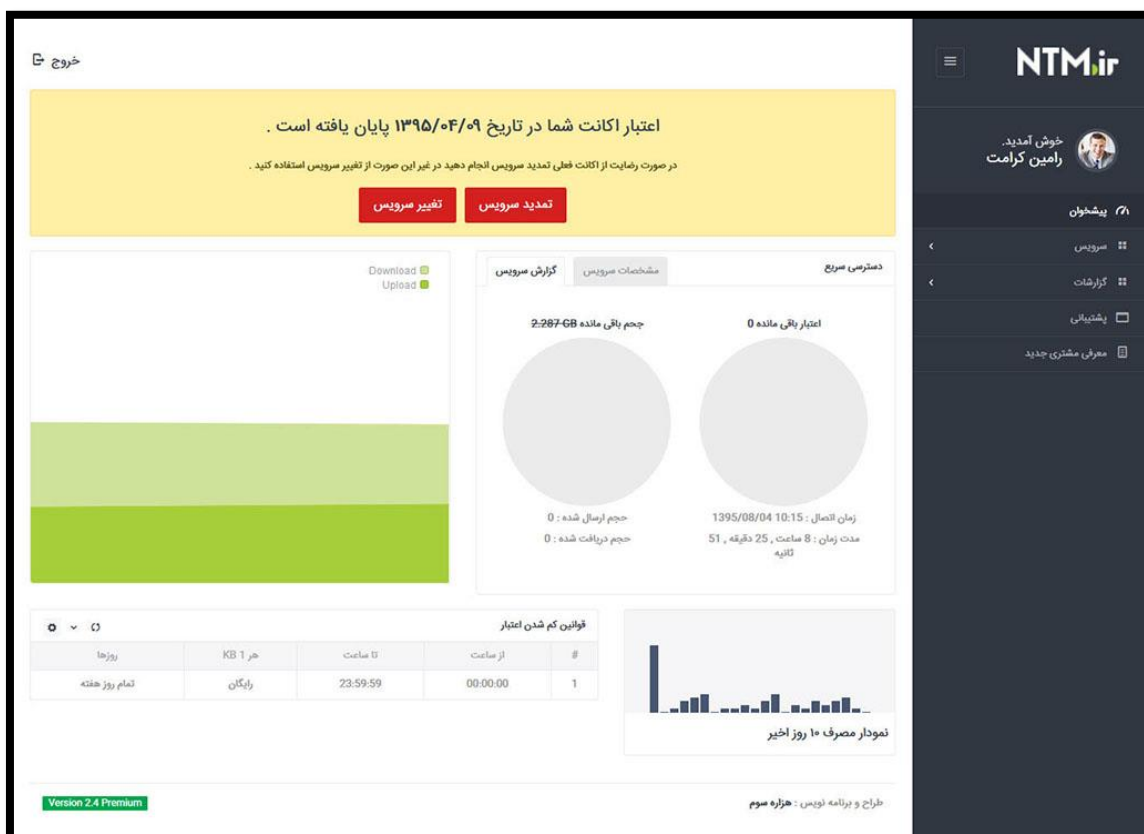
شکل ۲-۱: نرم افزار اکانتینگ سیگما

امکانات

- ✓ دریافت گزارشات اطلاعاتی، تحلیلی، آماری و لحظه ای از سایت های بازدید شده کاربران
- ✓ قابلیت یکپارچگی از طریق API با سایر نرم افزارهای سازمان
- ✓ امکان درخواست افزایش اعتبار توسط کاربر از طریق وب، پیام کوتاه و پرداخت الکترونیک
- ✓ درخواست افزایش اعتبار اینترنت از طریق درگاه پیام کوتاه
- ✓ مشاهده باقی مانده اعتبار توسط کاربر از طریق صفحه آتی هات اسپات

۳-۳-۱. نرم افزار اکانتینگ NTM

نرم افزار اکانتینگ مدیریت و حسابداری کاربران Servco ، FCP ویژه سرویس های اینترنت در شبکه می باشد که از هسته ی اکانتینگ IBS استفاده شده است. این سیستم به صورت یکپارچه سرویس های (Wireless ، LAN Accounting (PPTP , PPPOE , Hotspot , DHCP , Wireless ، ADSL را کنترل می کند و همیشه از پهنای باند خود بهترین استفاده را کرده است. سیستم NTM کاملاً تحت وب می باشند و گزارشات و نمودارهای متنوع آنها، سرویس دهندگان اینترنت را در هر چه بهتر کردن و بهینه کردن سرویس هایشان راهنمایی می نماید. طراحی زیبا و کاربری آسان این نرم افزار در شکل ۳-۱ قابل مشاهده می باشد.



شکل ۳-۱: نرم افزار اکانتینگ NTM

امکانات

- ✓ ثبت و نگهداری اطلاعات کامل مشتریان و دیواس ها
- ✓ لاگ گیری از IP ها برای سازمان تنظیم مقررات
- ✓ ارسال خودکار کاربران پایان اعتبار حجم و تاریخ به گروه منقضی
- ✓ شارژ خودکار اکانت کاربر به محض پرداخت صورت حساب خود از طریق درگاه های آنلاین بانکی
- ✓ گزارشگیری از وب سایتهای مشاهده شده کل سیستم ویا برای هرکاربر بر اساس بازه ی زمانی دلخواه
- ✓ امکان تغییر سرویس ، سرعت و افزایش حجم کاربر
- ✓ مدیریت سرویس های متنوع حجمی و زمانی اختصاصی و اشتراکی
- ✓ تست شده تا ۱۰ هزار کاربر آنلاین همزمان

۴-۱. سطوح دسترسی برنامه

افرادی که از این نرم افزار استفاده می کنند مشمول یکی از حالت های مدیر که شامل مدیر آی تی مجموعه یا مدیر مجموعه، کاربر عادی و فرد مشغول به کار در آن مجموعه که در هتل ها به عنوان کارمند پذیرش است.

کاربران عادی: کاربرانی که بعد از لاگین در نرم افزار امکان جست و جو ریز مصرف اکانت صادر شده را دارا می باشند.

مدیر: کاربر مدیر کسی است که با توجه به دسترسی به قسمت مدیریت پنل امکان گزارش گیری یا ایجاد کاربر یا ایجاد بسته های اینترنت یا محدود سازی آنها و بسیاری از امکانات دیگر را بر عهده دارد.

۵-۱. امکانات نهایی پروژه

امکانات اصلی و کلیدی پروژه شامل موارد زیر می باشد:

- ✓ ایجاد اکانت اینترنت به تعداد نامحدود
- ✓ ایجاد بسته اینترنت
- ✓ ایجاد محدودیت بر اساس سرعت و حجم مصرفی و یا مدت استفاده برای بسته اینترنت
- ✓ امکان اضافه کردن روتر به پنل اصلی و انجام تنظیمات آن
- ✓ امکان مشاهده کاربران پر مصرف در بازه زمانی خاص
- ✓ امکان مشاهده کاربران مشکوک در بازه زمانی خاص
- ✓ امکان مشاهده کلیه نشست ها بوجود آمده برای اکانت خاص یا کلیه اکانت ها
- ✓ امکان مشاهده کاربران آنلاین
- ✓ امکان جست و جو کاربران بر اساس نام کاربری، مک آدرس و یا آی پی
- ✓ امکان مشاهده ترافیک ورودی و خروجی از تمامی پورت های روتر بصورت روزانه، هفتگی، ماهیانه و سالیانه

۶-۱. ابزار مورد استفاده

ابزارهای مورد استفاده در طراحی و ساخت این پروژه به شرح زیر می باشد:

Python: پایتون یک زبان قدرتمند سطح بالا، شی گرا و در عین حال ساده در حوزه برنامه نویسی می باشد.

PyCharm: یک IDE محبوب و کارآمد جهت برنامه نویسی برای زبان پایتون می باشد.

Django: فریم ورک قدرتمند پایتون در حوزه طراحی وب سایت می باشد که کار را بیش از پیش برای برنامه نویسان حوزه وب آسان ساخته است.

HTML: زبان طراحی استاندارد وب سایت می باشد که تمامی کد های سمت سرور و سمت کلاینت در نهایت به کد های HTML تبدیل شده و به کاربر نمایش داده می شود.

CSS: زبان طراحی وب سایت و مکملی برای برای زبان HTML می باشد که باعث شده تا ظاهر وب سایت بصورت دلخواه و زیبا طراحی شود.

Java Script: جاوا اسکریپت یک زبان سطح بالا، شی گرا و پویا می باشد که در طراحی وب سایت کاربرد فراوانی دارد.

Sqlite^۳: یک سیستم ذخیره اطلاعات سبک و قدرتمند می باشد که برای ذخیره و بازیابی اطلاعات به عنوان پایگاه داده مورد استفاده قرار می گیرد.

بررسی پروژه از دید مهندسی

نرم افزار

۲-۱. مقدمه ای بر مهندسی نرم افزار

چندین تعریف متفاوت برای مهندسی نرم افزار وجود دارد که شامل موارد زیر می باشند:

تعریف مهندسی نرم افزار از دیدگاه پرفسور سامرویل: مهندسی نرم افزار یک نظام

مهندسی است که با جنبه های تولید نرم افزار سر و کار دارد. [۱]

تعریف مهندسی نرم افزار فرتیز باور: تصویب اصول مهندسی و استفاده از آنها برای به دست

آوردن یک نرم افزار مقرون به صرفه که قابل اطمینان بوده و روی ماشین واقعی به طور کارآمدی

اجرا شود. [۱]

تعریف مهندسی نرم افزار از دیدگاه IEEE: کاربرد یک روش سیستماتیک عمل و کمیت

پذیر در توسعه، راه اندازی و نگهداری نرم افزار، یعنی استفاده از مهندسی نرم افزار و مطالعه روش

های مختلفی که این خصوصیات را داشته باشند. [۱]

۲-۲. تعریف UML

زبان مدل سازی یکنواخت (unified modeling language) یا UML، یک زبان استاندارد در مورد بیان روابط، ویژگی ها، نمایش گرافیکی و مستند سازی اجزای یک سیستم نرم افزاری می باشد. آنچه که در نموداری های مختلف UML دیده می شود باعث ایجاد دیدگاه های گوناگون در مورد سیستم مورد استفاده است. از ویژگی های دیگر این زبان می توان به عدم وابستگی به زبان های دیگر نام برد.

۲-۲-۱. نمودار کلاس (Class Diagram)

در این نمودار ساختار سیستم توسط کلاس ها، ویژگی های کلاس های مورد نظر و همچنین روابط بین آن ها نمایش داده می شود. نمودار کلاس اصلی ترین جزء سازنده در مدلسازی شی گرایی می باشد، این نمودار هم برای مدلسازی مفهومی کلی برنامه و هم برای مدلسازی طراحی جزئیات برای ترجمه به کد برنامه نویسی به کار می رود. یک کلاس در این نمودار به صورت جعبه مستطیلی نمایش داده می شود که از سه بخش تشکیل شده است.

- قسمت بالایی نام کلاس را نمایش می‌دهد،
- قسمت میانی خصوصیات کلاس را نمایش می‌دهد،
- قسمت پایینی شامل متدها یا عملیاتی است که کلاس انجام می‌دهد یا بر روی آن انجام می‌شود.

۲-۲-۲. نمودار مورد کاربرد (Use Case Diagram)

این نمودار نحوه عملکرد سیستم را نشان می‌دهد در واقع این نمودار کارکرد یک سیستم را در قالب عملگرها (Actor) و اهداف سیستم که بصورت مورد کاربرد نشان داده می‌شود و نیز به وابستگی بین مورد های کاربرد می‌پردازد، به همین دلیل برای شناسایی اهداف سیستم از نمودار مورد کاربرد استفاده می‌شود.

۲-۲-۳. نمودار توالی (Sequence Diagram)

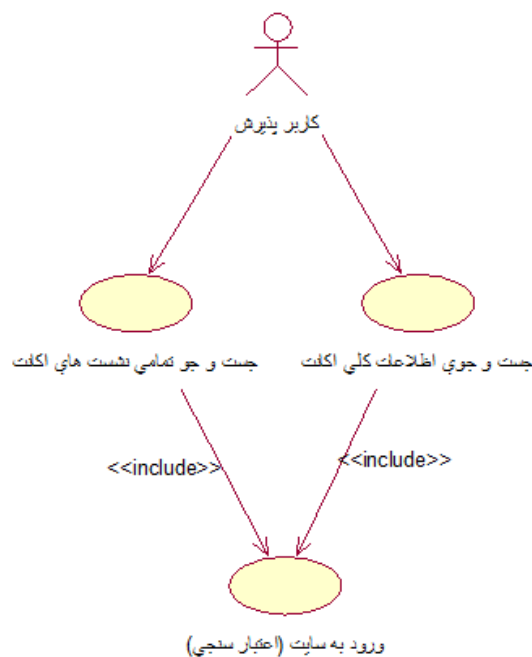
نمودار توالی یکی از نمودارهای Interaction می‌باشد که روندی در یک Use case را مرحله به مرحله نشان می‌دهد. همان طور که می‌توان از اسم آن فهمید، این نمودار با توالی سروکار دارد که از دنباله یا رشته ی پست سرهم از پیام ها که از یک شی به شی دیگر سرازیر می‌شوند (جریان دارند)، تشکیل می‌شود.

۲-۳. نمودار های پروژه

تحلیل و بررسی این پروژه با استفاده از ساختار UML و با استفاده از نرم افزار رشنال رز انجام شده است که در ادامه به بررسی آنها خواهیم پرداخت.

۲-۳-۱. نمودار Use Case کاربر پذیرش

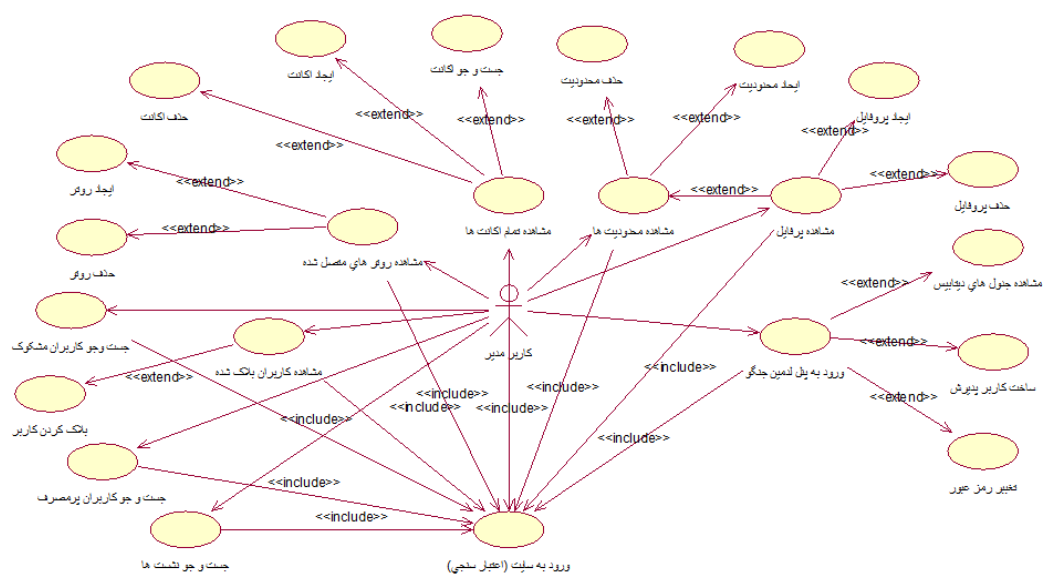
همان طور که در شکل ۱-۲ مشاهده می‌شود، این نمودار امکاناتی که در اختیار کاربر پذیرش وجود دارد را نشان می‌دهد. امکاناتی چون جست و جو اطلاعات کلی کاربر و ریز مصرف کاربر در اختیار کاربر پذیرش وجود دارد.



شکل ۱-۲: نمودار Use Case کاربر پذیرش

۲-۳-۲. نمودار Use Case کاربر مدیر

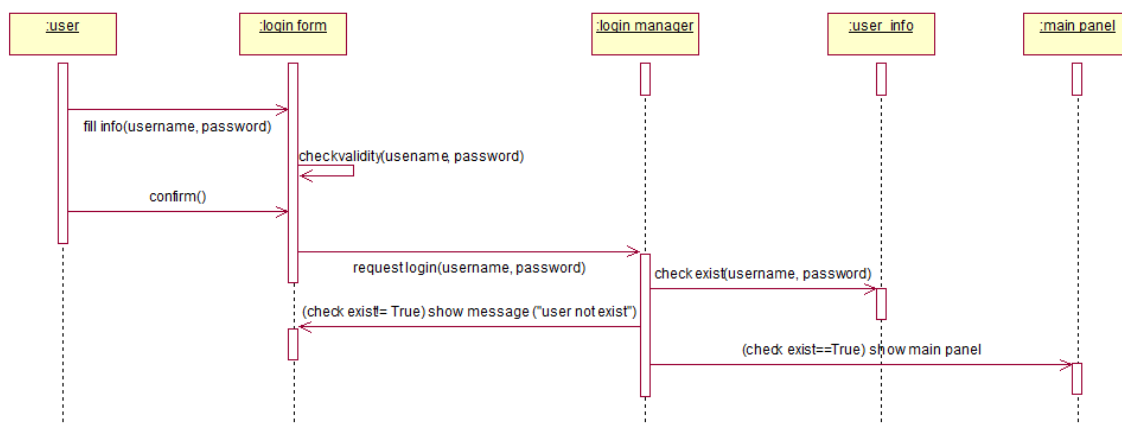
این نمودار نیز امکاناتی که در اختیار مدیر یا ادمین شبکه وجود دارد نشان می دهد. با توجه به شکل ۲-۲ امکاناتی از قبیل ایجاد اکانت، ایجاد پروفایل یا بسته اینترنتی، جست و جو پیشرفته و ... در اختیار مدیر قرار دارد.



شکل ۲-۲: نمودار Use Case کاربر مدیر

۲-۳-۳. نمودار sequence ورود به سایت

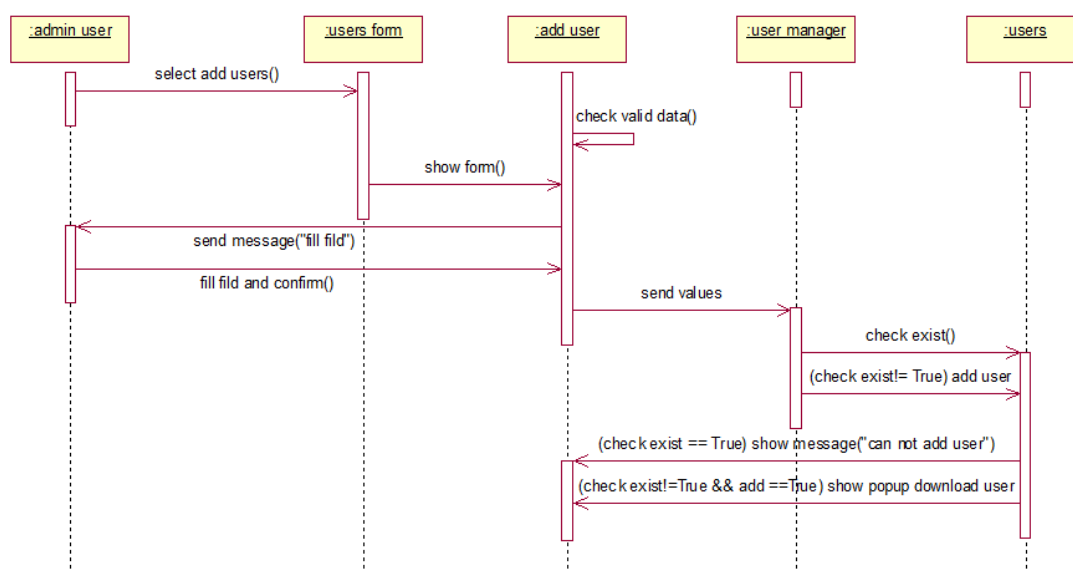
با توجه به شکل ۲-۳ می توان مشاهده نمود که این نمودار توالی و روال لازم برای ورود به سایت و استفاده از امکانات سایت را نشان می دهد.



شکل ۲-۳: نمودار Sequence ورود به سایت

۲-۳-۴. نمودار sequence ایجاد اکانت اینترنت

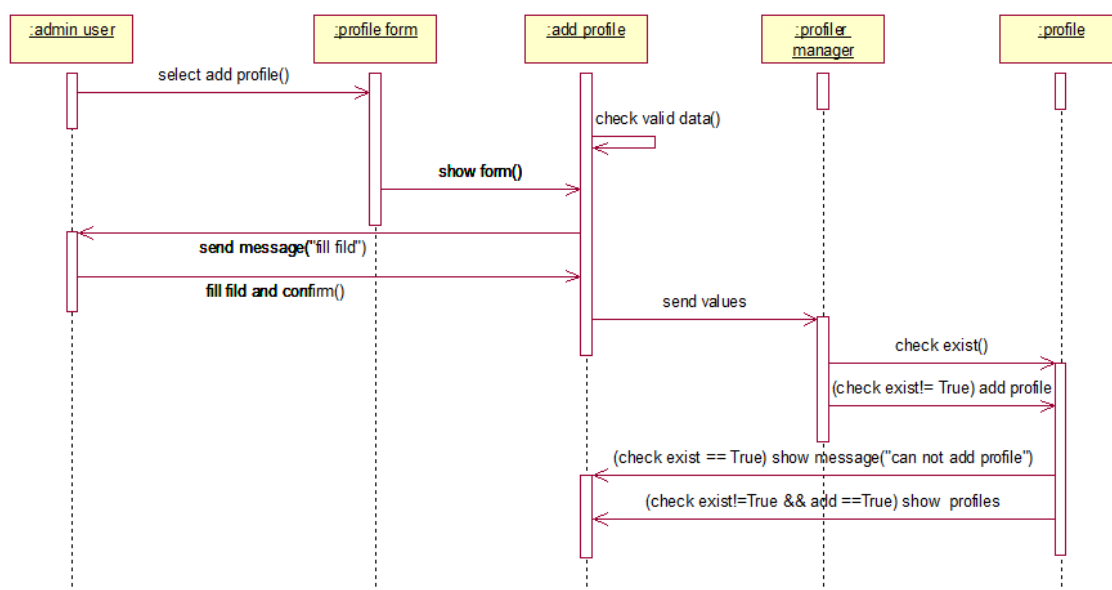
همان طور که در شکل ۲-۴ قابل مشاهده است، این نمودار توالی و روال لازم برای ایجاد اکانت توسط مدیر را نشان می دهد.



شکل ۲-۴: نمودار Sequence ایجاد اکانت اینترنت

۵-۳-۲. نمودار sequence ایجاد پروفایل اینترنت

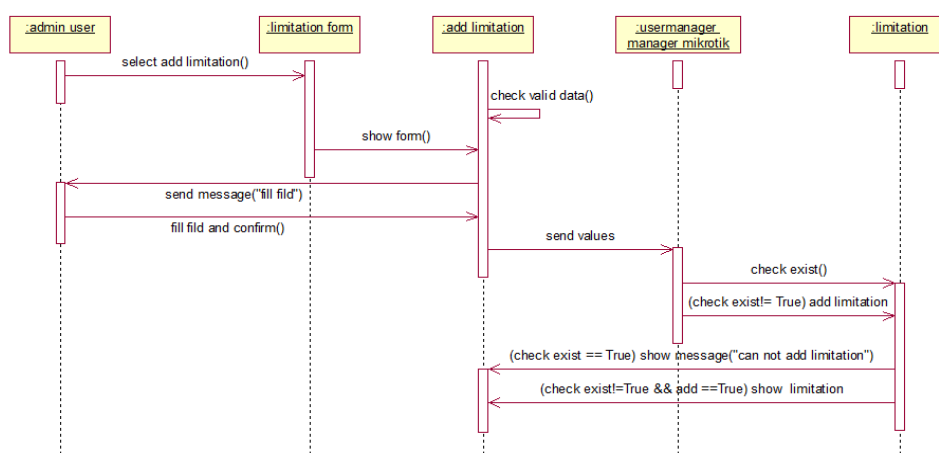
در شکل ۵-۲ توالی و روال لازم برای ایجاد پروفایل یا بسته اینترنتی توسط مدیر را نشان داده شده است.



شکل ۵-۲: نمودار Sequence ایجاد پروفایل اینترنت

۶-۳-۲. نمودار sequence ایجاد محدودیت برای بسته های اینترنت

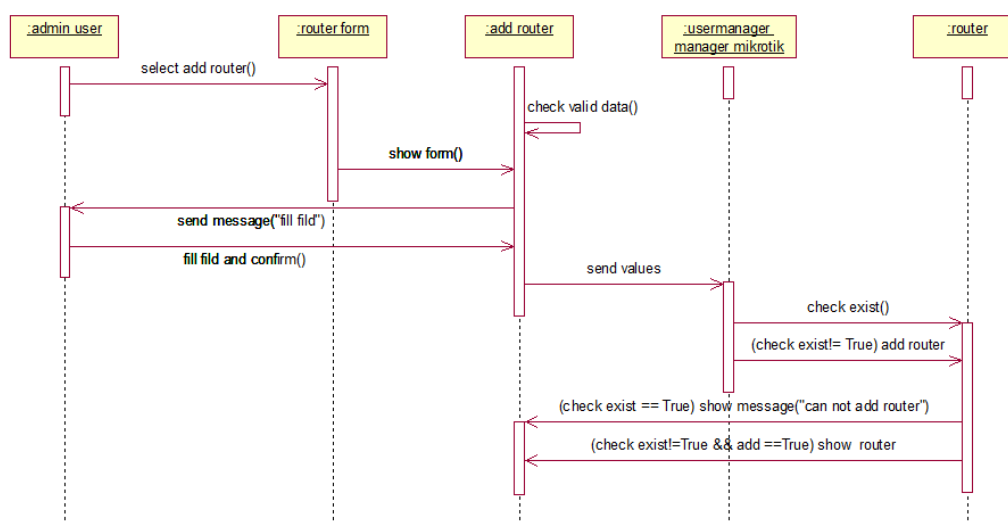
این نمودار توالی و روال لازم برای ایجاد محدودیت برای بسته های اینترنت توسط مدیر را نشان می دهد، در شکل ۶-۲ این توالی قابل مشاهده است.



شکل ۶-۲: نمودار Sequence ایجاد محدودیت برای بسته های اینترنت

۲-۳-۷. نمودار sequence اضافه کردن روتر

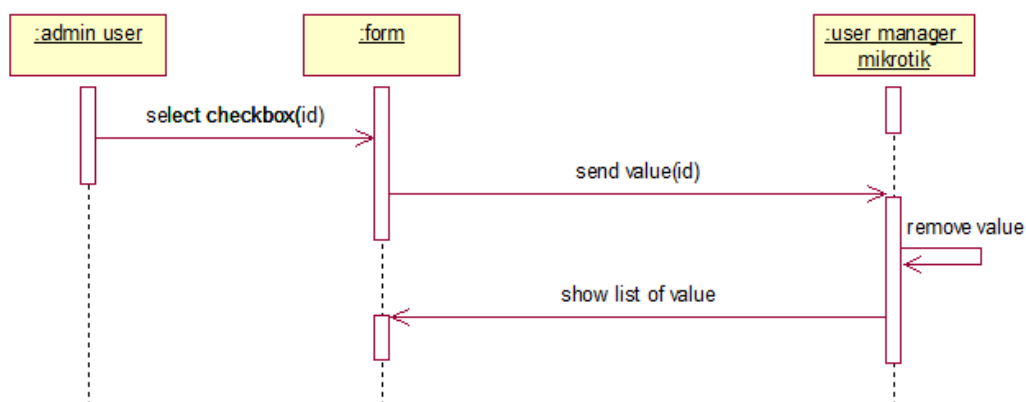
شکل ۷-۲ توالی و روال لازم برای اضافه کردن روتر و اتصال آن به برنامه را توسط مدیر نشان می دهد.



شکل ۷-۲: نمودار Sequence اضافه کردن روتر

۲-۳-۸. نمودار sequence حذف اطلاعات

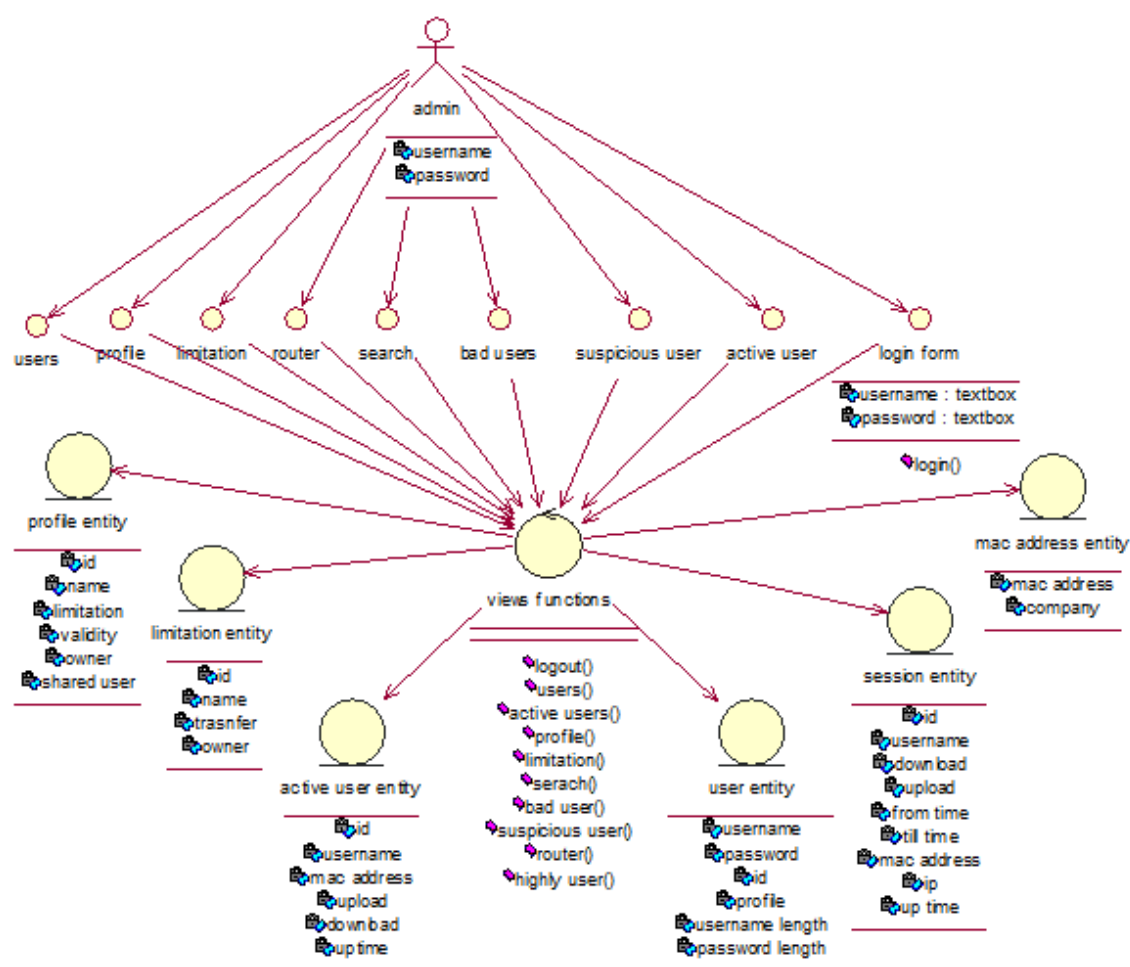
همان طور که در شکل ۸-۲ نشان داده شده است، این نمودار توالی و روال لازم برای حذف اطلاعاتی از قبیل اکانت اینترنت، پروفایل، محدودیت بسته های اینترنت و ... توسط مدیر را نشان می دهد.



شکل ۸-۲: نمودار Sequence حذف اطلاعات

۸-۳-۲. نمودار Class سایت

با توجه به شکل ۲-۹ این نمودار نشان دهنده ساختار سیستم با نشان دادن کلاس ها، خصوصیات کلاس ها و روابط بین آنها می باشد.



شکل ۹-۲. نمودار Class سایت

بررسی پایگاه داده پروژه

۳-۱. پایگاه داده

پایگاه داده مجموعه ای داده های ذخیره شده، پایا (مانا)، به هم مرتبط، مجتمع، حد الامکان فاقد افزونگی دارای معماری خاص خود، مبتنی بر یک مدل داده ای تحت مدیریت یک سیستم کنترل متمرکز، مورد استفاده یک یا چند کاربر، از یک محیط بصورت اشتراکی یا همزمان می باشد. [۲]

یا به عبارت دیگر سیستمی که مجموعه ای از اطلاعات را بصورت سازماندهی شده و منظم در خود نگه می دارد.

۳-۲. سیستم مدیریت پایگاه داده

سیستم مدیریت پایگاه داده یا به اختصار DBMS یک نرم افزار واسط بین برنامه های کاربران و محیط داخلی و فیزیکی ذخیره سازی است. این نرم افزار این امکان را به کاربران می دهد تا در یک محیط انتزاعی کار کنند و در عین حال به داده های ذخیره شده دسترسی داشته و عملیات مورد نظر خود را انجام دهند. با استفاده از این سیستم سرعت دسترسی به داده ها بالا تر می رود و امکان استفاده اشتراکی از داده ها وجود دارد. همچنین با استفاده از این سیستم امکان کنترل متمرکز روی تمام داده های عملیاتی وجود دارد و یکی از مزایای این کنترل متمرکز کاهش میزان افزودگی در ذخیره سازی داده ها است. [۳]

۳-۲-۱. انواع سیستم های مدیریت پایگاه داده

از معروف ترین نرم افزارهای مدیریت پایگاه داده می توان به موارد زیر اشاره کرد:

Oracle ✓

Microsoft SQL Server ✓

My SQL ✓

در این پروژه از SQLite به عنوان سیستم مدیریت پایگاه داده استفاده شده است.

۳-۳. SQLite چیست؟

SQLite یک سیستم مدیریت و ذخیره سازی اطلاعات به زبان سی می باشد. این سیستم بصورت یک کتابخانه و بسیار سبک و در عین حال قدرتمند در اختیار کاربران قرار گرفته است. وابسته نبودن به سیستم عامل خاص، کم حجم بودن، دارای محیط مدیریتی از ویژگی های چشم گیر این سیستم می باشد.

۳-۴. جداول پایگاه داده پروژه

در این پروژه با توجه به اینکه تمامی اطلاعات از روتر میکروتیک گرفته می شود اکثر جداول پایگاه داده درون خود روتر وجود دارند و این برنامه اطلاعات مورد نظر خود را از این جداول بدست می آورد.

۳-۴-۱. جدول Username

جدول مربوط به کاربران اینترنت در شکل ۳-۱ نشان داده شده است. این جدول مربوط به اکانت های صادر شده توسط مدیر می باشد. در این جدول مواردی چون نام کاربری، پسورد، سازنده، تعداد به اشتراک گذاشتن اکانت بصورت هم زمان، آخرین استفاده از اکانت و تاریخ ایجاد اکانت ذخیره و Id به عنوان کلید اصلی در نظر گرفته می شود.

۳-۱: جدول Username

ID	CUSTOMER	USERNAME	PASSWORD	SHARED USERS	LAST SEEN	DATE CREATED
*FED	admin	1gc7t	5m8	5	never	Dec. 30, 2019, 5:18 p.m.
*F8C	admin	1gaqh	hmv	5	never	Dec. 30, 2019, 5:18 p.m.
*F88	admin	1g5xx	hfn	5	never	Dec. 30, 2019, 5:18 p.m.
*F86	admin	1gvsf	s9p	5	never	Dec. 30, 2019, 5:18 p.m.
*F85	admin	1ghwf	vww	5	never	Dec. 30, 2019, 5:18 p.m.
*F79	admin	1g2ws	bq3	5	never	Dec. 30, 2019, 5:18 p.m.
*F74	admin	1gwjy	sn7	5	never	Dec. 30, 2019, 5:18 p.m.
*F38	admin	1gtbj	xfb	5	never	Dec. 30, 2019, 5:18 p.m.
*F1E	admin	1gsqz	dj9	5	never	Dec. 30, 2019, 5:18 p.m.
*F09	admin	1axm9	t9m	5	never	Dec. 30, 2019, 5:18 p.m.

۳-۴-۲. جدول Active User

با توجه به شکل ۳-۲، این جدول مربوط به کاربران فعالی می باشند که در زمان حال به روتر متصل بوده و از اینترنت اسفاده می کنند. مواردی چون سرور، نام اکانت وارد شده، آی پی، مک آدرس، پروتکل لاگین، مدت زمان استفاده از اکانت، مدت زمان باقی مانده جهت گرفتن آی پی از کاربر در صورت قطع ارتباط کاربر با روتر، مقدار دانلود و آپلود کاربر و مقدار بسته های ورودی و خروجی ذخیره و Id به عنوان کلید اصلی در نظر گرفته می شود.

۳-۲: جدول Active User

ID	SERVER	USER	ADDRESS	MAC ADDRESS	LOGIN BY	UPTIME	IDLE TIME	KEEPAIVE TIMEOUT	BYTES IN	BYTES OUT	PACKETS IN	PACKETS OUT
*C0A80AFD	hotspot1	admin	192.168.10.253	28:D2:44:DD:E5:A0	http-chap	1m25s	1m6s	2m	12831	1600	115	20

1 active user

۳-۴-۳. جدول Customer

با توجه به شکل ۳-۳، این جدول مربوط به کاربران یوزر منیجر میکروتیک می باشد که این کاربران قادر هستند به پنل اصلی یوزر منیجر میکروتیک دسترسی داشته باشند. در این جدول اطلاعاتی چون نام کاربری، رمز عبور، میزان دسترسی کاربر به امکانات یوزر منیجر و دسترسی های کلی کاربر ذخیره و Id به عنوان کلید اصلی در نظر گرفته می شود.

۳-۴: جدول Customer

<input type="checkbox"/>	ID	LOGIN	PASSWORD	ACCESS	PERMISSIONS
<input type="checkbox"/>	*3	user	123123	own-routers,own-users,own-profiles,own-limits,config-payment-gw,parent-routers,parent-users,parent-profiles,parent-limits,parent-payment-gw	read-only
<input type="checkbox"/>	*2	jahan	333212	own-routers,own-users,own-profiles,own-limits,config-payment-gw,parent-routers,parent-users,parent-profiles,parent-limits,parent-payment-gw	full
<input type="checkbox"/>	*1	admin	1119000	own-routers,own-users,own-profiles,own-limits,config-payment-gw	owner

3 customers

۳-۴-۴. جدول IP Binding

این جدول مربوط به کاربرانی است که بلاک شده اند و یا اینترنت آنها بدون محدودیت می باشد. در این جدول اطلاعاتی چون آی پی، مک آدرس، نوع ، حالت بلاک و حالت فعال بودن ذخیره و Id به عنوان کلید اصلی در نظر گرفته می شود. این اطلاعات در شکل ۴-۳ قابل مشاهده می باشد.

۳-۴: جدول IP Binding

<input type="checkbox"/>	ID	ADDRESS	MAC ADDRESS	TYPE	BLOCKED	DISABLED
<input type="checkbox"/>	*1	192.168.1.11	B8:D7:AF:73:59:6B	blocked	blocked	false

1 ip binding

۳-۴-۵. جدول Limitation

این جدول مربوط به اطلاعات محدودیت های بسته های اینترنت می باشد. اطلاعات ذخیره شده در این جدول شامل نام محدودیت، نرخ دانلود، نرخ آپلود، ترافیک کلی و مدت زمان استفاده می باشد. Id به عنوان کلید اصلی در نظر گرفته می شود. در شکل ۵-۳ جدول مربوط به محدودیت ها نمایش داده شده است.

۳-۵: جدول Limitation

<input type="checkbox"/>	ID	NAME	OWNER	DOWNLOAD LIMIT	UPLOAD LIMIT	TRANSFER LIMIT	UPTIME LIMIT
<input type="checkbox"/>	*1D	1G	admin	0	0	1000000000	0s
<input type="checkbox"/>	*1A	2 Gig	admin	0	0	2000000000	0s

2 limitations

۳-۴-۶. جدول Profile Limitation

با توجه به شکل ۳-۶ این جدول نیز قسمت دیگری از محدودیت اعمال شده برای بسته های اینترنت یا پروفایل را نشان می دهد. در این جدول مقادیر پروفایل، محدودیت، زمان شروع محدودیت، زمان پایان محدودیت و روز های هفته که این محدودیت مورد اعمال قرار گیرد، ذخیره می شود. Id به عنوان کلید اصلی در نظر گرفته می شود.

۳-۶: جدول Profile Limitation

<input type="checkbox"/>	ID	PROFILE	LIMITATION	FROM TIME	TILL TIME	WEEKDAYS
<input type="checkbox"/>	*1	*2C	2 Gig	0s	23h59m59s	sunday,monday,tuesday,wednesday,thursday,friday,saturday

1 profile limitation

۳-۴-۷. جدول Profile

در این جدول اطلاعات مربوط به پروفایل یا بسته اینترنتی قرار دارد. اطلاعات ذخیره شده در این جدول شامل نام پروفایل، سازنده پروفایل، اعتبار پروفایل، زمان شروع، میزان به اشتراک گذاری پروفایل و قیمت می باشد. Id به عنوان کلید اصلی در نظر گرفته می شود. در شکل ۳-۷ اطلاعات جدول بسته های اینترنت قابل مشاهده می باشد.

۳-۷: جدول Profile

<input type="checkbox"/>	ID	NAME	OWNER	VALIDITY	STARTS AT	OVERRIDE SHARED USERS	PRICE
<input type="checkbox"/>	*2A	2 Gig	admin	2w	login	5	0

1 profile

۳-۴-۸. جدول Router

همان طور که در شکل ۳-۸ مشاهده می شود، این جدول اطلاعات مربوط به روتر های متصل به یوزر منیجر را در خود نگهداری می کند. اطلاعات ذخیره شده در این جدول شامل نام روتر، نام کاربر، آی پی، رمز امنیتی به اشتراک گذاشته شده بین روتر و یوزر منیجر، نوع لاگ گیری، و استفاده از پورت CoA و شماره پورت آن می باشد. Id به عنوان کلید اصلی در نظر گرفته می شود.

۸-۳: جدول Router

<input type="checkbox"/>	ID	NAME	CUSTOMER	IP	SHARED SECRET	LOG	USE COA	COA PORT
<input type="checkbox"/>	*1	Main	admin	127.0.0.1	244532	auth-fail	no	1700
1 router								

۹-۴-۳. جدول Session

در این جدول اطلاعات مربوط به تمامی نشست های کاربران بصورت مداوم ذخیره می شود. اطلاعات این جدول شامل نام سازنده اکانت، نام کاربری اکانت، NAS port و نوع آن، مک آدرس، آی پی، وضعیت اکانت، تاریخ شروع استفاده، تاریخ پایان استفاده، مدت زمان مصرف، مقدار دانلود و آپلود وضعیت فعال بودن اکانت می باشد. Id به عنوان کلید اصلی در نظر گرفته می شود. در شکل ۳-۹ اطلاعات ذکر شده قابل رویت می باشد.

۹-۳: جدول Session

<input type="checkbox"/>	ID	CUSTOMER	USER	NAS PORT	NAS PORT TYPE	CALLING STATION ID	USER IP	STATUS	FROM TIME	TILL TIME	UPTIME	DOWNLOAD	UPLOAD	ACTIVE
<input type="checkbox"/>	*29FE	admin	5g8qw	2150629377	wireless-802.11	00:13:A9:8A:29:E7	172.16.0.1	start,stop	feb/27/2019 14:04:25	feb/27/2019 14:04:50	25s	3303	12795	[active]
<input type="checkbox"/>	*29FD	admin	1mu2	2162163712	wireless-802.11	48:C7:96:08:30:FD	172.16.0.17	start,interim,closed	feb/04/2019 12:15:22	feb/04/2019 12:19:23	4m2s	81239	55339	[active]
<input type="checkbox"/>	*29FC	admin	12wn	2161115160	wireless-802.11	70:F0:87:F0:D3:2A	172.16.0.230	start,interim,closed	feb/04/2019 11:21:23	feb/04/2019 12:08:23	47m1s	39520779	1531756	[active]
<input type="checkbox"/>	*29FB	admin	1y4d	2161115159	wireless-802.11	6C:C7:EC:42:80:1B	172.16.0.250	start,interim,closed	feb/04/2019 11:14:37	feb/04/2019 12:08:37	54m	184002095	6473500	[active]
<input type="checkbox"/>	*29FA	admin	54kb	2161115158	wireless-802.11	F8:38:80:C0:EF:64	172.16.0.238	start,interim,closed	feb/04/2019 11:11:42	feb/04/2019 12:08:43	57m	11511894	2527307	[active]
<input type="checkbox"/>	*29F9	admin	1ekv	2161115157	wireless-802.11	D8:68:C3:D5:2C:0F	172.16.0.228	start,interim,closed	feb/04/2019 09:44:41	feb/04/2019 12:08:46	2h24m5s	225672984	10809993	[active]

۱۰-۴-۳. جدول Mac Address

این جدول شامل دو مقدار نام کمپانی و مک آدرس می باشد. در شکل ۳-۱۰ این اطلاعات را می توان مشاهده نمود.

۱۰-۳: جدول Mac Address

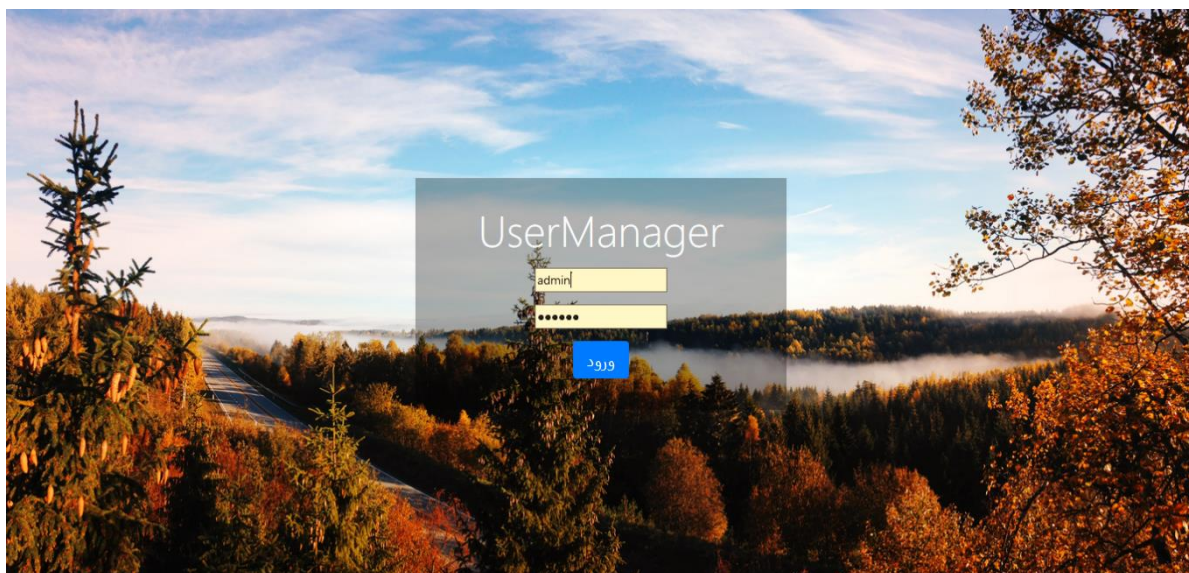
<input type="checkbox"/>	COMPANY	MAC ADDRESS
<input type="checkbox"/>	Aruba	B8:D4:E7
<input type="checkbox"/>	Apple	28:F0:33
<input type="checkbox"/>	Apple	6C:4A:85
<input type="checkbox"/>	Apple	24:D0:DF
<input type="checkbox"/>	Apple	D8:4C:90
<input type="checkbox"/>	Apple	BC:09:63
<input type="checkbox"/>	Shenzhen Skyworth Digital Technology CO.	78:53:0D
<input type="checkbox"/>	Ubee Interactive Co.	6C:38:A1
<input type="checkbox"/>	Ubee Interactive Co.	64:7C:34
<input type="checkbox"/>	Sichuan AI-Link Technology Co.	A4:29:85
<input type="checkbox"/>	HMD Global Oy	00:5E:0C
<input type="checkbox"/>	Intel Corporate	C8:58:C0

توضیح امکانات نرم افزار

در این فصل به بررسی بخش های مختلف برنامه خواهیم پرداخت همچنین بخشی از کد های مهم برنامه در بخش پیوست آمده است.

۱-۴. صفحه ورود به سایت

در صفحه ورود به سایت کاربر باید نام کاربری و رمز عبور خود را وارد کند، کاربر اصلی با دسترسی کامل به سایت دسترسی دارد و کاربر پذیرش تنها قادر خواهد بود یک صفحه برای گزارش گیری مشاهده کند. همان طور که در شکل ۴-۱ مشاهده می شود کاربر باید در دو تکست باکس به ترتیب نام کاربری خود و رمز عبور خود را وارد کرده و با کلیک بر روی دکمه ورود وارد پنل اصلی می شود.

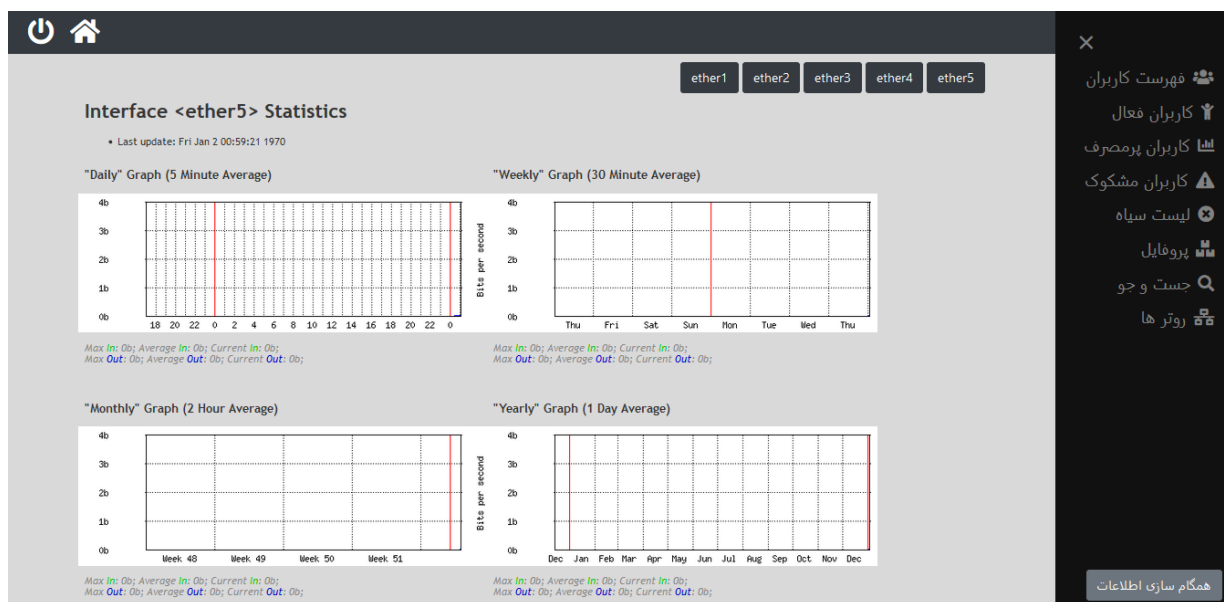


شکل ۱-۴: صفحه ورود به سایت

۲-۴. صفحه اصلی

با توجه به شکل ۴-۲ در این صفحه کاربر ادمین وضعیت ترافیک ورودی و خروجی پورت های روتر را مشاهده می کند که کاربر با کلیک بر روی دکمه های مربوط به هر پورت وضعیت آن پورت را بصورت نموداری مشاهده می کند. در سمت راست صفحه کاربر می تواند به بخش های مختلف سایت دسترسی داشته باشد به عنوان نمونه مدیر با کلیک بر روی فهرست کاربران به صفحه فهرست کاربران هدایت خواهد شد.

با کلیک بر روی دکمه همگام سازی اطلاعات برنامه تمام اطلاعات کاربران ساخته شده را از روتر استخراج کرده و به بانک اطلاعاتی خود منتقل می کند تا جست و جو در میان انبوهی از کاربران به سرعت انجام شود.



شکل ۲-۴: صفحه اصلی

۳-۴. صفحه فهرست کاربران

در این صفحه کاربر ادمین قادر خواهد بود تا تمامی کاربران ساخته شده را مشاهده کند، به جست و جو کاربر مورد نظر خود پردازد، کاربر مورد نظر خود را حذف کند و یا به تعداد دلخواه کاربر ایجاد کند. با توجه به شکل ۴-۳ این صفحه شامل جدول کاربران ساخته شده، دکمه های حذف کاربر جهت حذف کاربر مورد نظر، ایجاد کاربر جهت ساخت کاربر جدید، بازگشت جهت بازگشت به صفحه قبلی و دکمه جست و جو برای جست کاربر می باشد.

۴-۴. صفحه کاربران فعال

با توجه به شکل ۴-۵ در این صفحه کاربر ادمین قادر خواهد بود تا کاربرانی که به روتر وصل هستند و از اینترنت استفاده می کنند مشاهده کند، این صفحه حاوی جدول کاربران فعال می باشد که اطلاعاتی مانند نام کاربر، شرکت سازنده دستگاه کاربر مورد نظر، میزان دانلود و آپلود در آن قرار دارد که کاربر ادمین با انتخاب کاربر مورد نظر و کلیک بر روی دکمه حذف کاربر قادر خواهد بود تا اتصال کاربر با روتر را قطع کند.



آپلود	دانلود	Up Time	شرکت سازنده	مک آدرس	آی پی	نام کاربر	
80 Bytes	2.78 KiloBytes	14s	LCFC(HeFei) Electronics Technology Co.	28:D2:44:DD:E5:A0	192.168.10.254	admin	<input type="checkbox"/>

شکل ۴-۵: صفحه کاربران فعال

۴-۵. صفحه کاربران پرمصرف

در این صفحه کاربر ادمین قادر خواهد بود تا لیست کاربرانی که بیشترین مصرف اینترنت را در بازه زمانی مشخص مشاهده کند، همان طور که در شکل ۴-۶ مشاهده می شود این اطلاعات نمایش داده شده شامل مک آدرس کاربر، شرکت سازنده دستگاه و کل حجم مصرف شده توسط کاربر می باشد. با کلیک بر روی مک آدرس، کاربر ادمین به صفحه نشست های این مک آدرس هدایت می شود.

لیست کاربران پرمصرف		
تعداد کاربر مورد نظر جهت نمایش		
از تاریخ	تا تاریخ	تعداد کاربر مورد نظر جهت نمایش
08/08/2017	01/01/2020	5
بازگشت	🔍	
کل ترافیک مصرفی کاربر در تاریخ مورد نظر	شرکت سازنده	مک آدرس
79.97 GigaBytes	HMD Global Oy	A0:28:ED:8D:0D:2E
25.1 GigaBytes	Lenovo (Beijing) Co.	A0:32:99:CF:AD:DA
13.21 GigaBytes	Samsung Electronics Co.	48:C7:96:08:30:FD
12.89 GigaBytes	Apple	28:A0:2B:9F:70:5A
12.47 GigaBytes	Murata Manufacturing Co.	5C:F8:A1:E9:D4:E7

شکل ۴-۶: صفحه کاربران پر مصرف

۴-۶. صفحه کاربران مشکوک

در این صفحه کاربر ادمین قادر خواهد بود تا لیست کاربرانی که در بازه زمانی مشخص بیشترین اکانت را مختلف را استفاده کرده اند مشاهده کند، این اطلاعات نمایش داده شده شامل مک آدرس کاربر، شرکت سازنده دستگاه و اکانت های استفاده توسط کاربر می باشد. با کلیک بر روی مک آدرس یا اکانت، کاربر ادمین به صفحه نشست های این مک آدرس یا اکانت مورد نظر هدایت می شود. تمامی این اطلاعات در شکل ۴-۷ قابل مشاهده می باشد.

لیست کاربران مشکوک		
کاربرانی که در یک بازه خاص بیش از یک اکانت استفاده کرده اند		
تعداد کاربران : 91		
از تاریخ	تا تاریخ	تعداد کاربر مورد نظر جهت نمایش
01/04/2016	01/01/2020	5
بازگشت	🔍	
نام کاربری اکانت های استفاده شده	شرکت سازنده دستگاه	مک آدرس
1388 1ga9 1ppn 1z9n 1mbs 12zg 1mur 1crs 1g76 16nd 1pr2 1ajd 19jt 1y24 1hb5 1gk9 1565 1mmj 1q6 13ef 15n6 1zki 1jrw 1bsf 1ldp 1b5p 1rmt 1uja 13m3 1uey 1tym 1i3g 1cp3 1sx6 17fs 1g6p 1hq8 1t2v 1kfw 1vg7 1k76 19vt 1gxe 1p5v 1kja 1ael 1g4w 1gsh 1nau 1ppw 1gzn 1p9n 1tau 1xrg 174m 1krf 122p 187b 1akq 1kdb 1qci 162b 18cc 1zbh 1t79 1896 1az2 1iga 1dr2 1cn5 12ff 16rn 1srs 1pym 1rch 13fa 12pw 1h7z 1tud 1cmt 1ib5 12jz 1b86 13zg 13g5 1s4j 1g3q 1xch 136f 19ur 1fti 1qim 1bkk 158t 1bcx 1y6h 13jx 1n6q 1kug 1cwg 1t5d 173s 1sdm 1ypk 1wvg 1bya 1tua	HMD Global Oy	A0:28:ED:8D:0D:2E
1rrq 1wuk 1x5k 1xuf	Apple	C0:F2:F8:B0:5A:E1
1pmv 1hdb 1buc 1q7z	Apple	FC:D8:48:DC:7D:36
1ut7 1sn6 1iu9 1qcv 1ayv	Samsung Electronics Co.	40:D3:AE:EF:63:88
1jcy 1k9i 1z9y 1da8 1j5i 1nk7 1pc4 1542 1xmj 1tzb 1jzt 1a42 1g6q 1wta 1w92 1g9k 14c4 1eg2 1tzw 1ajk 1wq8 16iw 1f2z	Apple	40:98:AD:23:43:6A
1934 1kye 1cgs 1c96 1hbf 1tyt 1844 1wk9	Apple	98:CA:33:E1:3C:E4

شکل ۴-۷: صفحه کاربران مشکوک

۷-۴. صفحه کاربران مسدود شده

همان طور که در شکل ۴-۸ مشاهده می شود در این صفحه کاربر ادمین قادر خواهد بود تا لیست کاربرانی که مک آدرس آنها بلاک شده است مشاهده کند، این اطلاعات نمایش داده شده شامل مک آدرس کاربر و شرکت سازنده دستگاه می باشد همچنین در این صفحه کاربر قادر خواهد بود تا مک آدرس مورد نظر خود را وارد لیست سیاه کند و یا با انتخاب کاربر و کلیک بر روی دکمه خارج کردن از حالت مسدود از این لیست حذف کند. با کلیک بر روی مک آدرس، کاربر ادمین به صفحه نشست های این مک آدرس هدایت می شود.



شکل ۸-۴: صفحه کاربران مسدود شده

۸-۴. صفحه پروفایل

با توجه به شکل ۴-۹ در این صفحه کاربر ادمین می تواند لیست پروفایل ها یا بسته های اینترنتی ساخته مشاهده کند، با انتخاب پروفایل مورد نظر به حذف آن پردازد و یا با کلیک بر روی ایجاد پروفایل، بسته اینترنتی جدید ایجاد کند. همچنین با انتخاب لیست محدودیت ها کاربر به صفحه لیست محدودیت های هدایت می شود.



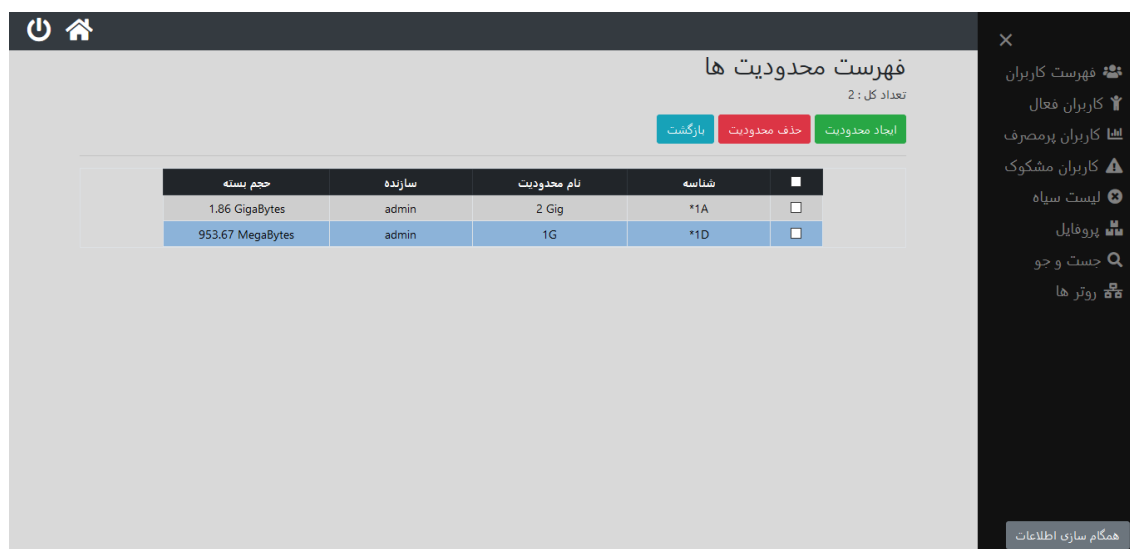
شکل ۹-۴: صفحه پروفایل

با کلیک بر روی ایجاد پروفایل فرم مربوط به ایجاد پروفایل برای کاربر نشان داده می شود، در این فرم کاربر ملزم است تا نام پروفایل، سازنده پروفایل، میزان اعتبار پروفایل، شروع اعتبار، تعداد استفاده هم زمان از اکانت و محدودیت مربوط به پروفایل را پر کند. تمامی این امکانات را می توان در شکل ۴-۱۰ مشاهده نمود.

شکل ۱۰-۴: فرم ایجاد پروفایل

۹-۴. صفحه لیست محدودیت ها

در این صفحه کاربر لیست محدودیت های ایجاد شده را مشاهده می کند، با توجه به شکل ۴-۱۱ در این صفحه کاربر قادر خواهد بود با انتخاب محدودیت مورد نظر به حذف آن بپردازد و یا با کلیک بر روی ایجاد محدودیت، محدودیت جدید ایجاد کند.



شکل ۴-۱۱: صفحه لیست محدودیت ها

با کلیک بر روی دکمه ایجاد محدودیت فرم مربوط به ایجاد محدودیت به کاربر نشان داده می شود، این فرم که در شکل ۴-۱۲ نشان داده شده است حاوی فیلدهای نام محدودیت، سازنده و حجم مورد ترافیک مصرفی برای محدودیت می باشد.

شکل ۴-۱۲: فرم ایجاد محدودیت

۱۰-۴. صفحه جست و جو

در این صفحه کاربر بر اساس نشست ها و یا بر اساس کاربران می تواند به جست و جو بپردازد.

۱۰-۴-۱. جست و جو بر اساس نشست ها

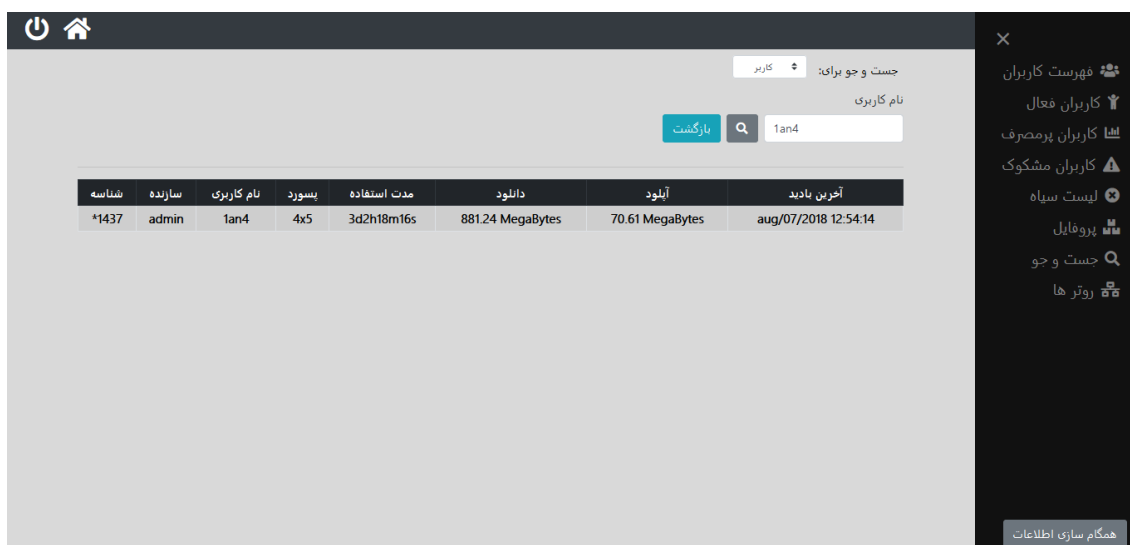
با انتخاب جست و جو بر اساس نشست ها کاربر قادر خواهد بود تا با وارد کردن نام اکانت یا آی پی مورد نظر و یا مک آدرس مورد نظر به جست و جو در نشست ها بپردازد، که با کلیک بر روی جست و جو جدول مربوط به نشست ها نمایش داده می شود که شامل اطلاعات کلی نشست می باشد. تمامی این اطلاعات و فیلدها در شکل ۲-۱۳ قابل مشاهده است.

شناسه	نام کاربری	سارنده	مک آدرس	آی پی	ار زمان	تا زمان	دانلود	آپلود	مدت استفاده
*1FC3	admin	1an4	C8:A8:23:72:36:CC	172.16.1.225	aug/04/2018 10:31:20	aug/05/2018 05:54:33	756.18 MegaBytes	59.01 MegaBytes	19h21m27s
*1FD6	admin	1an4	C8:A8:23:72:36:CC	172.16.1.225	aug/05/2018 05:57:05	aug/05/2018 06:06:27	4.37 MegaBytes	739.71 KiloBytes	9m22s
*1FD8	admin	1an4	C8:A8:23:72:36:CC	172.16.1.225	aug/05/2018 06:06:29	aug/05/2018 06:06:32	4.46 KiloBytes	1.71 KiloBytes	3s
*1FD9	admin	1an4	C8:A8:23:72:36:CC	172.16.1.225	aug/05/2018 06:06:33	aug/05/2018 06:06:37	92 Bytes	40 Bytes	4s
*1FDA	admin	1an4	C8:A8:23:72:36:CC	172.16.1.225	aug/05/2018 06:06:47	aug/05/2018 06:06:58	129 Bytes	181 Bytes	11s
*1FDB	admin	1an4	C8:A8:23:72:36:CC	172.16.1.225	aug/05/2018 06:07:05	aug/07/2018 12:54:14	120.68 MegaBytes	10.87 MegaBytes	2d6h47m9s

شکل ۱۳-۴: صفحه جست و جو بر اساس نشست ها

۱۰-۴-۲. جست و جو بر اساس کاربر

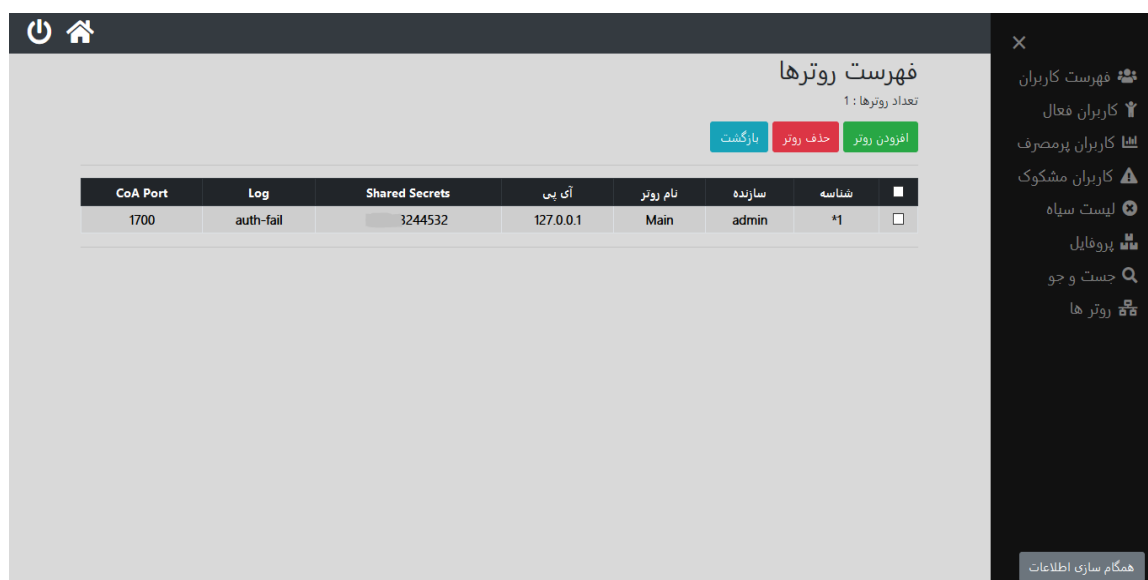
با انتخاب جست و جو بر اساس کاربر، کاربر ادمین قادر خواهد بود با وارد کردن نام اکانت به جست و جو در لیست کاربران بپردازد که شامل اطلاعاتی مانند پسوندد کاربر، میزان دانلود، میزان آپلود و یا آخرین استفاده از کانت می باشد. در شکل ۴-۱۴ این نوع جست و جو نمایش داده شده است.



شکل ۴-۱۴: صفحه جست و جو بر اساس کاربر

۴-۱۱. صفحه روترها

با توجه به شکل ۴-۱۵ در این صفحه کاربر می تواند لیست روتر های متصل شده به یوزر منیجر را مشاهده کند و یا روتر مورد نظر خود را از این لیست حذف کرده و یا یک روتر جدید به این لیست اضافه کند.



شکل ۴-۱۵: صفحه روترها

با کلیک بر روی گزینه افزودن روتر فرم اطلاعات روتر جدید جهت افزودن به کاربر نشان داده می شود. این فرم شامل نام روتر، نام سازنده دسترسی به روتر، عبارت امنیتی بین روتر و یوزر منیجر، آی

پی روتر، نوع لاگ، استفاده از پورت CoA و شماره پورت می باشد. این فرم در شکل ۴-۱۶ قابل مشاهده است.

The screenshot shows a web form titled "افزودن روتر" (Add Router). The form contains the following elements:

- نام روتر** (Router Name): A text input field with the placeholder "Enter Name Of Router".
- سازنده** (Manufacturer): A dropdown menu currently showing "admin".
- آی پی** (IP): A text input field with the placeholder "Enter Router IP".
- Shared Secret**: A text input field with the placeholder "Enter Shared Secret".
- Log**: A dropdown menu currently showing "acct-fail".
- Use CoA**: A dropdown menu currently showing "Disable".
- CoA Port**: A text input field with the placeholder "Enter CoA Port".
- ثبت** (Save): A blue button.
- خروج** (Exit): A red button.

شکل ۴-۱۶: فرم افزودن روتر

۴-۱۲. صفحه پنل پذیرش

همان گونه که در شکل ۴-۱۷ مشاهده می شود پس از لاگین کردن کاربر با دسترسی کاربر پذیرش صفحه گزارش گیری از اکانت مهمان به کاربر نمایش داده می شود، این صفحه حاوی ورودی نام کاربری اکانت جهت جست و جو می باشد که پس از جست و جو دو جدول در قالب های اطلاعات کلی کاربر و جریات مصرف اکانت به کاربر پذیرش نمایش داده می شود.

نام کاربری مورد نظر						
1an4						
اطلاعات کلی کاربر						
نام کاربری	رمز عبور	آخرین بازدید	مدت استفاده	دانلود	آپلود	کل ترافیک مصرفی
1an4	4x5	aug/07/2018 12:54:14	3d2h18m16s	881.24 MegaBytes	70.61 MegaBytes	951.84 MegaBytes
جزئیات مصرف اکانت						
مک آدرس	آی بی	مدت استفاده	آپلود	دانلود	از زمان	تا زمان
C8:A8:23:72:36:CC	172.16.1.225	19h21m27s	59.01 MegaBytes	756.18 MegaBytes	aug/04/2018 10:31:20	aug/05/2018 05:54:33
C8:A8:23:72:36:CC	172.16.1.225	9m22s	739.71 KiloBytes	4.37 MegaBytes	aug/05/2018 05:57:05	aug/05/2018 06:06:27
C8:A8:23:72:36:CC	172.16.1.225	3s	1.71 KiloBytes	4.46 KiloBytes	aug/05/2018 06:06:29	aug/05/2018 06:06:32
C8:A8:23:72:36:CC	172.16.1.225	4s	40 Bytes	92 Bytes	aug/05/2018 06:06:33	aug/05/2018 06:06:37
C8:A8:23:72:36:CC	172.16.1.225	11s	181 Bytes	129 Bytes	aug/05/2018 06:06:47	aug/05/2018 06:06:58
C8:A8:23:72:36:CC	172.16.1.225	2d6h47m9s	10.87 MegaBytes	120.68 MegaBytes	aug/05/2018 06:07:05	aug/07/2018 12:54:14

شکل ۱۷-۴: صفحه پنل پذیرش

۱۳-۴. صفحه مدیریت پایگاه داده و عملیات کاربر

این صفحه مربوط به خود فریم ورک جنگو می باشد، بصورت خلاصه در این صفحه کاربر ادمین قادر خواهد بود تا تمام اطلاعات پایگاه داده را مشاهده کند، اطلاعات کاربران را مشاهده کند، کاربر جدید ایجاد کند و یا رمز عبور خود و یا کاربران دیگر را تغییر دهد.

۱-۱۳-۴. صفحه ورود

با توجه به شکل ۴-۱۸ این صفحه شامل فیلد های نام کاربری و رمز عبور کاربر می باشد. در این صفحه کاربر ادمین با وارد کردن نام کاربری و رمز عبور خود به قسمت مدیریت پایگاه داده ها و کاربران دسترسی خواهد داشت.

Django administration

Username:
admin

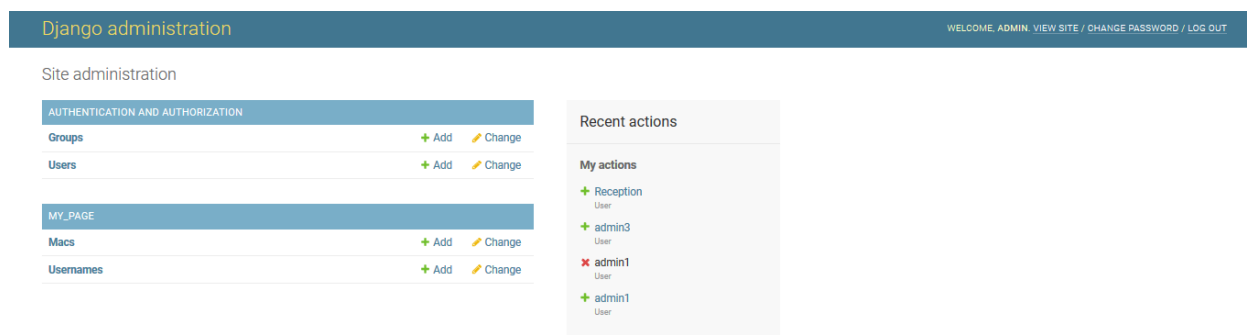
Password:

Log in

شکل ۱۸-۴: صفحه ورود مدیریت پایگاه داده و کاربران

۲-۱۳-۴. صفحه ادمین مدیریت پایگاه داده

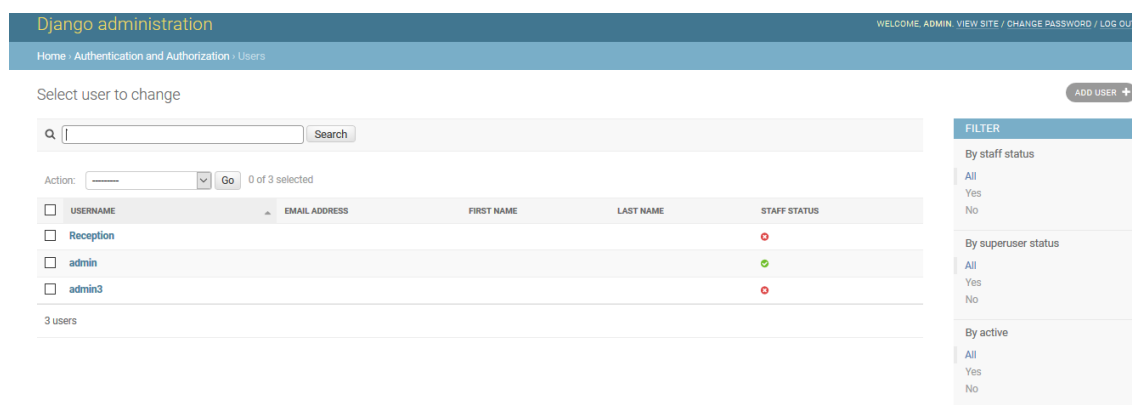
پس از لاگین کردن کاربر با دسترسی ادمین صفحه مدیریت پایگاه داده و کاربران که در شکل ۴-۱۹ ملاحظه می شود نمایش داده خواهد شد، این صفحه بصورت خلاصه شامل نام جدول های پایگاه داده، قسمت کاربران و گروه ها، تغییر پسورد ادمین و ... می باشد.



شکل ۴-۱۹: صفحه ادمین مدیریت پایگاه داده

۳-۱۳-۴. صفحه کاربران

با توجه به شکل ۴-۲۰ در این صفحه کاربر ادمین لیست تمامی کاربران موجود در برنامه را با تمام جزئیات آنها مشاهده می کند و همچنین با کلیک بر روی گزینه ADD USER می تواند کاربر جدید ایجاد کند.



شکل ۴-۲۰: صفحه کاربران

۴-۱۳-۴. صفحه ساخت کاربر

در این صفحه کاربر ادمین خواهد توانست با وارد کردن نام کاربری جدید و رمز عبور، کاربر جدید برای دسترسی به پنل پذیرش ایجاد کند. تمامی این امکانات در شکل ۴-۲۱ قابل مشاهده است.

Django administration

WELCOME, ADMIN / VIEW SITE / CHANGE PASSWORD / LOG OUT

Home / Authentication and Authorization / Users / Add user

Add user

First, enter a username and password. Then, you'll be able to edit more user options.

Username:

Required. 150 characters or fewer. Letters, digits and @/+/./_ only.

Password:

Your password can't be too similar to your other personal information.
Your password must contain at least 8 characters.
Your password can't be a commonly used password.
Your password can't be entirely numeric.

Password confirmation:

Enter the same password as before, for verification.

Save and add another

Save and continue editing

SAVE

شکل ۲۱-۴: صفحه ساخت کاربران

جمع بندی و کارهای آینده

۱-۵. خلاصه

بصورت کلی در این برنامه کاربر قادر خواهد بود تا با امکاناتی که در اختیار دارد به مدیریت کاربران اینترنت بپردازد. این امکانات شامل تمام اطلاعات کاربران، تمامی اطلاعات مربوط به بسته های اینترنت، اطلاعات مربوط به محدودیت های اعمال شده بر روی بسته های اینترنت، اطلاعات و ترافیک رد و بدل شده بر روی پورت های اترنت روتر میکروتیک، اطلاعات مربوط به روتر های متصل شده به برنامه می باشد. در این پروژه ابزار هایی نیز تعبیه شده است که به کاربر اجازه می دهد تا از مصرف اینترنت توسط افراد سود جو جلوگیری به عمل آید، این ابزار ها شامل جست جو کاربران پرمصرف، جست جو کاربران مشکوک و همچنین لیست سیاه می باشد. نحوه استفاده از دو ابزار اول به این صورت می باشد که کاربر با وارد کردن بازه زمانی مورد نظر خود می تواند لیست کاربران پرمصرف و مشکوک را مشاهده کند. با استفاده از ابزار سوم کاربر قادر خواهد بود تا به مسدود سازی کاربر مورد نظر پرداخته و لیست کاربران مسدود شده را مشاهده کند.

در این پروژه امکان جست و جو پیشرفته کاربران نیز طراحی شده است. این جست و جو به این گونه طراحی شده است که کاربر قادر خواهد با انتخاب جست و جو بر اساس اطلاعات کلی کاربر و یا بر اساس نشست های کاربر اطلاعات مورد نیاز خود را مشاهده کند.

پنل قسمت کاربر پذیرش این برنامه به این گونه است که کاربر با وارد کردن نام کاربری اکانت مورد نظر خود و انجام جست و جو قادر خواهد بود تا اطلاعات کاربر را در دو دسته اطلاعات کلی کاربر و ریز مصرف یا نشست های کاربر مشاهده کند. با توجه به استفاده افرادی بدون دانش قبلی در مورد فناوری اطلاعات و یا دانش شبکه، سعی بر آن شده تا این پنل بصورت ساده و بدون گزینه های پیچیده و در عین حال کاربردی طراحی و پیاده سازی شود.

۲-۵. جمع بندی

در این پایان نامه ابتدا به شرح کلی برنامه و دلایل نیاز به ساخت برنامه مدیریت کاربران اینترنت پرداخته شد، اهداف و راه کار های مورد نظر بررسی شده و چند نمونه از برنامه های مشابه جهت اکانتیگ مورد بررسی قرار گرفت. همچنین در این فصل ابزار های مورد استفاده در این پروژه نام برده شده و بصورت مجزا و خلاصه توضیح داده شده است. در فصل دوم پروژه بصورت کامل از دید مهندسی نرم افزار مورد بررسی قرار گرفت و نمودار های مختلف پروژه از قبیل نموداری های مورد کاربرد، نمودار کلاس و نمودار توالی ترسیم و بیان شد.

در فصل سوم پایگاه داده مورد استفاده در این پروژه بیان و توضیح داده شد. همچنین تمامی جداول مورد استفاده در این پروژه بصورت مجزا ذکر شده و به توضیح آن پرداخته شده است. فصل سوم این پایان نامه در مورد نحوه استفاده از برنامه می باشد. در این فصل تمامی قسمت های برنامه به همراه شکل هر قسمت توضیح داده شده است.

۳-۵. کار های آینده

در طرح های آینده می توان امکانات گوناگون و پر کاربرد دیگری را به این پروژه اضافه نمود که با توجه به محدودیت زمانی و وقت گیر بودن پیاده سازی این امکانات، امکان اضافه نمودن این طرح ها وجود ندارد. امکانات و طرح های مورد نظر شامل موارد زیر می باشند:

تاریخچه سایت های بازدید شده: با استفاده از این ویژگی تمامی وب سایت های بازدید شده توسط کاربر مهمان قابل مشاهده خواهد بود و همچنین می توان این تاریخچه را بصورت محدود در پایگاه داده برنامه ذخیره نمود.

متصل شدن برنامه به درگاه پرداخت بانکی: در این ویژگی کاربر مهمان پس از اتمام شارژ اینترنت خود قادر خواهد بود تا شارژ خود را تمدید کرده و یا میزان شارژ دلخواه خود را با متصل شدن به درگاه پرداخت بانکی خریداری نماید.

سفارشی سازی صفحه هات اسپات روتر میکروتیک: با استفاده از این ویژگی کاربر ادمین شبکه قادر خواهد بود تا ظاهر صفحه هات اسپات روتر را بر اساس سلیقه خود طراحی کند همچنین این بصورت کامل فارسی خواهد بود.

قابلیت یکپارچگی از طریق API با دیگر نرم افزار های سازمان: با طراحی API کارآمد این نرم افزار قادر خواهد بود تا بصورت مداوم با دیگر نرم افزار های سازمان کار کرده و از اطلاعات یکدیگر استفاده کنند. به عنوان نمونه با متصل شدن به نرم افزار های هتل داری تاریخچه کامل استفاده از اکانت اینترنت توسط مهمان در نرم افزار هتل داری ثبت شده و هزینه استفاده اینترنت به هزینه اقامت مسافر اضافه خواهد شد.

پیوست (کدهای اصلی برنامه)

در این بخش سعی شده است با توجه به تحت وب بودن برنامه تنها کد ها و تابع های مهم که با زبان پایتون نوشته شده است آورده شود.

کد های صفحه ورود و قسمت خروج

```
def login_view(request):
    if request.method == "POST":
        form = AuthenticationForm(data=request.POST)
        if form.is_valid():
            user = form.get_user()
            login(request, user)
            return redirect('my_page:index')
    else:
        form = AuthenticationForm()
    return render(request, 'my_page/login.html', {'form': form})
```

```
def logout_view(request):
    if request.method == 'POST':
        logout(request)
        return redirect('my_page:login')
```

```
def login_view_guest(request):
    if request.method == "POST":
        form = AuthenticationForm(data=request.POST)
        if form.is_valid():
            user = form.get_user()
            login(request, user)
            return redirect('my_page:guest')
    else:
        form = AuthenticationForm()
    return render(request, 'my_page/login_guest.html', {'form': form})
```

کد های صفحه اصلی پنل با دسترسی ادمین

```
@user_passes_test(lambda u: u.is_superuser, login_url="/Home/login")
@login_required(login_url="/Home/login")
def index(request):
    if 'sync' in request.POST:
        api = connection_router()
        if api == None:
            messages.error(request, 'روتر با ارتباط خطای')
        else:
```

```

get_user = api.get_resource('/tool/user-manager/user')
list_user = get_user.get()
for i in list_user:
    n = models.Username(id=i['id'], customer=i['customer'], username=i['username'],
password=i['password'],
                        last_seen=i['last-seen'], shared_users=i['shared-users'])
    n.save()
url = "http://۱۹۲,۱۶۸,۲۰,۱/graphs/iface/"
intf_d = ""
interface_name = []
api = connection_router()
if api == None:
    messages.error(request, 'روتر با ارتباط خطای')
else:
    get_interface = api.get_resource('/tool/graphing/interface')
    list_graph_interface = get_interface.get()
    for i in range(len(list_graph_interface)):
        ob = list_graph_interface[i]
        if ob['interface'] == 'all':
            interface_name = []
            get_interface = api.get_resource('/interface')
            list_interface = get_interface.get()
            for j in range(len(list_interface)):
                obj = list_interface[j]
                interface_name.append(obj['name'])
                intf_d = obj['name']
            break
        else:
            interface_name.append(ob['interface'])
            intf_d = ob['interface']
    url = url + intf_d
context = {'interface': interface_name, 'url': url}
return render(request, 'my_page/base.html', context=context)

```

کد های صفحه فهرست کاربران

```

@user_passes_test(lambda u: u.is_superuser, login_url="/Home/login")
@login_required(login_url="/Home/login")
def users(request):
    # 'if' for check box for delete item
    if 'inputs' in request.POST:
        # gives list of id of inputs
        list_of_id_selected = request.POST.getlist('inputs')
        models.Username.objects.filter(id__in=list_of_id_selected).delete()
        for i in range(len(list_of_id_selected)):
            api = connection_router()

```

```

        get_user = api.get_resource('/tool/user-manager/user')
        get_user.remove(id=list_of_id_selected[i])
# 'if' for search users in list of user and create user
if 'username_s' in request.POST:
    user_s = request.POST['username_s']
    users = models.Username.objects.filter(username__contains=user_s)
    count = models.Username.objects.count()
    api = connection_router()
    get_customer = api.get_resource('/tool/user-manager/customer')
    list_customer = get_customer.get()
    get_profile = api.get_resource('/tool/user-manager/profile')
    list_profile = get_profile.get()
    username = ""
# 'else' for display list of user and create user
else:
    api = connection_router()
    get_customer = api.get_resource('/tool/user-manager/customer')
    list_customer = get_customer.get()
    get_profile = api.get_resource('/tool/user-manager/profile')
    list_profile = get_profile.get()
    get_user = api.get_resource('/tool/user-manager/user')
    username = ""
if 'prefix' in request.POST:
    number_of_user = request.POST['number_of_user']
    prefix = request.POST['prefix']
    u_length = request.POST['u_length']
    p_length = request.POST['p_length']
    owner = request.POST['owner']
    profile_ = request.POST['profile']
    number_of_user = int(number_of_user)
    u_length = int(u_length)
    p_length = int(p_length)
    length_of_prefix = len(prefix)
    # content-type of response
    response = HttpResponse(content_type='application/ms-excel')
    # decide file name
    response['Content-Disposition'] = 'attachment; filename="ListOfUser.xls"'
    # creating workbook
    wb = xlwt.Workbook(encoding='utf-8')
    # adding sheet
    ws = wb.add_sheet("sheet")
    # Sheet header, first row
    row_num = 0
    font_style = xlwt.XFStyle()
    # headers are bold
    font_style.font.bold = True

```

```

# column header names, you can use your own headers here
columns = ['Username', 'Password']
# write column headers in sheet
for col_num in range(len(columns)):
    ws.write(row_num, col_num, columns[col_num], font_style)
# Sheet body, remaining rows
font_style = xlwt.XFStyle()
if u_length > length_of_prefix:
    id_length = u_length - length_of_prefix
    for i in range(number_of_user):
        random_id = uuid.uuid4()
        random_id_user = str(random_id)[:id_length]
        username = prefix + random_id_user
        password = str(random_id)[:p_length]
        try:
            get_user.add(customer=owner, disabled='no', password=password,
username=username)
        except Exception as error:
            error_text = str(error)
            if error_text[0:3] == 'failure: such username alrea':
                random_id = uuid.uuid4()
                random_id_user = str(random_id)[:id_length]
                username = prefix + random_id_user
                get_user.add(customer=owner, disabled='no', password=password,
username=username)
            get_user.create_and_activate_profile(numbers=username, customer=owner,
profile=profile_)
            row_num = row_num + 1
            ws.write(row_num, 0, username, font_style)
            ws.write(row_num, 1, password, font_style)
            wb.save(response)
            return response
    else:
        messages.error(request, 'ورودی اطلاعات در خطا'
(باشد می))
count = models.Username.objects.count()
objs = models.Username.objects.all().order_by('-created_date')
paginator = Paginator(objs, 9)
page = request.GET.get('page', 1)
try:
    users = paginator.page(page)
except PageNotAnInteger:
    users = paginator.page(1)
except EmptyPage:
    users = paginator.page(paginator.num_pages)

```



```

context = {'users': users,
           'count': count,
           'customer': list_customer,
           'profile': list_profile,
           'n': username}
return render(request, 'my_page/users.html', context=context)

```

کد های صفحه کاربران فعال

```

@user_passes_test(lambda u: u.is_superuser, login_url="/Home/login")
@login_required(login_url="/Home/login")
def active_users(request):
    user_list = ""
    length = 0
    if 'inputs' in request.POST:
        # gives list of id of inputs
        list_of_id_selected = request.POST.getlist('inputs')
        for i in range(len(list_of_id_selected)):
            api = connection_router()
            if api == None:
                messages.error(request, 'روتر با ارتباط خطای')
            else:
                get_router = api.get_resource('/ip/hotspot/active')
                get_router.remove(id=list_of_id_selected[i])
    api = connection_router()
    if api == None:
        messages.error(request, 'روتر با ارتباط خطای')
    else:
        get_active_user_list = api.get_resource('/ip/hotspot/active')
        user_list = get_active_user_list.get()
        length = len(user_list)
        for i in range(length):
            object = user_list[i]
            object['bytes-in'] = format_bytes(int(object['bytes-in']))
            object['bytes-out'] = format_bytes(int(object['bytes-out']))
            object['mac_address'] = object.pop('mac-address')
            object['bytes_in'] = object.pop('bytes-in')
            object['bytes_out'] = object.pop('bytes-out')
            cut_mac = object['mac_address'][0:8]
            company =
list(models.Mac.objects.filter(mac_address__contains=cut_mac).values_list('company',
flat=True))
            object.update({'company': ' '.join(map(str, company))})

```

```
context = {'users': user_list, 'count': length}
return render(request, 'my_page/active_users.html', context=context)
```

کد های صفحه پروفایل

```
@user_passes_test(lambda u: u.is_superuser, login_url="/Home/login")
@login_required(login_url="/Home/login")
def profile(request):
    global list_limitation, list_customer
    api = connection_router()
    get_profile = api.get_resource('/tool/user-manager/profile')
    count_profile = len(get_profile.get())
    get_p_l = api.get_resource('/tool/user-manager/profile/profile-limitation')
    if 'inputs' in request.POST:
        list_of_id_selected = request.POST.getlist('inputs')
        for i in range(len(list_of_id_selected)):
            api = connection_router()
            get_profile.remove(id=list_of_id_selected[i])
    if 'name' in request.POST:
        name = request.POST['name']
        owner = request.POST['owner']
        validity = request.POST['validity']
        shared = request.POST['shared']
        start_at = request.POST['start_at']
        name_l = request.POST['name_l']
        try:
            get_profile.add(name=name, owner=owner, validity=validity,
override_shared_users=shared, starts_at=start_at)
            get_p_l.add(profile=name, limitation=name_l)
        except Exception as error:
            error_text = str(error)
            if error_text[۹:۴۹] == 'invalid time value for argument validity':
                messages.error(request, "باشد نمی ورودی اعتبار")
            elif error_text[۹:۵۲] == "failure: profile with such name already exi":
                messages.error(request, "باشد می تکراری پروفایل")
    list_profile = get_profile.get()
    n = ۰
    for i in list_profile:
        i['number'] = n
        n = n + ۱
    get_customer = api.get_resource('/tool/user-manager/customer')
    list_customer = get_customer.get()
    get_limitation = api.get_resource('/tool/user-manager/profile/limitation')
    list_limitation = get_limitation.get()
    context = {'length': list_profile,
```

```

        'customer': list_customer,
        'limit': list_limitation,
        'count': count_profile}
return render(request, 'my_page/profile_fp.html', context=context)

```

کد های صفحه لیست محدودیت ها

```

@user_passes_test(lambda u: u.is_superuser, login_url="/Home/login")
@login_required(login_url="/Home/login")
def limitation(request):
    api = connection_router()
    get_limitation = api.get_resource('/tool/user-manager/profile/limitation')
    get_customer = api.get_resource('/tool/user-manager/customer')
    list_customer = get_customer.get()
    count = len(get_limitation.get())
    if 'inputs' in request.POST:
        list_of_id_selected = request.POST.getlist('inputs')
        for i in range(len(list_of_id_selected)):
            get_limitation.remove(id=list_of_id_selected[i])
    if 'name' in request.POST:
        name = request.POST['name']
        owner = request.POST['owner']
        transfer = request.POST['transfer_limit']
        try:
            get_limitation.add(name=name, owner=owner, transfer_limit=transfer)
        except Exception as error:
            error_text = str(error)
            if error_text[9:12] == 'value of transfer-limit contains invalid tr':
                messages.error(request, 'باید نام صحیح بسته حجم')
            elif error_text[9:12] == 'failure: such name already exists' executin':
                messages.error(request, 'باید نام تکراری محدودیت نام')
    list_limitation = get_limitation.get()
    n = 0
    for i in list_limitation:
        i['transfer-limit'] = format_bytes(int(i['transfer-limit']))
        i['transfer_limit'] = i.pop('transfer-limit')
        i['number'] = n
        n = n + 1
    context = {'length': list_limitation, 'customer': list_customer, 'count': count}
    return render(request, 'my_page/limitation_fp.html', context=context)

```

```
@user_passes_test(lambda u: u.is_superuser, login_url="/Home/login")
@login_required(login_url="/Home/login")
def search(request):
    api = connection_router()
    if 'Username' in request.POST:
        username = request.POST['Username']
        get_user = api.get_resource('/tool/user-manager/user')
        user_list = get_user.get(username=username)
        user = user_list[0]
        user['download-used'] = format_bytes(int(user['download-used']))
        user['upload-used'] = format_bytes(int(user['upload-used']))
        user['download_used'] = user.pop('download-used')
        user['upload_used'] = user.pop('upload-used')
        user['last_seen'] = user.pop('last-seen')
        user['uptime_used'] = user.pop('uptime-used')
        user['shared_users'] = user.pop('shared-users')
        context = {'user_list': user_list}
    elif 'username_s' in request.POST:
        username = request.POST['username_s']
        mac = request.POST['mac_s']
        ip = request.POST['ip_s']
        if username != "" or mac != "" or ip != "":
            get_user = api.get_resource('/tool/user-manager/session')
            info_session = get_user.get(user=username, calling_station_id=mac, user_ip=ip)
            length = len(info_session)
            if length == 0:
                messages.error(request, 'ندارد وجود نشستی')
            for i in range(length):
                object = info_session[i]
                object['download'] = format_bytes(int(object['download']))
                object['upload'] = format_bytes(int(object['upload']))
                object['calling_station_id'] = object.pop('calling-station-id')
                object['user_ip'] = object.pop('user-ip')
                object['from_time'] = object.pop('from-time')
                object['till_time'] = object.pop('till-time')
                context = {'info_session': info_session}
            else:
                messages.error(request, 'شود پر باید فیلد یک حداقل')
                context = {'info_session': ""}
    elif 'ip' in request.POST:
        ip = request.POST['ip']
        get_user = api.get_resource('/tool/user-manager/session')
        info_session = get_user.get(user_ip=ip)
```

```

length = len(info_session)
for i in range(length):
    object = info_session[i]
    object['download'] = format_bytes(int(object['download']))
    object['upload'] = format_bytes(int(object['upload']))
    object['calling_station_id'] = object.pop('calling-station-id')
    object['user_ip'] = object.pop('user-ip')
    object['from_time'] = object.pop('from-time')
    object['till_time'] = object.pop('till-time')
    context = {'info_session': info_session}
elif 'mac_address' in request.POST:
    mac_address = request.POST['mac_address']
    get_user = api.get_resource('/tool/user-manager/session')
    info_session = get_user.get(calling_station_id=mac_address)
    length = len(info_session)
    for i in range(length):
        object = info_session[i]
        object['download'] = format_bytes(int(object['download']))
        object['upload'] = format_bytes(int(object['upload']))
        object['calling_station_id'] = object.pop('calling-station-id')
        object['user_ip'] = object.pop('user-ip')
        object['from_time'] = object.pop('from-time')
        object['till_time'] = object.pop('till-time')
        context = {'info_session': info_session}
else:
    context = {'b': "error"}
return render(request, 'my_page/search.html', context=context)

```

کد های صفحه روتر

```

@user_passes_test(lambda u: u.is_superuser, login_url="/Home/login")
@login_required(login_url="/Home/login")
def router(request):
    if 'inputs' in request.POST:
        # gives list of id of inputs
        list_of_id_selected = request.POST.getlist('inputs')
        for i in range(len(list_of_id_selected)):
            api = connection_router()
            get_router = api.get_resource('/tool/user-manager/router')
            get_router.remove(id=list_of_id_selected[i])
    api = connection_router()
    get_customer = api.get_resource('/tool/user-manager/customer')
    list_customer = get_customer.get()
    get_router = api.get_resource('/tool/user-manager/router')
    list_router = get_router.get()

```

```

length = len(list_router)
for i in range(length):
    object = list_router[i]
    object['ip_address'] = object.pop('ip-address')
    object['shared_secret'] = object.pop('shared-secret')
    object['coa_port'] = object.pop('coa-port')
if 'name' in request.POST:
    name = request.POST['name']
    owner = request.POST['owner']
    ip_address = request.POST['ip_address']
    shared_secret = request.POST['shared_secret']
    log = request.POST['log']
    use_coa = request.POST['use_coa']
    coa = request.POST['coa']
    get_router.add(name=name, customer=owner, ip_address=ip_address,
shared_secret=shared_secret, log=log,
                    use_coa=use_coa, coa_port=coa)
context = {'router': list_router,
            'count': length,
            'customer': list_customer}
return render(request, 'my_page/router.html', context=context)

```

کد های صفحه کاربران پر مصرف

```

@user_passes_test(lambda u: u.is_superuser, login_url="/Home/login")
@login_required(login_url="/Home/login")
def highly_user(request):
    list = []
    final_dict = {}
    count = 0
    list_dict = []
    if 'start_day' in request.POST:
        number_of_user = request.POST['number_of_user']
        start_day = request.POST['start_day']
        if start_day == "":
            start_day = '۲۰۰۰-۱۲-۲۴'
        start_day = start_day + ' ۰۰:۰۰:۰۰'
        start_day = datetime.datetime.strptime(start_day, '%Y-%m-%d %H:%M:%S')
        end_day = request.POST['end_day']
        if end_day == "":
            end_day = '۲۹۹۹-۱۲-۲۴'
        end_day = end_day + ' ۲۳:۵۹:۵۹'
        end_day = datetime.datetime.strptime(end_day, '%Y-%m-%d %H:%M:%S')
        api = connection_router()
        get_session = api.get_resource('/tool/user-manager/session')

```

```

list_session = get_session.get()
for j in range(len(list_session)):
    obj = list_session[j]
    date = obj['from-time']
    select_day = datetime.datetime.strptime(date, '%b/%d/%Y %H:%M:%S')
    if start_day <= select_day <= end_day:
        list\ .append(list_session[j])
items = defaultdict(list)
for row in list\ :
    items[row['calling-station-id']].append(row['download'])
    items[row['calling-station-id']].append(row['upload'])
t = dict(items)
bit = 0
for i in t:
    traffic = t[i]
    for j in traffic:
        bit = int(j) + bit
    t[i] = bit
    bit = 0
list_dict = []
sorted_list = sorted(t.items(), key=itemgetter(1), reverse=True)
count = len(sorted_list)
if number_of_user == "":
    number_of_user = len(sorted_list)
for k in range(int(number_of_user)):
    ob = sorted_list[k]
    val = list(ob)
    val[1] = format_bytes(val[1])
    # final_dict.update({ val[0]: val[1] })
    cut_mac = val[0][0:8]
    company =
list(models.Mac.objects.filter(mac_address__contains=cut_mac).values_list('company',
flat=True))
    final_dict.update({"mac": val[0], "traffic": val[1], "company": ' '.join(map(str,
company))})
    p = final_dict.copy()
    list_dict.append(p)
context = {'list\': list_dict, 'count': count}
return render(request, 'my_page/highly_user.html', context=context)

```

کد های صفحه کاربران مشکوک

```
@user_passes_test(lambda u: u.is_superuser, login_url="/Home/login")
@login_required(login_url="/Home/login")
def risky_user(request):
    start_day = '۲۰۱۹-۱۲-۲۴ ۰۰:۰۰:۰۰'
    list\ = []
    if 'start_day' in request.POST:
        start_day = request.POST['start_day']
        if start_day == "":
            start_day = '۲۰۰۰-۱۲-۲۴'
        start_day = start_day + ' ۰۰:۰۰:۰۰'
        start_day = datetime.datetime.strptime(start_day, '%Y-%m-%d %H:%M:%S')
        end_day = request.POST['end_day']
        if end_day == "":
            end_day = '۲۹۹۹-۱۲-۲۴'
        end_day = end_day + ' ۲۳:۵۹:۵۹'
        end_day = datetime.datetime.strptime(end_day, '%Y-%m-%d %H:%M:%S')
        api = connection_router()
        get_session = api.get_resource('/tool/user-manager/session')
        list_session = get_session.get()
        for j in range(len(list_session)):
            obj = list_session[j]
            date = obj['from-time']
            select_day = datetime.datetime.strptime(date, '%b/%d/%Y %H:%M:%S')
            if start_day <= select_day <= end_day:
                list\ .append(list_session[j])
    li = { }
    list_dict = []
    count = ۰
    items = defaultdict(list)
    for row in list\:
        items[row['calling-station-id']].append(row['user'])
    t = dict(items)
    for i in t:
        t[i] = list(dict.fromkeys(t[i]))
        if len(t[i]) > ۳:
            count = count + ۱
            cut_mac = i[۰:۸]
            company =
list(models.Mac.objects.filter(mac_address__contains=cut_mac).values_list('company',
flat=True))
li.update({'mac': i, 'user': t[i], 'company': ' '.join(map(str, company))})
p = li.copy()
list_dict.append(p)
```



```
context = {'list': li, 'count': count, 'ops': list_dict}
return render(request, 'my_page/risky_user.html', context=context)
```

کد صفحه لیست سیاه

```
@user_passes_test(lambda u: u.is_superuser, login_url="/Home/login")
@login_required(login_url="/Home/login")
def bad_user(request):
    main_list = []
    api = connection_router()
    get_user_info = api.get_resource('/ip/hotspot/ip-binding')
    if 'inputs' in request.POST:
        # gives list of id of inputs
        list_of_id_selected = request.POST.getlist('inputs')
        for i in range(len(list_of_id_selected)):
            api = connection_router()
            if api == None:
                messages.error(request, 'روتر با ارتباط خطای')
            else:
                get_router = api.get_resource('/ip/hotspot/ip-binding')
                get_router.remove(id=list_of_id_selected[i])
    if 'mac_address' in request.POST:
        mac = request.POST['mac_address']
        try:
            get_user_info.add(mac_address=mac, type='blocked')
        except Exception as error:
            error = str(error)
            text = error[9:37]
            if text == 'failure: such client already':
                messages.error(request, 'است شده بلاک قبلا نظر مورد کاربر آدرس مک')
            else:
                messages.error(request, 'باشند می اشتباه آدرس مک فرمت')
    list_bad_user = get_user_info.get()
    for i in range(len(list_bad_user)):
        obj = list_bad_user[i]
        if 'mac-address' in obj:
            obj['mac_address'] = obj.pop('mac-address')
            cut_mac = obj['mac_address'][:8]
            company =
list(models.Mac.objects.filter(mac_address__contains=cut_mac).values_list('company',
flat=True))
        obj.update({'company': ' '.join(map(str, company))})
        main_list.append(list_bad_user[i])
    context = {'list': main_list}
    return render(request, 'my_page/block_user.html', context=context)
```

کد های صفحه نشست ها

```
@user_passes_test(lambda u: u.is_superuser, login_url="/Home/login")
@login_required(login_url="/Home/login")
def user_info(request, value):
    api = connection_router()

    get_user = api.get_resource('/tool/user-manager/session')
    info_session = get_user.get(user=value)
    length = len(info_session)
    for i in range(length):
        object = info_session[i]
        object['download'] = format_bytes(int(object['download']))
        object['upload'] = format_bytes(int(object['upload']))
        object['calling_station_id'] = object.pop('calling-station-id')
        object['user_ip'] = object.pop('user-ip')
        object['from_time'] = object.pop('from-time')
        object['till_time'] = object.pop('till-time')
    context = {'info_session': info_session}
    return render(request, 'my_page/user_info.html', context=context)
```

کد های صفحه کاربر پذیرش

```
@login_required(login_url="/Home/login_guest")
def normal_user(request):
    user_list = ""
    info_session = ""
    if 'username' in request.POST:
        username = request.POST['username']
        api = connection_router()
        get_user = api.get_resource('/tool/user-manager/user')
        user_list = get_user.get(username=username)
        if not user_list:
            messages.error(request, 'ندارد وجود نظر مورد اکانت')
        else:
            user = user_list[0]
            if user['last-seen'] != 'never':
                traffic = int(user['download-used']) + int(user['upload-used'])
                traffic = format_bytes(traffic)
                user['download-used'] = format_bytes(int(user['download-used']))
                user['upload-used'] = format_bytes(int(user['upload-used']))
                user['download_used'] = user.pop('download-used')
                user['upload_used'] = user.pop('upload-used')
                user['last_seen'] = user.pop('last-seen')
                user['uptime_used'] = user.pop('uptime-used')
                user.update({'traffic': traffic})
```

```

get_user = api.get_resource('/tool/user-manager/session')
info_session = get_user.get(user=username)
length = len(info_session)
for i in range(length):
    object = info_session[i]
    object['download'] = format_bytes(int(object['download']))
    object['upload'] = format_bytes(int(object['upload']))
    object['calling_station_id'] = object.pop('calling-station-id')
    object['user_ip'] = object.pop('user-ip')
    object['from_time'] = object.pop('from-time')
    object['till_time'] = object.pop('till-time')
else:
    user_list = "
    messages.error(request, 'است نشده استفاده نظر مورد اکاننت')
context = {'user_list': user_list, 'session': info_session}
return render(request, 'my_page/normal_user.html', context=context)

```

فهرست منابع

- کجباف، محسن. مهندسی نرم افزار. پندار پارس [۱]
- روحانی رانکوهی، سید محمد تقی. مفاهیم بنیادی پایگاه داده ها. انتشارات جلوه [۲]
- مقسمی، حمید رضا. پایگاه داده ها. گسترش علوم پایه [۳]