



دانشگاه صنعتی امیر کبیر
(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر

پروژه کارشناسی

برنامه تلفن همراه کنترل مودم‌های خانگی

نگارنده

سپهر مقیسهء

استاد راهنما

سجاد شیرعلی شهرضا

بهمن ۱۴۰۲

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

به نام خدا

تعهدنامه اصالت اثر

تاریخ: بهمن ۱۴۰۲

اینجانب سپهر مقیسهء متعهد می‌شوم که مطالب مندرج در این پایان‌نامه حاصل کار اینجانب تحت نظارت و راهنمایی اساتید دانشگاه صنعتی امیرکبیر بوده و به دستاوردهای دیگران که در این پژوهش از آنها استفاده شده است مطابق مقررات و روال متعارف ارجاع و در فهرست منابع و مآخذ ذکر گردیده است. این پایان‌نامه قبل برای احراز هیچ مدرک هم سطح یا بالاتر ارائه نگردیده است. در صورت اثبات تخلف در هر زمان، مدرک تحصیلی صادر شده توسط دانشگاه از درجه اعتبار ساقط بوده و دانشگاه حق پیگیری قانونی خواهد داشت. کلیه نتایج و حقوق حاصل از این پایان‌نامه متعلق به دانشگاه صنعتی امیرکبیر است. هرگونه استفاده از نتایج علمی و عملی، واگذاری اطلاعات به دیگران یا چاپ و تکثیر، نسخه‌برداری، ترجمه و اقتباس از این پایان‌نامه بدون موافقت کتبی دانشگاه صنعتی امیرکبیر ممنوع است. نقل مطالب با ذکر مآخذ بلامانع است.

سپهر مقیسهء

امضا

اینجانب بدین وسیله از استاد گرامی جناب آقای سجاد شیرعلی شهرضا نهایت سپاس و تشکر خود را دارم که با انتقال دانش خود در مورد روش های گردآوری مطالب و انجام تحقیق و ثبت اطلاعات به دست آمده زمینه لازم برای انجام این پایان نامه را فراهم نمودند. بر این باور دارم بدون کمک ایشان، پایان این مسیر نامشخص بود.

سپهر مقیسهء

بهمن ۱۴۰۲

چکیده

در حال حاضر، با پیشرفت فناوری و رشد روزافزون ارتباطات اینترنتی، اینترنت به یکی از پایه‌های اصلی جوامع مدرن تبدیل شده است. از ارتباطات شخصی و اجتماعی گرفته تا فعالیت‌های تجاری و صنعتی، اینترنت وابستگی بسیاری از افراد و سازمان‌ها را به خود جلب کرده است. با این وجود، هنگامی که افراد در خانه‌هایشان به اینترنت متصل می‌شوند، ممکن است با مشکلاتی روبرو شوند که از جمله آن‌ها می‌توان به پیچیدگی تنظیم مودم‌ها اشاره کرد. بسیاری از کاربران، به‌ویژه کسانی که تخصص فنی در زمینه‌ی شبکه ندارند، با مشکلاتی مانند نصب، پیکربندی، و تنظیمات مودم‌ها روبرو می‌شوند. این مشکلات می‌توانند باعث کاهش سرعت اینترنت، قطع و وصل مکرر اتصال، یا حتی ناتوانی در دسترسی به برخی از خدمات آنلاین شوند. برای حل این مسائل و ارتقای تجربه کاربری، ضروری است که راهکارهایی ساده و کارآمد برای تنظیم مودم‌ها ارائه شود.

در این راستا، طراحی یک برنامه موبایل که به کاربران کمک کند تا به سرعت و به‌سهولت تنظیمات مودم خود را انجام دهند، اهمیت زیادی دارد. این برنامه با ارائه راهنمایی‌ها و ابزارهایی ساده و قابل فهم، به کاربران این امکان را می‌دهد تا به طور مستقل و بدون نیاز به دانش فنی عمیق، مودم خود را پیکربندی کرده و از امکانات آن به‌صورت کامل بهره‌مند شوند.

باتوجه به اهمیت اینترنت و نیاز روزافزون جامعه به دسترسی به آن، ارائه چنین راهکارهایی می‌تواند به بهبود کیفیت زندگی افراد و افزایش بهره‌وری در جوامع مدرن کمک شایانی نماید. این توسعه در نهایت منجر به توسعه‌ی پایدار و پویای فضای اینترنتی خواهد شد و در بهبود ارتباطات و ارتقای فناوری‌های ارتباطی نقش مؤثری خواهد داشت.

کلیدواژه‌ها: مودم‌های اینترنت، پایتون، برنامه تلفن هوشمند، نقطه دسترسی، شبکه، اندروید، طراحی نرم‌افزار

فهرست

فصل اول مقدمه	۱
۱-۱ مقدمه	۲
۲-۱ اوپن دبلیو - آر - تی چیست؟	۲
۱-۲-۱ ویژگی‌های اوپن دبلیو - آر - تی	۳
۲-۲-۱ سایر سیستم‌عامل‌ها	۳
۲-۱ کارهای مشابه	۴
۱-۲-۱ مزیت‌های برنامه ما نسبت به کارهای مشابه و اهداف پروژه	۶
۳-۱ ساختار پایان‌نامه	۷
فصل دوم ابزار مورد استفاده در برنامه	۸
۱-۲ ابزار طراحی سامانه	۹
۲-۲ ابزار طراحی سامانه	۹
۱-۲-۲ پایتون	۹
۲-۲-۲ بش اسکریپ	۱۰
۳-۲-۲ بانک داده	۱۱
۴-۲-۲ طراحی نمودار	۱۱
۳-۲ روسازه	۱۲
۱-۳-۲ کیوی	۱۲
۴-۲ ابزار خروجی گرفتن	۱۴
۱-۴-۲ بیلدوزر	۱۴
۲-۴-۲ جیسون آر - پی - سی	۱۵
۳-۴-۲ آ-د-ب	۱۵
۵-۲ چرا از دیگر ابزار استفاده نشد؟	۱۶
۶-۲ جمع‌بندی	۱۶
فصل سوم بررسی ساختار برنامه	۱۷
۱-۳ کلیات نما	۱۸
۲-۳ معماری سامانه	۱۸
۳-۳ معماری بانک داده	۱۹
۴-۳ معماری پایتون	۲۱
۵-۳ معماری منوها	۲۲
۶-۳ جمع‌بندی	۲۶

فصل چهارم نحوه پیاده‌سازی	۳۳
۴-۱ جزئیات فنی برنامه	۲۹
۴-۲ صفحه ورود	۳۰
۴-۲ صفحه خانه	۳۲
۴-۳ نمودارها	۳۴
۴-۴ نام‌گذاری دستگاه‌ها	۳۷
۴-۵ جمع‌بندی	۳۹
فصل پنجم خروجی گرفتن و چالش‌های پیش‌آمده	۴۰
۵-۱ مقدمه	۴۱
۵-۲ استفاده از ابزار بیلدوزر	۴۱
۵-۳ استفاده از ابزار آ-د-ب	۴۵
۵-۳-۱ استفاده از ابزار عیب‌یابی	۴۶
۵-۴ چالش‌های مطرح شده	۴۷
۵-۶ جمع‌بندی	۴۹
فصل ششم نظرسنجی و تست	۳۳
۶-۱ مقدمه	۵۱
۶-۲ نظرسنجی	۵۱
۶-۲-۱ نقدهای کاربران	۵۲
۶-۴ جمع‌بندی	۵۶
فصل هفتم جمع‌بندی، نتیجه‌گیری و هدف‌های آینده	۵۷
۷-۱ جمع‌بندی اجمالی	۵۹
۷-۲ نتیجه‌گیری	۶۰
۷-۳ هدف و کارهای آینده	۶۱
منابع و مراجع	۶۲
پیوست	۶۴

فهرست شکل‌ها

شکل ۱- صفحه ورودی برنامه	۴
شکل ۲- لوگوی برنامه	۵
شکل ۳- کیوی	۱۴
شکل ۴- صفحه ورود	۱۸
شکل ۵- صفحه خانه	۱۹
شکل ۶- بانک داده های موجود	۲۰
شکل ۷- روابط بانک داده ها	۲۰
شکل ۸- طراحی روسازه	۲۱
شکل ۹- صفحه رمز عبور	۲۲
شکل ۱۰- صفحه آزمون سرعت	۲۳
شکل ۱۱- نمودار آزمون سرعت سرور	۲۴
شکل ۱۲- نمودار آزمون سرعت کاربر	۲۵
شکل ۱۳- نمودار زمان اتصال هر دستگاه به مودم	۲۵
شکل ۱۴- نمودار حجم مصرفی	۲۶
شکل ۱۵- تابع اصلی	۲۹
شکل ۱۶- صفحات برنامه	۳۰
شکل ۱۷- تنظیمات روسازه قسمت اطلاعات ورود	۳۰
شکل ۱۸- طریق کارکرد صفحه ورود	۳۱
شکل ۱۹- اطلاعات مربوط به پیام	۳۱
شکل ۲۰- سرویس صفحه ورود	۳۲
شکل ۲۱- تنظیمات لیست دستگاه های متصل	۳۳
شکل ۲۲- ادامه کد لیست دستگاه های متصل	۳۳
شکل ۲۳- اطلاعات دستگاه های متصل	۳۴
شکل ۲۴- کوثری های مورد استفاده	۳۴
شکل ۲۵- بش مورد استفاده برای تاریخچه سرعت	۳۵
شکل ۲۶- دستور مورد استفاده	۳۵
شکل ۲۷- دستور مورد استفاده دوم	۳۵
شکل ۲۸- منطق سرویس آزمون سرعت سرور	۳۶
شکل ۲۹- پیاده سازی روسازه نمودار	۳۶
شکل ۳۰- مقدار حجم مصرفی روزانه به کمک vnstat	۳۶
شکل ۳۱- گرفتن اطلاعات حجم مصرفی	۳۷
شکل ۳۲- تبدیل سازی مقادیر	۳۷
شکل ۳۳- ذخیره سازی اطلاعات ورود	۳۷
شکل ۳۴- نام گذاری	۳۸
شکل ۳۵- روسازه تغییر نام	۳۸
شکل ۳۶- زیرسازه تغییر نام	۳۹
شکل ۳۷- نصب بیلدوزر	۴۱
شکل ۳۸- نصب پیش نیاز ها	۴۱

شکل ۳۹- ایجاد فایل تنظیمات.....	۴۲
شکل ۴۰- قسمت‌های تغییر داده شده از فایل تنظیمات.....	۴۲
شکل ۴۱- قسمت واردکردن پیش‌نیازها.....	۴۲
شکل ۴۲- تنظیمات لوگوی برنامه.....	۴۳
شکل ۴۳- دسترسی‌های برنامه.....	۴۳
شکل ۴۴- دستور خروجی گرفتن.....	۴۳
شکل ۴۵- تصویری از محیط بیلدور.....	۴۴
شکل ۴۶- هنگام خروجی گرفتن برنامه.....	۴۴
شکل ۴۷- دستور لیست دستگاه‌های متصل.....	۴۵
شکل ۴۸- دستور انتقال پورت.....	۴۵
شکل ۴۹- دستور نصب برنامه بر روی لینوکس.....	۴۵
شکل ۵۰- دستور اتصال به دستگاه.....	۴۶
شکل ۵۱- دستور نصب برنامه.....	۴۶
شکل ۵۲- دستور عیب‌یابی.....	۴۶
شکل ۵۳- دلیل عیب برنامه.....	۴۷
شکل ۵۴- روش ایجاد شده.....	۴۸
شکل ۵۵- راه‌حل ارائه شده توسط یکی از کاربران.....	۴۸
شکل ۵۶- نتایج تست برنامه.....	۵۴
شکل ۵۷- آزمون تابع تست سرعت کلاینت.....	۵۴
شکل ۵۸- تست تابع ثبت اطلاعات ورود در بانک داده.....	۵۵
شکل ۵۹- تست آزمون سرعت سرور.....	۵۶
شکل ۶۰- صفحه موردنظر.....	۵۸
شکل ۶۱- دالود فایل موردنظر.....	۵۹
شکل ۶۲- صفحه تنظیمات مودم.....	۶۰

فهرست جداول

جدول ۱- نظرسنجی کاربران.....	۵۱
------------------------------	----

فصل اول

مقدمه

۱-۱ مقدمه

مطمئناً همگان بر این آگاه هستند که دنیای امروز بدون اینترنت امکان‌پذیر نیست؛ چرا که اکثر نیازهای روزمره ما از جمله نیازهایی مانند خوراک، خرید و حمل نقل به اینترنت وابسته است و عملاً می‌توان در این خلاصه کرد که زندگی بدون اینترنت مانند زندگی بدون اکسیژن است. همان‌طور که می‌دانید برای مدیریت اینترنت خانگی حداقل نیاز به یک دانش‌بنیادی از رایانه است و در همین راستا، مودم‌ها به‌عنوان ابزارهای ارتباطی اساسی در جوامع مدرن برای اتصال به اینترنت و دسترسی به خدمات مختلف استفاده می‌شوند. با این حال، بسیاری از کاربران خانگی با مشکلاتی در تنظیم و استفاده از مودم‌های خود روبه‌رو می‌شوند که این مشکلات می‌تواند از دسترسی به اینترنت بهینه و استفاده از خدمات آن محدود کند. به همین دلیل بسیاری از افراد که خود این دانش را دارا نیستند و یا دست کمکی که دارای این اطلاعات باشند را ندارند، از استفاده از اینترنت پرهیز می‌کنند. چرا که حتی نمی‌توانند اینترنت خانگی خود را مدیریت کنند و بتوانند اطلاعات آن را پیدا و در جهت بهبود کارایی اینترنت استفاده کنند؛ بنابراین، با توجه به چالش‌های موجود و به‌منظور بهبود تجربه کاربری، ارائه یک برنامه موبایل جهت تنظیمات آسان مودم‌ها امری ضروری و اساسی به نظر می‌رسد. این برنامه موبایل با ارائه راهکارهایی ساده و کاربرپسند، به کاربران کمک می‌کند تا به سرعت و به‌سهولت تنظیمات مودم خود را انجام داده و از تمامی امکانات آن بهره‌مند شوند.

۱-۲ اوپن دبلو - آر - تی چیست؟

اوپن دبلو - آر - تی^۱ یک سیستم‌عامل مبتنی بر لینوکس است که برای روترها و تجهیزات شبکه مورد استفاده قرار می‌گیرد. این سیستم‌عامل به‌عنوان یک سیستم‌عامل متن‌باز^۲ توسعه‌یافته است و به کاربران امکان می‌دهد تا قابلیت‌های روترهای خود را به شکلی قابل تنظیم و سفارشی‌سازی شده ارتقا دهند.

مهمترین ویژگی اوپن دبلو-آر-تی قابلیت اجرای برنامه‌ها و نرم‌افزارهای دلخواه بر روی روترها است. این به کاربران اجازه می‌دهد تا از روترهای خود به‌عنوان یک سرور کوچک، فایروال، فیلترشکن^۳، یا حتی یک آنتن بی‌سیم استفاده کنند. با استفاده از اوپن دبلو-آر-تی، کاربران قادرند تا به‌آسانی تنظیمات شبکه، امنیت، سرویس‌های شبکه، و بسیاری از ویژگی‌های دیگر را مطابق با نیازهای خود تغییر دهند. همچنین این سیستم‌عامل دارای یک جامعه‌ی فعال توسعه‌دهندگان و کاربران است که به بهتر شدن و توسعه‌ی مستمر آن کمک می‌کنند.[۳]

^۱ OpenWRT

^۲ Open Source

^۳ VPN

۱-۲-۱ ویژگی‌های اوپن دبلو - آر - تی

اوپن دبلو - آر - تی چند ویژگی منحصر به فرد و برجسته دارد که آن را از سایر سیستم‌های عامل موجود برای روترها و تجهیزات شبکه متمایز می‌کند.

متن باز: یکی از ویژگی‌های بارز این است که کاملاً متن باز است، به این معنی که کد منبع آن به طور عمومی قابل دسترسی و تغییر است. این امکان به توسعه‌دهندگان و کاربران اجازه می‌دهد تا به آسانی تغییرات لازم را اعمال و سفارشی‌سازی کنند.

۱. **پشتیبانی از برنامه‌های دلخواه:** قابلیت اجرای برنامه‌های و نرم‌افزارهای دلخواه است. این به کاربران اجازه می‌دهد تا از روترهای خود به عنوان یک سرور وب، فیلترشکن^۴، پروکسی، و غیره استفاده کنند.
۲. **جامعه فعال:** دارای یک جامعه فعال از توسعه‌دهندگان و کاربران است که به بهبود و توسعه‌ی مستمر آن کمک می‌کنند. این جامعه فعال منجر به ایجاد و پشتیبانی از اکوسیستمی پویا و پرنشاط از برنامه‌ها و ابزارهای مختلف برای اوپن دبلو - آر - تی شده است.
۳. **دارای قابلیت نصب افزونه‌های مختلف:** مهم‌ترین قابلیت این سیستم عامل که آن را از سایر سیستم‌عامل‌ها بسیار متمایز می‌کند، وجود قابلیت شخصی‌سازی است و باعث می‌شود کاربر برای نیاز خود افزونه‌های مختلف را نصب کند.

در ادامه به دو صورت کارهای مشابه را بررسی می‌کنیم؛ اول سیستم‌عامل‌های مشابه و در انتها کارهای مشابه برای سیستم‌عامل مورد تمرکز ما. [۴]

برای نصب این سیستم عامل به پیوست ۱ مراجعه کنید.

۱-۲-۲ سایر سیستم‌عامل‌ها

علاوه بر OpenWRT، سیستم‌عامل‌های دیگری نیز برای استفاده در تجهیزات شبکه موجود هستند که ممکن است بر اساس نیازها و شرایط خاص به کار گرفته شوند. برخی از این سیستم‌عامل‌ها شامل موارد زیر هستند:

۱. **DD-WRT:** مانند OpenWRT، DD-WRT نیز یک سیستم‌عامل مبتنی بر لینوکس برای روترها و تجهیزات شبکه است. این سیستم‌عامل امکانات متنوعی برای سفارشی‌سازی و مدیریت شبکه ارائه می‌دهد.
۲. **Tomato:** نیز یک سیستم‌عامل معروف برای روترها است که امکانات بسیاری از جمله فیلترشکن، QoS، و شبکه بی‌سیم دو باند را فراهم می‌کند.

۳. **pfSense**: یک سیستم عامل مبتنی بر FreeBSD است که به عنوان یک فایروال و روتر برای استفاده در شبکه های متوسط تا بزرگ مناسب است. این سیستم عامل دارای امکانات امنیتی بسیار قوی و متنوعی است.

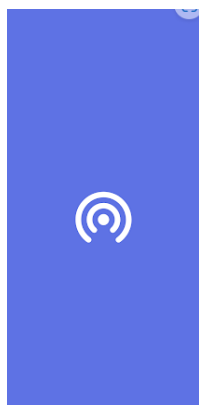
این فقط چند نمونه از سیستم عامل های موجود برای استفاده در تجهیزات شبکه هستند و هر کدام از آنها ممکن است بر اساس نیازهای خاص مورد استفاده قرار گیرند.[۳]

۱-۲ کارهای مشابه

در این قسمت کارهای مشابه که سایر افراد با کمک اوپن دبیو - آر - تی ایجاد کردند رو به نمایش می گذاریم. این کارها شامل برنامه های تلفن همراه، برنامه های تحت وب و برنامه های کنسولی می شود که برای مودم ها با سیستم عامل اوپن دبیو - آر - تی تعبیه شدند و قابلیت کنترل مودم را به کاربر می دهند، می شوند. در ادامه هم نقاط قوت و ضعف هر کدام را قرار دادیم.

۱. OpenWR LuCI App

این برنامه از فروشگاه های برنامه های اندرویدی قابلیت دانلود دارد. شکل زیر صفحه ورود برنامه است.



شکل ۱- صفحه ورودی برنامه

مزیت های برنامه:

- دسترسی به بسیاری از برنامه های LuCI مانند اعم از آمار^۵ LuCI
- قابلیت ایجاد ip شخصی سازی شده
- قابلیت اتصال به تنظیمات سایر مودم هایی که سیستم عامل متفاوت دارند

نقاط ضعف:

^۵ Luci statistics

- **محدودیت‌های عملکردی:** LuCI در برخی از امکانات و قابلیت‌های پیشرفته ممکن است محدودیت‌هایی داشته باشد. برای کاربرانی که به دنبال تنظیمات پیشرفته و پیچیده‌تر هستند، LuCI ممکن است کافی نباشد و نیاز به استفاده از رابط خط فرمان یا تنظیمات مستقیم در فایل‌های پیکربندی داشته باشند.
 - **مصرف منابع سیستمی:** LuCI به دلیل واسط گرافیکی آن و تمامیت قابلیت‌ها، ممکن است مصرف منابع سیستمی بیشتری داشته باشد. این موضوع می‌تواند به‌ویژه برای دستگاه‌هایی با منابع سخت‌افزاری محدود، مشکل ایجاد کند و عملکرد دستگاه را کاهش دهد.
 - **کندی در عملکرد:** برخی کاربران گزارش داده‌اند که LuCI در برخی موارد ممکن است با کندی در عملکرد مواجه شود، به‌خصوص در دستگاه‌هایی که دارای پردازنده و منابع سخت‌افزاری ضعیفی هستند.
- در کل این برنامه عملاً صفحات وب مودم را به نمایش می‌گذارد و عملاً هیچ قابلیت جدیدی دارا نیست. در زمانی که علاقه دارید به‌جای استفاده از مرورگر از برنامه استفاده کنید می‌تواند این برنامه را استفاده کنید.

۲. OpenWRT Manager

ای برنامه به نسبت برنامه قبلی عملکرد بهتری دارد و دارای قابلیت‌های جدید بهتری است.



شکل ۲- لوگوی برنامه

مزیت‌های برنامه:

- **ثبت بیشتر از یک مودم:** می‌توان بیشتر از یک مودم openwrt در این برنامه ذخیره کرد.
- **به‌روزرسانی خودکار:** این برنامه به‌صورت خودکار در زمان تعیین شده توسط کاربر وضعیت مودم‌ها را به‌روزرسانی می‌کند.

- قطع یک کاربر وصل بی سیم: می توان یک کاربر دلخواه را از دسترسی به اینترنت قطع کرد.
- نمایش کامل تاریخچه^۶ سیستم و کرنل مودم

نقاط ضعف:

- قابلیت های محدود و ساده ای دارد.
- در آخر مناسب برای مشاهده وضعیت است و تغییرات خاصی نمی توان روی مودم انجام داد.

۱-۲-۱ مزیت های برنامه ما نسبت به کارهای مشابه و اهداف پروژه

برنامه ما که به اصطلاح در این پروژه از آن به عنوان MyWRT نام می بریم، در مقابل برنامه های بالا دارای مزیت های بیشتری است.

به طور مثال اولین مزیتی که به نسبت برنامه اول دارد این است که کاملاً شخصی سازی شده است و وابسته به صفحات وب نیست و تماماً دارای روسازه^۷ شخصی سازی شده است و فقط تماس هایی با سرور برقرار می کند تا اطلاعات مورد نظر را گرفته تا پردازش کند. دومین مزیت به نسبت سایر برنامه ها دارد این است که ویژگی های کاربردی اعم از آزمون سرعت، نمایش افراد وصل شده به صورت بی سیم به مودم، نمایش تاریخچه های آزمون سرعت، آزمون سرعت توسط مودم به صورت یک ساعت یکبار و قابلیت نام گذاری برای دستگاه های متصل به مودم هستند. نوع خروجی گرفتن برای تلفن همراه متفاوت از نوع زبان برنامه نویسی است و عملاً حالت میکروسرویس را در میان امکانات ایجاد کرده است و فقط کافی است در هنگام خروجی گرفتن برنامه، تنظیمات را برای تلفن همراه های دارای سیستم عامل iOS، مشخص کرد.

از اهداف این برنامه می توان به ایجاد برنامه ای که دارای ویژگی هایی مانند آزمون سرعت و تاریخچه آن، اضافه کردن امنیت اینترنت به کمک مدیریت دستگاه های متصل به صورت نوین تر و مشخص تر به طوری که بتوان عمل نام گذاری را برای آن ها انجام داد، انجام بعضی از سناریو ها به صورت خودکار مانند آزمون سرعت که توسط مودم در ساعاتی معین انجام می شود و بتواند میزان دقایق اتصالی افراد به این دستگاه را نیز مشخص کند؛ همچنین نیاز به ذکر است تمامی این اهداف باید در مسیر خودکارسازی نیز انجام شوند و در زمانی معین برنامه بتواند اطلاعات خود را به روزرسانی کند.

با تمامی این امکانات کاربرها با دانش ناچیز از دنیای برنامه نویسی نیز می توانند اینترنت خانگی خود را مدیریت کنند و به راحتی با آن ارتباط ایجاد کنند.

⁶ Log

⁷ Front-End

۱-۳ ساختار پایان نامه

فصل اول شامل مقدماتی همچون تعریف سیستم عامل، بررسی پیشینه کارهای مشابه، نقاط مثبت و منفی این برنامه ها و اهداف پروژه است. در فصل دوم به معرفی ابزار استفاده شده در پروژه و دلیل استفاده از آنان می پردازیم تا دید بهتری قبل از شروع معرفی ساختار خود سامانه داشته باشیم. در فصل سوم به معرفی سامانه، انواع صفحه ها و نحوه کارکرد هر بخش می پردازیم تا تمامی وجوه سامانه مورد بررسی قرار گیرد. در فصل چهارم به بررسی پیاده سازه سامانه می پردازیم و تمامی بخش های ایجاد شده را مطالعه می کنیم. در فصل پنجم نحوه خروجی گرفتن از سامانه و همچنین نصب و عیب یابی را به اشتراک می گذاریم، در فصل ۶ نتایج استفاده افراد از این سامانه و نتایج تست را لیست می کنیم و در نهایت در فصل ۷ مروری بر آنچه که در مقاله و پروژه گذشت می کنیم و می توان اهدافی که در آینده نه چندان دور، دستیابی به آنها ناممکن نیست، بپردازیم.

فصل دوم

ابزار مورد استفاده در برنامه

۲-۱ ابزار طراحی سامانه

هدف ما، طراحی یک برنامه برای تسهیل مدیریت مودم‌های خانگی و مشاهده وضعیت اینترنت است. واضحاً این برنامه نیاز به یک روسازه مناسب و همچنین یک سرویس زیرسازه^۸ قوی دارد تا بتواند با مودم تعامل داشته باشد و این اطلاعات را پردازش کند و نتیجه را به روسازه ارسال کند تا بتوانیم آن اطلاعات را به گونه‌های مختلف (جدول، نمودار و لیست) نمایش دهیم. همچنین نیاز به نصب بسته‌های متعددی مانند مدیریت بانک داده، ارسال تماس و برقراری ارتباط، ابزار کرل^۹ بر روی مودم نیاز داریم که در ادامه آموزش نصب این بسته‌ها را نمایش می‌دهیم. نیاز به اشاره است علاوه بر این بسته‌ها نیاز به کتابخانه‌های موردنظر در پایتون^{۱۰} نیز هستیم. چرا که زبان اصلی مورد استفاده برای روسازه و زیرسازه پایتون است و در مواقعی بر روی مودم از زبان بش^{۱۱} استفاده می‌کنیم.

در ادامه هر دو روسازه و زیرسازه را به صورت جدا مورد بررسی قرار می‌دهیم تا کاملاً با ابزار مورد استفاده آشنا شویم.

۲-۲ ابزار طراحی سامانه

۲-۲-۱ پایتون

پایتون یک زبان برنامه‌نویسی است که به شکل گسترده‌ای در زمینه‌های مختلف استفاده می‌شود. طی دهه گذشته، پایتون مسیری روشن را از طریق دنیای فناوری طی کرده است و هیچ نشانه‌ای از توقف ندارد. در واقع، طبق یک نظرسنجی، در سال ۲۰۲۳ پایتون به عنوان دومین زبان برنامه‌نویسی موردنظر جامعه توسعه‌دهندگان پس از جاوا اسکریپت انتخاب شده است.

توسعه وب با پایتون برای انواع پروژه‌ها از ساده تا پیچیده مناسب است. این زبان در زمینه‌های مختلفی مانند مسافرت، بهداشت، حمل و نقل، امور مالی و بسیاری دیگر برای توسعه وب و آزمون نرم‌افزار، اسکریپت‌نویسی و تولید به طور گسترده‌ای استفاده می‌شود. پایتون مشابه زبان انگلیسی است که در زندگی روزمره ما استفاده می‌شود. سادگی دستور زبان امکان می‌دهد تا شما با سیستم‌های پیچیده مواجه شوید و اطمینان حاصل کنید که تمام عناصر رابطه واضحی با یکدیگر دارند. به لطف این امر، بیشتر کدنویسان مبتدی می‌توانند زبان را یاد بگیرند و سریع‌تر به جامعه برنامه‌نویسی بپیوندند. نمایش داده‌ها به صورتی که به راحتی قابل درک باشد، با استفاده از نمودارها و چارت‌های مختلف امکان‌پذیر است. شرکت‌های توسعه وب از کتابخانه‌های پایتون استفاده می‌کنند که امکان بصری‌سازی داده‌ها و ایجاد گزارش‌های واضح و قابل فهم را فراهم می‌کنند. نسبت به زبان‌های کدنویسی، پایتون یک رویکرد کمتر محدود دارد. این زبان چندین الگوی

⁸BackEnd

⁹ Curl

¹⁰ Python

¹¹ Bash Script

برنامه‌نویسی را پشتیبانی می‌کند دیگر مانند جاوا^{۱۲} و می‌تواند انواع مختلفی از سبک‌های برنامه‌نویسی از جمله روش‌های روی‌های، شیء‌گرا و تابعی را پشتیبانی کند. این امر پایتون را یک‌زبان عالی برای شروع‌های نوپا می‌کند؛ زیرا ممکن است در هر لحظه به تغییر رویکرد خود نیاز داشته باشید. برخی از برترین شرکت‌ها از زبان پایتون در پشته فناوری خود استفاده می‌کنند. [۲]

۲-۲-۲ بش اسکرپت

بش اسکرپت یک‌زبان برنامه اسکرپت‌نویسی ساده و درعین‌حال پرکاربرد است که دستورات گنو/لینوکس^{۱۳} را در یک فایل متنی کنترل می‌کند. بش یک پردازشگر دستور است که معمولاً در یک پنجره ترمینال اجرا می‌شود. این پوسته مانند همه پوسته‌های یونیکس، می‌تواند دستوراتش را از یک فایل بگیرد. همچنین این پوسته کلمات اساسی، قواعد دستوری، و بقیه ویژگی‌های اساسی را از پوسته‌ی بورن کپی کرده است. ویژگی‌های دیگری مثل تاریخ از پوسته سی و پوسته کورن کپی شده است. بش منطبق با استاندارد پازیکس است.

نام بش سرنام در انگلیسی یک بازی با کلمات و ایهام است به‌این‌ترتیب که هم اشاره به پوسته بورن می‌کند و هم به کلمه تولد دوباره^{۱۴} اشاره دارد که به معنی دوباره متولد شده است. یک حفره امنیتی موجود در پوسته بش در سپتامبر سال ۲۰۱۴ کشف شد و به‌سرعت باعث یک سری حمله در پهنه اینترنت شد. وقتی محل اشکال پیدا شد، زود تغییرات به‌روزرسانی انجام شد و مسئله رفع شد. قواعد دستوری فرمان‌های بش ابر مجموعه‌ی قواعد دستوری فرمان‌های پوسته بورن هستند. بش از افزونه آکولادی، تکمیل خط فرمان، اشکال‌زدایی کردن ابتدایی، رفع‌ورجوع استثناها و بقیه‌ی موارد پشتیبانی می‌کند. بش می‌تواند اکثریت بسیار بالایی از اسکرپت‌های پوسته‌ی بورن را بدون نیاز به تغییر اجرا کند. بش بعضی از ایده‌ها را از پوسته کورن^{۱۵} و گرفته است.

یک اسکرپت بش را می‌توانید به‌عنوان یک برنامه کوچک در نظر بگیرید که شامل دستورات لینوکس است. شما می‌توانید اسکرپت‌های بش را برای انجام تقریباً هر کاری بنویسید، مانند ایجاد پشتیبان، تنظیم متغیرها، باز کردن برنامه‌ها، پوشش به دایرکتوری‌های خاص، ایجاد فایل‌ها، و بسیاری موارد دیگر. در واقع، با یک کمی خلاقیت، امکانات اسکرپت‌های بش نامحدود است. اما به یاد داشته باشید، اسکرپت‌های بش فقط یک اسکرپت هستند... این‌ها برنامه‌های رابط کاربری گرافیکی نیستند، و هیچ رابط گرافیکی‌ای وجود ندارد که شما را از طریق فرایند ایجاد یک اسکرپت هدایت کند. به‌عبارت‌دیگر، اسکرپت‌های بش یک کم پیچیده‌تر از استفاده از یک پردازنده متنی، مرورگر وب، یا کاربر ایمیل هستند.

¹² JAVA

¹³ GNU/Linux

¹⁴ Born Again

¹⁵ KHS

این به این معنی نیست که اسکریپت‌های بش فقط برای کاربران پیشرفته مناسب هستند. بله، اسکریپت‌های بش می‌توانند بسیار پیچیده شوند، اما در ابتدا می‌توانند به‌سادگی شما مورد استفاده قرار گیرند.

اسکریپت‌های بش مستقیماً کامپایل نمی‌شوند؛ مانند زبان‌های برنامه‌نویسی دیگر مانند سی یا سی پلاس پلاس. به جای آن، اسکریپت‌های بش به‌عنوان متنی تفسیر می‌شوند و توسط مفسر بش^{۱۶} اجرا می‌شوند. برای اجرای یک اسکریپت بش، نیازی به کامپایل کردن آن نیست؛ به جای آن، می‌توانید به‌صورت مستقیم فایل اسکریپت را اجرا کنید. [۱۰]

برای آموزش بش‌های به پیوست ۲ مراجعه نمایید.

۳-۲-۲ بانک داده

اس - کیو - لایت در تمام تلفن‌های همراه و بیشتر کامپیوترها داخلی است و در داخل بی‌شماری از برنامه‌های دیگر که مردم روزانه استفاده می‌کنند، بسته‌بندی شده است. قالب فایل اس - کیو - لایت پایدار، چند سکوی و سازگاری با نسخه‌های قبلی را دارد و توسعه‌دهندگان عهد دارند که این ویژگی‌ها را تا سال ۲۰۵۰ حفظ خواهند کرد. همچنین به شما امکان می‌دهد تا هم‌زمان بر روی چندین پایگاه داده در یک جلسه کار کنید، این امر آن را انعطاف‌پذیر می‌کند و کد منبع اس - کیو - لایت در دامنه عمومی است و برای همه به‌صورت رایگان برای هر هدفی قابل استفاده است. برای پروژه‌های کوچک تا متوسط یا در مراحل اولیه توسعه، اس - کیو - لایت می‌تواند به‌عنوان یک انتخاب مناسب عمل کند. این به توسعه‌دهندگان امکان می‌دهد به‌سرعت یک پایگاه داده را بدون هزینه پیکربندی زیاد راه‌اندازی کنند. اوپن دبیو - آر - تی از اس - کیو - لایت پشتیبانی دارد که باعث ادغام ساده آن با چارچوب می‌شود. این امر به توسعه‌دهندگان اجازه می‌دهد که بدون نیاز به راه‌اندازی اضافی برای یک سیستم پایگاه داده متفاوت، به‌سرعت به کار پردازند.

۴-۲-۲ طراحی نمودار

مت‌پلاتلیب^{۱۷} یک کتابخانه متن‌باز و بسیار قدرتمند است که برای تجزیه و تحلیل داده و رسم نمودارهای دوبعدی و سه‌بعدی در زبان برنامه‌نویسی پایتون استفاده می‌شود. این کتابخانه به برنامه‌نویسان و داده‌کاوانان امکان می‌دهد تا داده‌های خود را به‌صورت بصری تحلیل کنند و الگوها و روابط مختلف در داده‌ها را شناسایی کنند. مت‌پلاتلیب به شما امکان می‌دهد تا نمودارهای مختلفی از داده‌های خود را با سفارشی‌سازی‌های مختلف ایجاد کنید. شما می‌توانید رنگ‌ها، اندازه‌ها، استایل‌ها و محورهای نمودارهای خود را به‌دلخواه تغییر دهید. همچنین، می‌توانید عناصر متنی و گرافیکی مانند متن، خطوط، مستطیل‌ها، دایره‌ها و... را به نمودارهای خود اضافه کنید تا اطلاعات را بهبود بخشید. این کتابخانه امکان رسم نمودارهای سه‌بعدی را نیز فراهم می‌کند که به شما امکان می‌دهد داده‌های خود را در یک فضای سه‌بعدی بررسی کنید. همچنین،

¹⁶ Bash interpreter

¹⁷ Matplotlib

ابزارهای تعاملی مانند افزودن ابزارهای ناوبری، ایجاد منوها، واکنش به رویدادها و... را برای کاربران فراهم می‌کند تا با داده‌ها تعامل بیشتری داشته باشند.

با استفاده از مت‌پلاتیب، شما می‌توانید داده‌های خود را به صورت بصری نشان داده و تحلیل کنید و اطلاعات مفیدی از آنها به دست آورید. این کتابخانه ابزار بسیار کارآمدی برای تجزیه و تحلیل داده و ارائه گزارش‌های تجزیه و تحلیلی است که به برنامه‌نویسان و تحلیل‌گران کمک می‌کند تا داده‌ها را به شکل بهتری درک کنند و تصمیم‌گیری‌های بهتری اتخاذ کنند. [۱۱]

۳-۲ روسازه

برای روسازه از کتابخانه کیوی^{۱۸} پایتون استفاده کردیم و در ادامه نیز از ابزار بیلدورز^{۱۹} استفاده می‌کنیم تا از این برنامه خروجی اندروید بگیریم. نیاز به ذکر است این ابزار وابسته به خود کیوی است و از هر برنامه‌ای که با کیوی نوشته شده باشد می‌توان با بیلدورز برای دو سیستم عامل اندروید و آی - او - اس خروجی گرفت.

۱-۳-۲ کیوی

کیوی و کیوی ام - دی هر دو ابزارهایی هستند که برای توسعه برنامه‌های کاربردی و رابط‌های کاربری استفاده می‌شوند. این ابزارها به برنامه‌نویسان اجازه می‌دهند تا برنامه‌های با رابط‌های کاربری جذاب و انعطاف‌پذیر را برای انواع دستگاه‌ها مانند تلفن‌های هوشمند، تبلت‌ها و کامپیوترها ایجاد کنند.

۲-۳-۱-۱ کیوی:

کیوی یک ابزار برنامه‌نویسی است که به شما این امکان را می‌دهد تا برنامه‌هایی با رابط‌های کاربری خلاقانه و فوق‌العاده قابل استفاده برای دستگاه‌های مختلف بسازید. با استفاده از کیوی، می‌توانید برنامه‌هایی بسازید که همراه با حرکات چندانی بر روی صفحه لمسی و رابط‌های کاربری زیبا باشند. این ابزار از زبان برنامه‌نویسی پایتون استفاده می‌کند که یک زبان برنامه‌نویسی ساده و قدرتمند است.

¹⁸ kivy

¹⁹ Buildozer

۲-۳-۱-۲ کیوی ام - دی ۲۰:

کیوی ام - دی یک کتابخانه است که بر پایه کیوی ساخته شده و از طراحی‌های متریکال دیزاین گوگل^{۲۱} الهام گرفته است. این کتابخانه به شما ابزارها و قطعات رابط کاربری متریکال دیزاین را در اختیار می‌گذارد که می‌توانید آنها را در برنامه‌های خود استفاده کنید. با استفاده از کیوی ام - دی، می‌توانید رابط‌های کاربری جذاب و انعطاف‌پذیری برای برنامه‌های خود بسازید که به شکل‌ها و استانداردهای طراحی متریکال دیزاین گوگل مطابقت دارند.

۲-۳-۱-۳ چرا از آنها برای توسعه فرانت استفاده می‌شود؟

۱. **انعطاف‌پذیری:** هر دو کیوی و کیوی ام - دی ابزارهایی انعطاف‌پذیر هستند که به شما اجازه می‌دهند تا رابط‌های کاربری چند سکو^{۲۲} با استفاده از کدهای یکسان بسازید. این امر به شما کمک می‌کند تا زمان و هزینه توسعه را کاهش دهید.

۲. **رابط کاربری زیبا:** کیوی ام - دی از طراحی متریکال دیزاین الهام گرفته است که به رابط کاربری برنامه‌های شما یک ظاهر زیبا و حرفه‌ای می‌بخشد. این به کاربران احساس رضایت بیشتری می‌دهد و تجربه کاربری بهتری را ارائه می‌دهد.

۳. **پشتیبانی از چند سکو:** با استفاده از کیوی و کیوی ام - دی، می‌توانید برنامه‌هایی بسازید که بر روی انواع دستگاه‌ها از جمله تلفن‌های هوشمند، تبلت‌ها و کامپیوترها اجرا شوند. این به شما این امکان را می‌دهد که بازار بزرگی را برای برنامه‌های خود هدف قرار دهید.

۴. **جامعیت:** کیوی و کیوی ام - دی دارای ابزارها و قطعات مختلفی هستند که به شما اجازه می‌دهند تا انواع عناصر رابط کاربری از جمله دکمه‌ها، منوها، لیست‌ها و... را در برنامه‌های خود استفاده کنید. این جامعیت باعث می‌شود که بتوانید رابط‌های کاربری متنوعی را بسازید.

باتوجه به این مزایا، کیوی و کیوی ام - دی ابزارهای محبوبی برای توسعه فرانت برنامه‌ها هستند که توسعه‌دهندگان را قادر می‌سازند تا برنامه‌هایی با رابط کاربری زیبا و کارآمد بسازند و آنها را بر روی انواع دستگاه‌ها اجرا کنند. شکل زیر لوگوی ابزار را نمایش می‌دهد.

²⁰ KivyMd

²¹ Google

²² platform



شکل ۳- کیوی

باتوجه به روند رو به افزایش استفاده از کیوی و کیوی ام - دی و همچنین روندهای مشابه در دیگر چارچوب‌ها و ابزارهای فرانت‌اند، می‌توان به طور تخمینی گفت که در حال حاضر تقریباً 5-10% از برنامه‌نویسان فرانت‌اند از کیوی و کیوی ام - دی برای توسعه برنامه‌های خود استفاده می‌کنند. این عدد تقریبی براساس روندها و تجربیات گذشته برآورد شده است و ممکن است در آینده تغییر کند. [۹]

۲-۴ ابزار خروجی گرفتن

در ادامه ابزار مورد استفاده برای خروجی اندروید را مورد بررسی قرار می‌دهیم که با این ابزار می‌توان کدهای دارای کیوی (می‌توانند دارای منطق باشند) را تبدیل به خروجی صفر - یک مناسب اندروید می‌کند. توجه کنید که این ابزار به دلیل نو بودن ممکن است در بسیاری از موارد مشکلات متعددی داشته باشد.

۲-۴-۱ بیلدوزر^{۲۳}

بیلدوزر یک ابزار بسیار کارآمد و کاربردی برای توسعه‌دهندگان نرم‌افزاری است. این ابزار برای توسعه‌دهندگانی که از زبان برنامه‌نویسی پایتون استفاده می‌کنند، طراحی و توسعه داده شده است. با استفاده از بیلدوزر، توسعه‌دهندگان می‌توانند به سرعت و با سهولت برنامه‌های قابل اجرا بر روی دستگاه‌های همراه خود را ایجاد و توسعه دهند. این ابزار فرایند ساخت برنامه‌های موبایل را بهبود بخشیده و به توسعه‌دهندگان کمک می‌کند تا با سرعت بیشتری برنامه‌ها را توسعه دهند. بیلدوزر علاوه بر سرعت و کارایی، سهولت استفاده، پشتیبانی از سکوها مختلف و تسهیل فرایند توسعه، از دیگر مزایای مهم خود است. با استفاده از این ابزار، توسعه‌دهندگان نیازی به دانش عمیق در زمینه‌های سیستم‌عامل‌ها یا فرایندهای پیچیده ندارند و می‌توانند به راحتی برنامه‌های خود را بسازند و اجرا کنند. این امر زمان و هزینه را برای توسعه‌دهندگان صرفه‌جویی می‌کند. همچنین، با بیلدوزر می‌توانید برنامه‌های خود را برای سیستم‌عامل‌های مختلفی مانند اندروید و آی‌اواس توسعه داده و منتشر کنید.

برای استفاده از بیلدوزر، ابتدا باید آن را بر روی سیستم خود نصب کنید. سپس پروژه‌ی مورد نظر خود را ایجاد کرده و فایل تنظیمات بیلدوزرات اسپک^{۲۴} را پیکربندی کنید. سپس با استفاده از دستورات

²³ buildozer

²⁴ Buildozer.spec

بیلدوزر، بسته‌ی نرم‌افزاری خود را برای اندروید یا آی‌اواس بسازید. در نهایت، بسته‌ی نرم‌افزاری خود را بر روی دستگاه موردنظر خود نصب کرده و برنامه را تست کنید. شاید متوجه شده باشید که این روش شباهت بسیار زیادی به داکر^{۲۵} و فرستادن بر روی داکرها^{۲۶} دارد و عملاً شما تمامی تنظیمات را در داخل یک فایل قرار می‌دهید و با اجرای دستورات مناسب، بیلدوزر از آن فایل تنظیمات را خوانده و با توجه به آن برنامه شما را اجرا می‌کند.

با این ویژگی‌ها، بیلدوزر ابزاری کارآمد و قدرتمند برای توسعه‌دهندگان است که به آن‌ها کمک می‌کند تا به راحتی برنامه‌های قابل اجرا بر روی دستگاه‌های همراه را بسازند و استفاده کنند. [۸]

۲-۴-۲ جیسون آر - پی - سی^{۲۷}

پروتکل ارتباطی جیسون آر - پی - سی یک پروتکل است که برای تبادل اطلاعات بین دو نرم‌افزار مورد استفاده قرار می‌گیرد. این پروتکل از فرمت جیسون برای ارسال و دریافت داده‌ها استفاده می‌کند و به مشتری^{۲۸} امکان می‌دهد تا درخواست‌هایی را به سرور ارسال کند و سپس سرور به این درخواست‌ها پاسخ دهد. در محیط اوپن دابلو - آر - تی، جیسون آر - پی - سی می‌تواند به عنوان یک رابط برنامه‌نویسی برای دسترسی به ویژگی‌ها و خدمات مختلف سیستم مورد استفاده قرار بگیرد. به عنوان مثال، می‌توان از جیسون آر - پی - سی برای ارسال درخواست‌ها به سرور اوپن دابلو - آر - تی جهت مدیریت شبکه (مانند تنظیمات شبکه، دسترسی به لیست دستگاه‌های متصل و...) استفاده کرد. این امکانات به توسعه‌دهندگان اجازه می‌دهد تا از طریق برنامه‌نویسی مستقیم با سیستم اوپن دابلو - آر - تی ارتباط برقرار کنند و اقدامات مورد نیاز را انجام دهند، بدون نیاز به واسطه کاربری گرافیکی. با استفاده از جیسون آر - پی - سی در محیط اوپن دابلو - آر - تی، توسعه‌دهندگان می‌توانند برنامه‌هایی را ایجاد کنند که قابلیت‌های سیستم را به صورت دینامیک استفاده کرده و امکانات گسترده‌ای را ارائه دهند. این امر به توسعه‌دهندگان اجازه می‌دهد تا برنامه‌هایی با قابلیت‌های پیچیده و چندمنظوره را برای سیستم اوپن دابلو - آر - تی توسعه دهند و به کاربران این امکان را بدهند تا به راحتی با سیستم خود تعامل کنند. [۱۲]

۲-۴-۳ آ-د-ب

آ-د-ب یک ابزار کاربردی برای ارتباط با دستگاه‌های اندروید است. با استفاده از آ-د-ب، کاربران می‌توانند به طور مستقیم با دستگاه‌های اندروید خود ارتباط برقرار کنند و دسترسی به انواع اعمال و تنظیمات

²⁵ docker

²⁶ Docker hub

²⁷ JSON-RPC

²⁸ Clie

نوعی داشته باشند. این ابزار به وسیله دستورالعمل‌هایی که از طریق ترمینال یا خط فرمان دستگاه اجرا می‌شوند، کنترل می‌شود. با استفاده از آ-د-ب، می‌توانید فایل‌ها را به و از دستگاه منتقل کنید، برنامه‌ها را نصب و حذف کنید، اطلاعات دستگاه را مدیریت کنید، اسکرین‌شات بگیرید، دسترسی به لاگ‌ها و دیباگ کردن برنامه‌ها داشته باشید و انجام موارد دیگری که به کنترل و مدیریت دستگاه مربوط می‌شود. آ-د-ب یک ابزار بسیار قدرتمند برای توسعه‌دهندگان اندروید است و به عنوان یک ابزار اساسی در توسعه و تست برنامه‌ها استفاده می‌شود.

۲-۵ چرا از دیگر ابزار استفاده نشد؟

قبل از شروع پروژه قرار بود از دیگر ابزارهایی مانند استودیو اندروید استفاده شود؛ اما متأسفانه به یک سری دلایل مانند تحریم‌های بسیار زیاد، نبود ابزار اندرویدی در دست برای وصل به برنامه و سرعت پایین دانلود ابزار و بسته‌های داخل برنامه از جمله دلایلی بود که تصمیم بر استفاده از زبان برنامه‌نویسی پایتون و کتابخانه کیوی شود.

۲-۶ جمع‌بندی

در این فصل باتوجه به توضیحات ارائه شده، سعی کردیم تا با استفاده از به‌روزترین زبان‌ها، کتابخانه‌ها و چهارچوب توسعه‌ها پروسه طراحی برنامه تنظیمات مودم را آغاز کنیم و یک توضیح اجمالی در ارتباط با روابط میان روسازه و زیرسازه و همین‌طور ابزار خروجی گرفتن دادیم و در آخر هم دلایل استفاده از سایر ابزارها را مشخص کردیم.

فصل سوم

بررسی ساختار برنامه

۱-۳ کلیات نما

مانند تمامی صفحات مودم‌ها، برنامه به چند صفحه تقسیم می‌شود که اولین صفحه‌ای که به کاربر نمایش داده می‌شود، صفحه ورود به برنامه است تا بتوانیم توکن موردنظر را بگیریم و بتوانیم درخواست‌ها را ارسال کنیم. سپس در ادامه بخش‌های صفحه خانه، صفحات تغییر رمز، صفحات آمار آزمون سرعت و... نمایش داده می‌شود. هرکدام از این صفحات دارای منطق خاص خودش است و از کتاب خانه‌های مختلف پایتون برای دستیابی به اطلاعات موردنظر استفاده شده است.

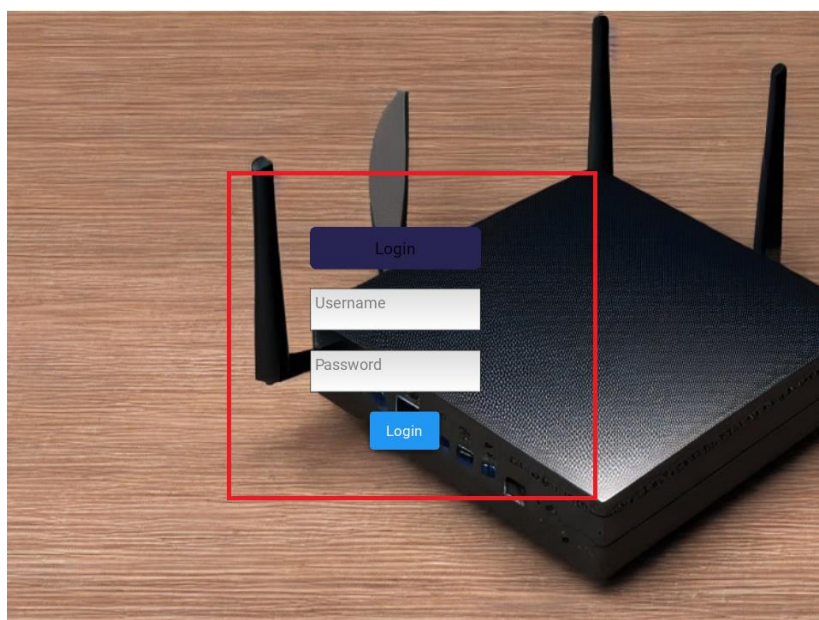
۲-۳ معماری سامانه

در اولین قدم بعد از اجرای برنامه با صفحه ورود مواجه می‌شویم که از ما نام کاربری و کلمه عبور را تقاضا می‌کند.

در اینجا سه حالت رخ می‌دهد:

- نام کاربری و رمز عبور درست است و به مودم با سیستم عامل اوپن دابلو - آر - تی متصل است که وارد می‌شود.
- نام کاربری و یا رمز عبور اشتباه است که با خطا مواجه می‌شود
- دستگاه به مودم با سیستم عامل مناسب وصل نیست و خطا دریافت می‌کند.

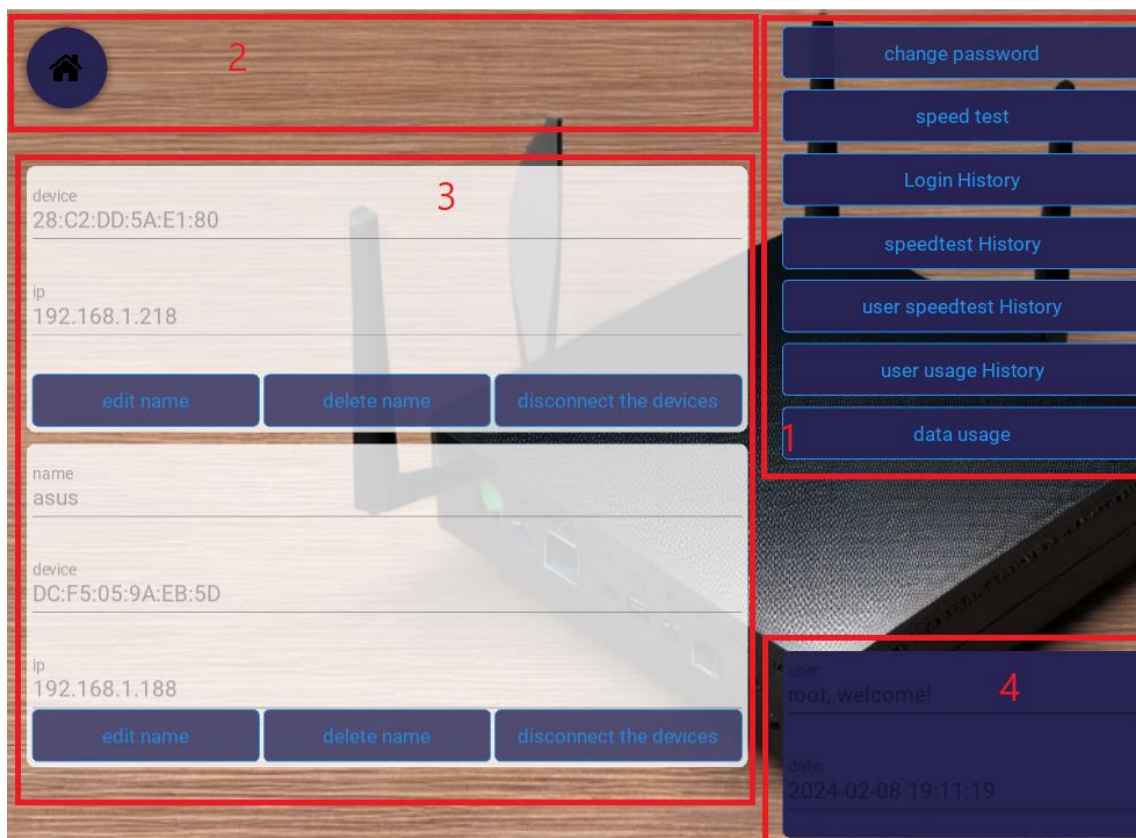
در تصویر زیر صفحه ورود را مشاهده می‌کنید.



شکل ۴- صفحه ورود

بعد از ورود موفق بعد از چند ثانیه اطلاعات پردازش می‌شود و صفحه خانه نمایش داده می‌شود.

در صفحه خانه با چند بخش روبه‌رو می‌شویم. در سمت راست منوها قرار دارد که در شکل پایین با عدد یک مشخص شده است. در ادامه با نوار بالای صفحه که کلیدهای راهنما هستند مواجه می‌شویم. بخش سوم مربوط به دستگاه‌های متصل بی‌سیم به مودم است که اگر نام داشته باشند با نام مشخص می‌شوند؛ همچنین هر دستگاه متصل به مودم دارای سه گزینه متشکل از تغییر نام، حذف نام و قطع کردن دستگاه از مودم می‌باشد. در بخش آخر هم تاریخ روزانه و کاربر وارد به برنامه را می‌توان مشاهده کرد.



شکل ۵- صفحه خانه

منو به ترتیب از بالا شامل قسمت‌های تغییر رمز، تست سرعت کاربر، تاریخچه ورود، تاریخچه تست سرعت سرور، تاریخچه تست سرعت کاربر، جمع زمان اتصال هر دستگاه و مقدار حجم مصرفی می‌شود و در شماره سه به ترتیب از راست به چپ شامل قطع کردن دستگاه از مودم، حذف نام انتخابی و تغییر نام انتخابی قرار دارد.

۳-۳ معماری بانک داده

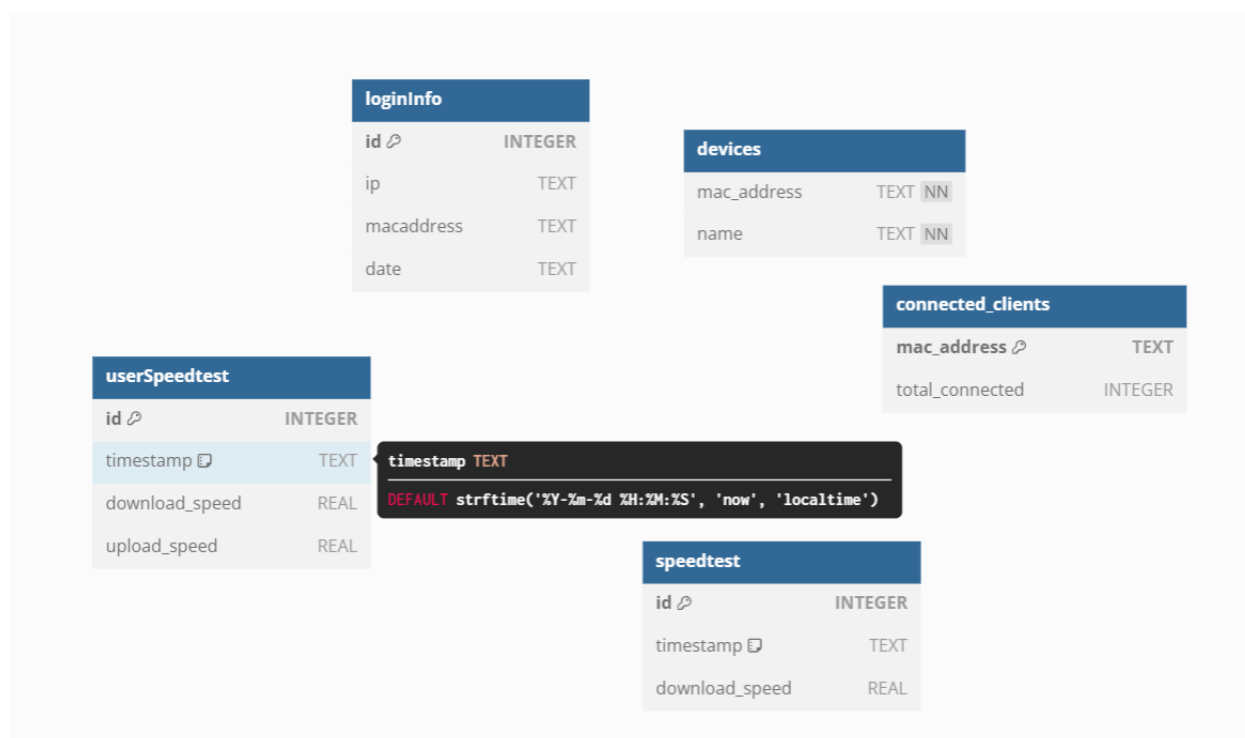
ما چند نوع بانک داده برای ذخیره‌سازی اطلاعات ایجاد کردیم که تمامی آن‌ها به کمک اس - کیو - لایت مدیریت می‌شوند و تماماً در داخل مدیریت برنامه‌های خود مودم ایجاد شده‌اند و هیچ‌گونه بانک داده‌ای بر روس سیستم مشتری و کاربر وجود ندارد

این بانک داده‌ها شامل لیست زیر می‌شوند:

connected_clients	loginInfo	userSpeedtest
devices	speedtest	

شکل ۶- بانک داده‌های موجود

روابط و اطلاعات هر بانک داده به این صورت است:



شکل ۷- روابط بانک داده‌ها

بانک داده دستگاه‌ها مربوط به نگهداری اطلاعات نام آن‌ها است و به کاربر این امکان داده شده است برای هر دستگاه نام دلخواه بگذارد و در زمان دلخواه آن نام را تغییر دهد و یا حذف کند.

بانک داده کاربران متصل مربوط به نگهداری اطلاعات میزان اتصال هر دستگاه به مودم است که بر حسب دقیقه هر ۵ دقیقه یکبار بر اساس الگوریتم مشخص شده در بش که جلوتر به آن می‌پردازیم اطلاعات را نگه می‌دارد.

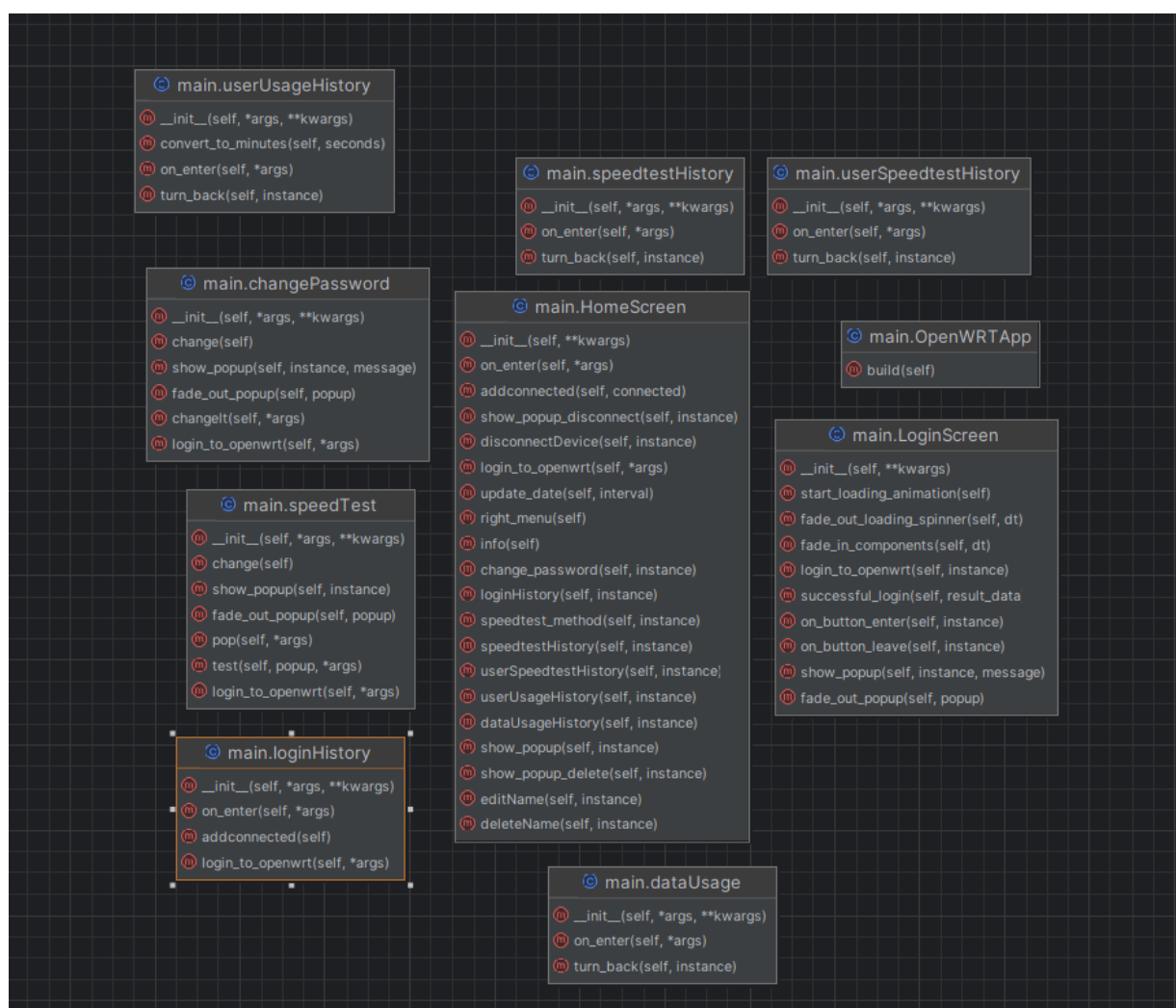
بانک داده اطلاعات ورود مربوط به تاریخچه ورود به برنامه است و در خود دو حالت اطلاعات را ذخیره می‌کند. اگر دستگاه نام داشته باشد نام را به کاربر نشان می‌دهد در غیر این صورت آی - پی کاربر را در خود نگه می‌دارد و در صفحه تاریخچه ورود ده رکورد آخر را به کاربر نشان می‌دهد.

بانک داده آزمون سرعت کاربر اطلاعات آزمون سرعت‌هایی که توسط کاربر در برنامه انجام می‌شود را نگهداری می‌کند و ده رکورد آخر را در نمودار به کاربر نشان می‌دهد.

بانک داده آزمون سرعت اطلاعات آزمون‌های سرعتی که توسط مودم به صورت خودکار هر یک ساعت یکبار انجام می‌شود در خود نگهداری می‌کند و در یک نمودار به کاربر در برنامه نشان داده می‌شود. نیاز به ذکر است دو بانک داده آخر که ذکر کردیم زمان ذخیره شده را از سیستم به کمک دستور ذخیره می‌کند و نیاز به ورودی از سمت کاربر ندارند.

۳-۴ معماری پایتون

ما دو فایل پایتون ایجاد کرده‌ایم یکی برای طراحی روسازه و یکی برای طراحی زیرسازه طراحی روسازه به این صورت است. شکل زیر کلاس‌های کد های روسازه را نمایش می‌دهد.



شکل ۸- طراحی روسازه

همان‌طور که مشخص است هشت صفحه داریم و یک تابع اجرا. این هشت صفحه شامل صفحات آزمون سرعت، تغییر رمز عبور، تاریخچه‌های مختلف و نمودارهای مختلف هست که در ادامه به توضیح هر صفحه به صورت کامل می‌پردازیم.

۳-۵ معماری منوها

در ادامه از بالا گزینه تغییر رمز را انتخاب می‌کنیم و وارد صفحه زیر می‌شویم و باید رمز عبور جدید و تکرار آن را وارد کنیم. اگر با هم برابر باشند آن را تغییر می‌دهد و به صفحه ورود برمی‌گردد در غیر این صورت خطایی برای عدم تطابق رمزها به کاربر نمایش می‌دهد.



شکل ۹- صفحه رمز عبور

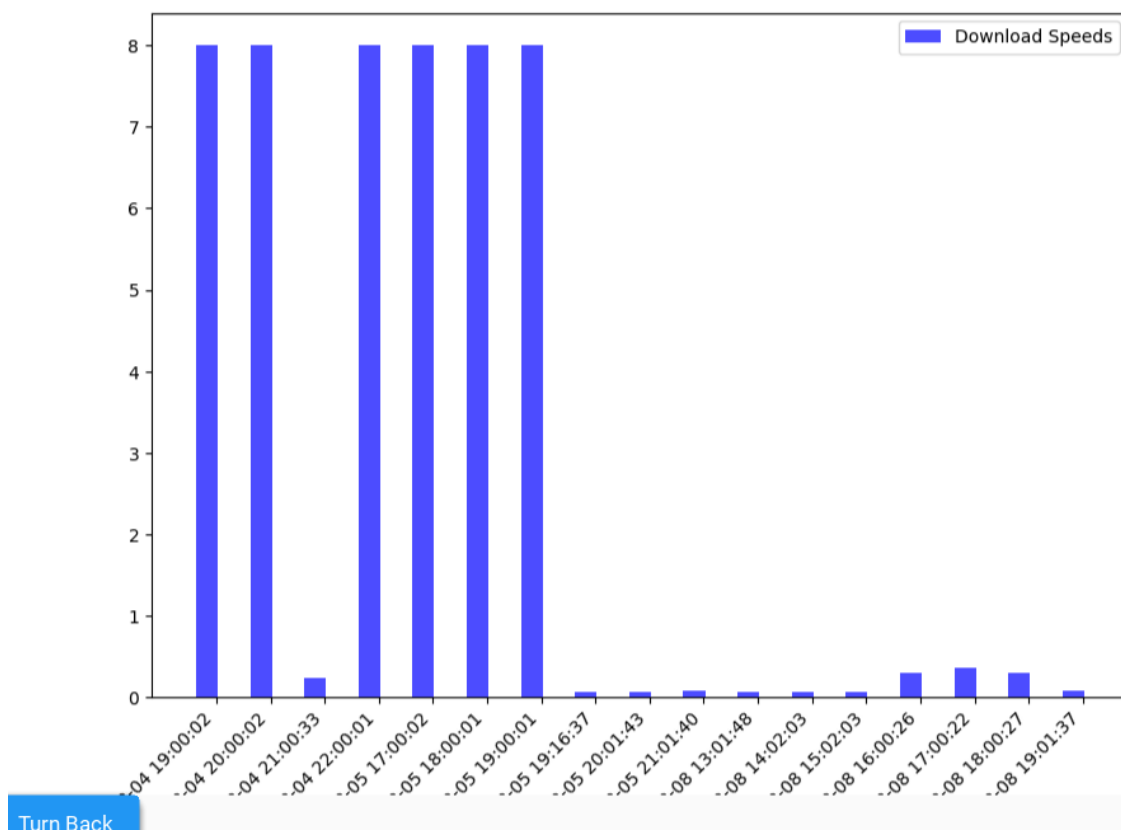
گزینه دوم در صفحه خانه در قسمت منوها مربوط به صفحه تست سرعت کاربر می‌شود. با ورود به این صفحه و انتخاب گزینه تست، تست سرعت گرفته می‌شود و نتایج بارگیری و بارگذاری به کاربر نمایش داده می‌شود و در داخل بانک داده مشخص ذخیره می‌شود. تصویر پایین نتیجه یک آزمون سرعت را نمایش می‌دهد.



شکل ۱۰- صفحه آزمون سرعت

بخش سوم مربوط به بخش تاریخچه ورودها است. بانک داده با هر ورود آپدیت می‌شود و در این بخش به کاربر نمایش داده می‌شود و اگر دستگاهی که به سامانه ورود می‌کند نام داشته باشد، نام آن را به جای آی - پی نمایش می‌دهد. نیاز به ذکر است همیشه ده ردیف آخر بانک داده نمایش داده می‌شود و به دلیل بهینه نگه داشتن سامانه رکوردهای قدیمی پاک می‌شود.

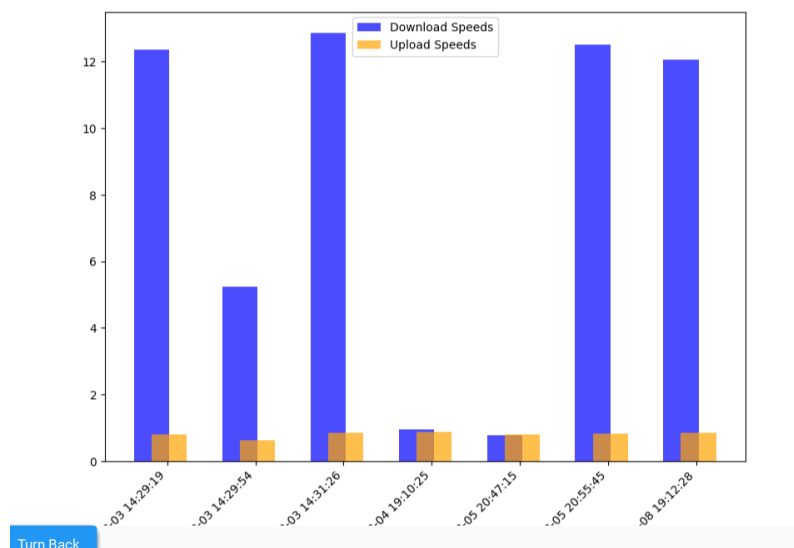
می‌رسیم به قسمت نمودارها که اگر نمودار اول را از منو که مربوط به نمودار آزمون‌های سرعت مودم است انتخاب کنیم با عکس پایین روبه‌رو می‌شویم. نمودار عمودی مربوط به سرعت‌ها بر حسب مگابیت بر ثانیه است و نمودار افقی مربوط به زمان‌ها است و همان‌طور که مشخص است به صورت مداوم هر یک ساعت یکبار تست سرعت به صورت خودکار انجام شده است و ده رکورد جدید بانک داده به کاربر به صورت نمودار نمایش داده می‌شود. تصویر پایین مربوط به آزمون سرعت توسط سرور، اشاره دارد.



شکل ۱۱- نمودار آزمون سرعت سرور

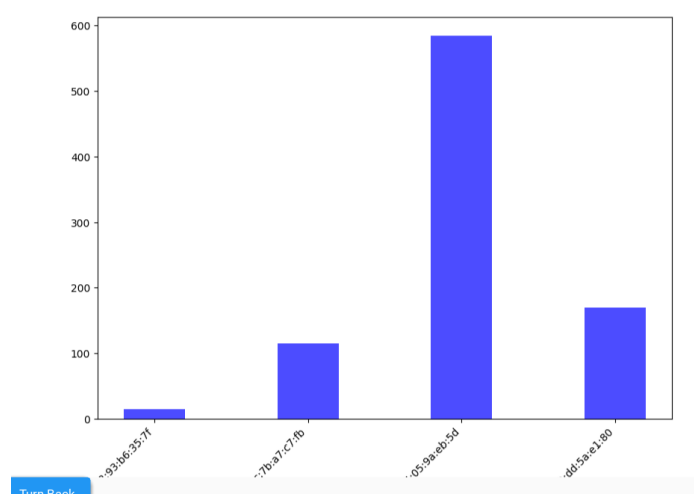
نمودار دوم مربوط به نمودار آزمون‌های سرعتی است که کاربر انجام داده است که با هر انجام در بانک داده مشخص ذخیره شده و ده رکورد آخر به کاربر نمایش داده می‌شود. همان‌طور که مشخص است میله‌های

نارنجی مربوط به سرعت بارگذاری است و میله‌های آبی مربوط به سرعت بارگیری است و نمودار افقی مربوط به زمان‌های آزمون است.



شکل ۱۲- نمودار آزمون سرعت کاربر

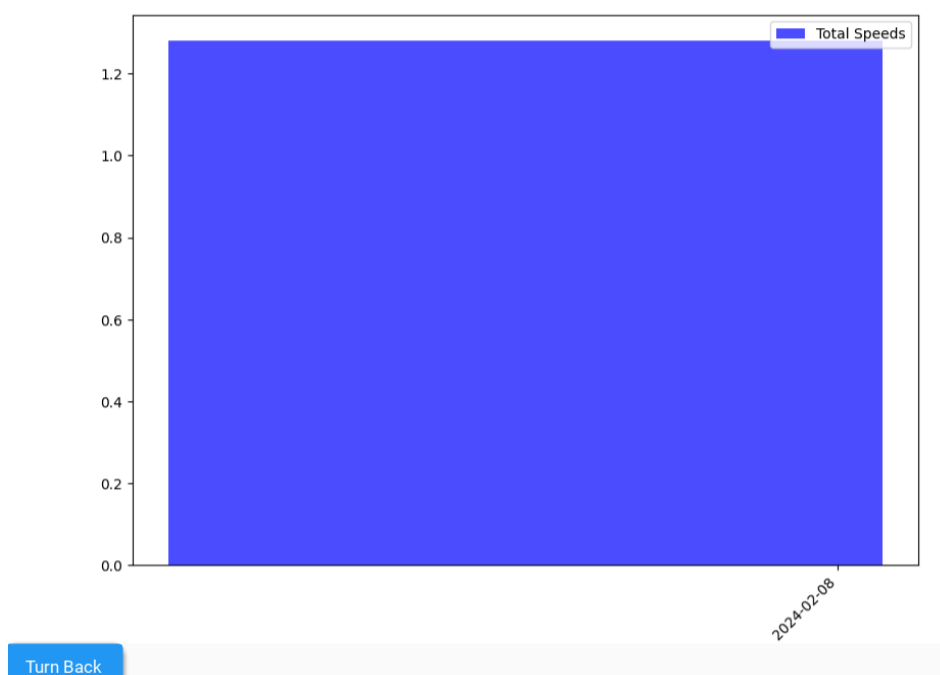
نمودار بعدی مربوط به نمودار زمان اتصالی هر دستگاه به مودم است که برحسب ساعت مشخص می‌کند و نمودار افقی مربوط به آدرس مک^{۲۹} هر دستگاه است. اطلاعات این نمودار هر ۵ دقیقه یکبار به صورت خودکار آپدیت می‌شود و در هنگام چک کردن اتصالات دستگاه‌های مختلف به مودم زمان آن‌ها را آپدیت می‌کند. تصویر زیر مربوط به این نمودار است.



شکل ۱۳- نمودار زمان اتصال هر دستگاه به مودم

²⁹ Mac address

نمودار آخر هم میزان حجم مصرفی دستگاه راه بر حسب گیگ به صورت روزانه نمایش می دهد و این نمودار با هر بار صدازدن مقادیر خود را قبل از نمایش به کار به روزرسانی انجام می دهد و برای هر روز را به صورت مرتب به کاربر نمایش می دهد. نمودار عمودی میزان حجم مصرفی به را حسب گیگ را نمایش داده و نمودار افقی روزها را مشخص کرده است.



شکل ۱۴- نمودار حجم مصرفی

۳-۶ جمع بندی

در این فصل با رویکرد و ساختار برنامه به طور جزئی آشنا شدید، این برنامه می تواند کمک بزرگی برای افراد با دانش کم در ارتباط با ساختار مودم و برنامه های تنظیمات مودم باشد و می تواند بسیاری ایده های نو و جذاب به شرکت های اینترنتی القا کند به طوری که آنها می توانند به فکر اضافه کردن قابلیت های مختلف به برنامه های تنظیمات مودم خود باشند و تحولی در دنیای برنامه نویسی شبکه رخ دهد.

فصل چهارم

نحوه پیاده‌سازی

۴-۱ جزئیات فنی برنامه

در فصول قبل به این اشاره شد که از یک چهارچوب توسعه برای طراحی و ساخت این پروژه استفاده شده است. در این فصل قصد داریم تا روند پیاده‌سازی پروژه در چهارچوب توسعه کیوی و استفاده از ابزار موردنظر مانند بش و بیلدوزر را مرور کنیم. پیاده‌سازی سامانه ما شامل بخش‌های متعددی می‌شود، بخش‌های طراحی روسازه به کمک کیوی است که ممکن است سازه بعضی صفحات (مانند صفحات نمودار) یکسان باشد و بعضی صفحات (مانند صفحات ورود و خانه) ساختار متفاوتی داشته باشند که کامل توضیح داده می‌شوند. بخش بعدی مربوط به زیرسازه است که از کتابخانه‌های مختلف پایتون برای پیاده‌سازی استفاده شده است و ممکن است به بعضی ابزار بیرونی تماس ارسال شود (مانند صندوقچه ابری^{۳۰} ابرآروان). همچنین یک بخش به برنامه‌نویسی به کمک بش در سیستم‌عامل مودم مربوط می‌شود که کاملاً مواردی مانند منطق کد، طریقه اتصال به قسمت کنسولی مودم و خودکارسازی این نوشته‌ها پوشش داده می‌شود. بخش آخر هم مربوط به نحوه کار با ابزار بیلدوزر، نحوه اتصال به دستگاه، دیباگ کردن، خروجی گرفتن از برنامه مربوط می‌شود و آموزش کار با فایل تنظیمات بیلدوزر (همان بیلدوزر دات اسپک) داده می‌شود. در ادامه دو فایل پایتون را مشاهده می‌کنید که یکی برای طراحی روسازه است و دیگری مربوط به زیرسازه است.

در این فایل تمام تنظیمات روسازه ایجاد شده است و از طریق تابع اصلی^{۳۱} زیر پروژه فراخوانی می‌شود.

```
831 if __name__ == "__main__":
832     OpenWRTApp().run()
833
```

شکل ۱۵- تابع اصلی

در ادامه در این قسمت ما تمامی صفحات موردنیاز برنامه را به کمک مدیر صفحات تنظیم می‌کنیم. تمامی این اشارات^{۳۲} مربوط به یک کلاس است و هر کدام از این کلاس‌ها کدهای مربوط به هر صفحه است. تصویر ۱۷ مربوط به این قسمت است.

³⁰ Cloud Storage

³¹ main

³² references

```

1 usage  👤 sepehrmoghiseh
815 class OpenWRTApp(MDApp):
    👤 sepehrmoghiseh
816 ② def build(self):
817     sm = MDScreenManager()
818     sm.add_widget(LoginScreen(name='login'))
819     sm.add_widget(HomeScreen(name='home'))
820     sm.add_widget(changePassword(name='changePass'))
821     sm.add_widget(speedTest(name='speedtest'))
822     sm.add_widget(loginHistory(name='history'))
823     sm.add_widget(speedtestHistory(name='speedtestHistory'))
824     sm.add_widget(userSpeedtestHistory(name='userSpeedtestHistory'))
825     sm.add_widget(userUsageHistory(name='userUsageHistory'))
826     sm.add_widget(dataUsage(name='dataUsage'))
827
828     return sm
829

```

شکل ۱۶- صفحات برنامه

برای هر کدام می‌توان نام دلخواه انتخاب کنیم و در هر قسمت از برنامه می‌توانیم با ایجاد دکمه‌های متعدد صفحات مختلف را صدا بزنیم.

۴-۲ صفحه ورود

این صفحه اولین صفحه‌ای است که بعد از اجرا کردن توسط کاربر به وی نمایش داده می‌شود. از فرد یک نام کاربری و یک رمز عبور برای ورود تقاضا می‌شود که نام کاربری همان نام کاربری تنظیمات مودم است (که معمولاً ریشه ۳۳ هست) با وارد کردن اطلاعات فرد برای درخواست توکن احراز هویت اطلاعات فرد به کمک یک درخواست به آدرس ارسال می‌شود و در صورت بروز هر گونه خطا، پیام مشخص به کاربر نمایش داده می‌شود. در تصاویر پایین نحوه ساخت روسازه برای این قسمت می‌باشد.

```

47 self.username_input = TextInput(hint_text='Username', multiline=False, line_height=1.5, size_hint=(None, None),
48                                width=200, height=50, opacity=0)
49
50 # Password input (initially invisible)
51 self.password_input = TextInput(hint_text='Password', password=True, size_hint=(None, None),
52                                 width=200, opacity=0, height=50, multiline=False)
53
54 # Login button with mouse hover animation (initially invisible)
55 self.login_button = MDRaisedButton(text='Login', size_hint=(None, None), width=200, height=50, opacity=0,
56                                    pos_hint={'x': 0.35, 'y': 0.5})

```

شکل ۱۷- تنظیمات روسازه قسمت اطلاعات ورود

```

1 usage  👤 sepehrmoghiseh
96      def login_to_openwrt(self, instance):
97          username = self.username_input.text
98          password = self.password_input.text
99
100         try:
101             result_data = login(username, password)
102
103             # Check if login was successful
104             if result_data is not None:
105                 self.show_popup(self, message: "yes")
106                 self.successful_login(result_data)
107
108             else:
109                 self.show_popup(self, message: "no")
110
111         except requests.RequestException as e:
112             self.show_popup(self, message: "error")
113

```

شکل ۱۸- طریق کارکرد صفحه ورود

```

3 usages  👤 sepehrmoghiseh
130     def show_popup(self, instance, message):
131         # Create a Popup with some content
132         popup_content = MDBoxLayout(orientation='vertical', spacing=10, padding=10)
133         if message == "yes":
134             popup_content.add_widget(MDLabel(text='Login Successful!'))
135         elif message == "no":
136             popup_content.add_widget(MDLabel(text='Login Unsuccessful!'))
137         else:
138             popup_content.add_widget(MDLabel(text='Error!'))
139         popup = Popup(title='Login Status', content=popup_content, size_hint=(None, None), size=(300, 200))
140         anim_in = Animation(opacity=1, duration=2)
141         anim_in.start(popup)
142         Clock.schedule_once(lambda dt: anim_in.start(popup), timeout=0)
143
144         # Schedule the fade-out animation after two seconds
145         Clock.schedule_once(lambda dt: self.fade_out_popup(popup), timeout=4)
146
147         popup.open()
148

```

شکل ۱۹-اطلاعت مربوط به پیام

در ادامه بعد از بارگیری اطلاعات از مودم و سرویس‌های مختلف اطلاعات به کاربر نمایش داده می‌شود. سرویس زیرسازه برای این قسمت در عکس پایین قابل مشاهده است. در این سرویس اطلاعات به صورت یک بدنه جیسون برای آدرس موردنظر (که به کمک بسته نصب شده بر روی مودم می‌توان از امکانات استفاده کرد و اطلاعات های مختلف مودم را با کمک این بسته گرفت ایجاد شده است). ارسال کرده و توکن احراز هویت

را دریافت می‌کنیم. از این پس برای استفاده از سایر امکانات مودم باید این توکن هم به سرور ارسال شود تا اجازه دسترسی داشته باشیم. همان طور که در عکس زیر مشخص است نتیجه را به صورت عمومی درست می‌کنیم تا بعدها در سرویس‌های مختلف از این توکن بتوان استفاده کرد.

```

76  def login(username, password):
77      rpc_request = {
78          "jsonrpc": "2.0",
79          "id": 1,
80          "method": "login",
81          "params": [username, password]
82      }
83      response = requests.post(url="http://192.168.1.1/cgi-bin/luci/rpc/auth", json=rpc_request)
84      global result_data
85      result_data = response.json()['result']

```

شکل ۲۰- سرویس صفحه ورود

۴-۲ صفحه خانه

این صفحه، اصلی‌ترین صفحه برنامه است و تمامی امکانات در این صفحه جای‌گذاری شده‌اند. این صفحه دارای یک منو در سمت راست یک جعبه دقیقاً در زیر این منو است که اطلاعاتی مانند تاریخ روز و کاربر وارد شده به برنامه را نمایش می‌دهد و به صورت زنده زمان به‌روزرسانی می‌شود. در سمت چپ یک لیست از دستگاه‌های متصل به مودم آورده شده است که اگر هر کدام نام مشخص داشته باشند، آن‌ها هم نمایش داده می‌شود. همچنین کاربر می‌تواند با انتخاب دکمه‌های ایجاد شده برای هر دستگاه، برای آن‌ها نام بگذارد، نام را تغییر و عوض کند و همچنین دستگاه را از مودم قطع کند. در آخر هم از منوی سمت راست می‌تواند وارد قسمت‌های مختلف برنامه شود و از ویژگی‌های مختلف استفاده کند. در شکل زیر پیاده‌سازی این قسمت را مشاهده می‌کنید.

```

211         self.text_input1 = MDTextField(
212             hint_text="device",
213             text=key,
214             readonly=True,
215             multiline=False
216         )
217
218         self.text_input2 = MDTextField(
219             hint_text="ip",
220             text=connected[key]['ipv4'],
221             readonly=True,
222             multiline=False
223         )
224         if findName(key) != None:
225             self.text_input3 = MDTextField(
226                 hint_text="name",
227                 text=findName(key),
228                 readonly=True,
229                 multiline=False
230             )

```

شکل ۲۱- تنظیمات لیست دستگاه های متصل

```

233         self.optionBox = MDBoxLayout(orientation='horizontal', size_hint=(1, 0.3), spacing=4)
234         self.edit = MDRectangleFlatButton(text="edit name",
235                                           md_bg_color=(0.154, 0.14, 0.33), size_hint=(3, None))
236         self.delete = MDRectangleFlatButton(text="delete name",
237                                             md_bg_color=(0.154, 0.14, 0.33), size_hint=(3, None))
238         self.disconnect = MDRectangleFlatButton(text="disconnect the devices",
239                                                  md_bg_color=(0.154, 0.14, 0.33), size_hint=(3, None))
240         self.optionBox.add_widget(self.edit)
241         self.optionBox.add_widget(self.delete)
242         self.optionBox.add_widget(self.disconnect)
243
244         self.card1.add_widget(self.text_input1)
245         self.card1.add_widget(self.text_input2)
246         self.card1.add_widget(self.optionBox)
247         self.edit.bind(on_press=self.show_popup)
248         self.delete.bind(on_press=self.show_popup_delete)
249         self.disconnect.bind(on_press=self.show_popup_disconnect)
250
251         # Add the MDCard to the BoxLayout
252         self.box_layout.add_widget(self.card1)

```

شکل ۲۲- ادامه کد لیست دستگاه های متصل

همان طور که از شروط مختلف مشخص است هر کدام از دستگاه های که اسم داشت، یک قسمت برای اسم آنها در نظر گرفته می شود.

در تصویر زیر زیرسازه مربوط به این لیست را مشاهده می کنید که با یک سری شروط خاص اطلاعات خواسته شده را برمی گرداند.

```

124 def what_device_is_connected(result_data):
125     rpc_url = "http://192.168.1.1/cgi-bin/luci/rpc/sys?auth=" + result_data
126     rpc_request = {
127         "method": "net.host_hints"
128     }
129     response = requests.post(rpc_url, json=rpc_request)
130     hosts = response.json()['result']
131     devices = {}
132
133     for key in hosts:
134         if key.lower() in stations():
135             devices[key]=hosts[key]
136         else:
137             continue
138     return devices
139     # return devices

```

شکل ۲۳- اطلاعات دستگاه های متصل

۳-۴ نمودارها

برای نمودارها ما نیاز به ذخیره سازی اطلاعات در بانک داده های موجود در مودم هستیم. برای این بانک داده ها از اس - کیو لایت ۳ استفاده می کنیم و بانک داده های مورد نیاز را بعد از اتصال به مودم با دستورات اس - اس - اچ^{۳۴} ایجاد می کنیم. کوئری در تصویر پایین مشخص است.

```

sqlite3 database.db "CREATE TABLE speedtest (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    download_speed TEXT,
    timestamp DATETIME DEFAULT CURRENT_TIMESTAMP
);"

```

شکل ۲۴- کوئری های مورد استفاده

یک به یک جداول را به همین صورت ایجاد می کنیم. سپس برای هر بانک داده کدهای مورد نظر را می نویسیم تا رفتارهایی مشخص داشته باشند.

برای نمودار تاریخچه سرعت سرور نیاز به ایجاد یک نوشتار بش داریم که هر یک ساعت یک بار ایجاد شود. در تصویر زیر نحوه پیاده سازی این تنظیمات قابل دسترسی است.

³⁴ SSH

```
# Path to SQLite database file
DB_FILE="database.db"
SPEEDTEST_URL="http://speedtest.ftp.otenet.gr/files/test1Mb.db"

# Run speed test
start_time=$(date +%s.%N)
curl -o /dev/null -s "$SPEEDTEST_URL"
end_time=$(date +%s.%N)
echo "Download Speed: $start_time"
echo "Download Speed: $end_time"
# Calculate download speed in bps with two decimal places
download_speed=$(echo "scale=2; 8 / ($end_time - $start_time)" | bc)

# Print speed for debugging
echo "Download Speed: $download_speed bps"

# Check if download_speed is non-empty
if [ -n "$download_speed" ]; then
    # Insert into SQLite database
    sqlite3 "$DB_FILE" <<EOF
    INSERT INTO speedtest (download_speed) VALUES ('$download_speed');
EOF
    echo "Data inserted into SQLite database."
else
    echo "Error: Download speed is empty. Data not inserted into SQLite database."
fi
```

شکل ۲۵- بش مورد استفاده برای تاریخچه سرعت

همان طور که در این عکس مشخص است، یک فایل را به کمک روش های موجود در لینوکس دریافت می کنیم و بر اساس متدهای متعدد سرعت دانلود را پیدا می کنیم.

سپس این بش را با نام speedtest ذخیره کرده و با کمک دستور زیر برای انجام خودکار هر یک ساعت یکبار برنامه ریزی می کنیم.

```
root@OpenWrt:~# crontab -e
```

شکل ۲۶- دستور مورد استفاده

```
* * * * * ../speedtest.sh >> ../logfile.log 2>&1
```

شکل ۲۷- دستور مورد استفاده دوم

سپس تمامی این اطلاعات را به کمک سرویس پردازش کرده و به قسمت نمودار منتقل می کنیم.

```

245     rpc_request = {"method": "exec", "params": ["sqlite3 database.db \"SELECT * FROM speedtest\""]}
246     result = requests.post(rpc_url, json=rpc_request).json()['result']
247     lines = result.splitlines()
248     times = []
249     speeds = []
250     uploads = []
251     for line in lines:
252         first = line.find("|")
253         second = line.find("|", first + 1)
254         third = line.find("|", second + 1)
255         speeds.append(float(line[second + 1:third]))
256         times.append(datetime.strptime(line[first + 1:second], _format="%Y-%m-%d %H:%M:%S"))
257         uploads.append(float(line[third + 1:]))
258     print(uploads)
259     return times, speeds, uploads

```

شکل ۲۸- منطق سرویس آزمون سرعت سرور

در عکس بالا اطلاعات را از دیتابیس خوانده و خطبه خط پردازش کرده و خروجی را آماده می‌کنیم. در قسمت پایین هم این اطلاعات را به کمک کتابخانه متپلات لیب به کاربر نمایش می‌دهیم.

```

720     # Plot the data
721     ax.bar(bar_positions_download, self.speeds, width=0.4, align='center', label='Download Speeds', color='blue',
722           alpha=0.7)
723     ax.bar(bar_positions_download, self.uploads, width=0.4, align='edge', label='Upload Speeds', color='orange',
724           alpha=0.7)
725     ax.set_xticks(range(len(self.times)))
726     ax.set_xticklabels([date.strftime("%Y-%m-%d %H:%M:%S") for date in self.times], rotation=45, ha="right")
727

```

شکل ۲۹- پیاده‌سازی روسازه نمودار

تمامی نمودارها به همین‌گونه ایجاد می‌شوند و یک‌به‌یک به کاربر نمایش داده می‌شوند. لازم به ذکر است بعضی از منطق‌ها دارای نوشتار بش نیستند و اطلاعات را از سیستم مستقیم می‌گیرند؛ مانند نمودار تاریخچه ورود و یا تاریخچه آزمون سرعت کاربر که نیاز به ذخیره خودکار ندارند.

تقریباً تمامی نمودارها به کمک اطلاعات پردازش شده توسط ما ایجاد می‌شوند به جز اطلاعات مقدار حجم مصرفی که به کمک بسته vnstat ایجاد می‌شوند و خود حتی دیتابیس هم در داخل مودم ایجاد می‌کند و می‌توان آن‌ها را فراخوانی کرد.

```

root@OpenWrt:~# vnstat -i phy0-ap0 -d

phy0-ap0 / daily

  day      rx      |      tx      |      total      |      avg. rate
-----+-----+-----+-----+
  02/08/24  819 KiB |    4.81 MiB |    5.61 MiB |    0.54 kbit/s
  02/12/24    0 KiB |         0 KiB |         0 KiB |    0.00 kbit/s
-----+-----+-----+-----+
estimated  --      |         --      |         --      |

```

شکل ۳۰- مقدار حجم مصرفی روزانه به کمک vnstat

سپس این مقادیر به کمک سرویس زیر به کاربر نمایش داده می‌شوند.

```
305     rpc_request = {"method": "exec", "params": ["vnstat -d -i phy0-ap0"]}
306
307     result = requests.post(rpc_url, json=rpc_request).json()['result']
308
```

شکل ۳۱- گرفتن اطلاعات حجم مصرفی

همچنین تمامی مقادیر به مگابایت و کیلوبایت به گیگابایت تبدیل می‌شوند.

```
318     if "MiB" in line[sec+1:third]:
319         stripped_string = line[sec+1:third].strip()
320         space_index = stripped_string.find(' ')
321         float_string = stripped_string[:space_index]
322         float_value = float(float_string)
323         float_value=float_value*(2**-10)
324         data_usages.append(float_value)
325
326     elif "KiB" in line[sec+1:third]:
327         stripped_string = line[sec + 1:third].strip()
328         space_index = stripped_string.find(' ')
329         float_string = stripped_string[:space_index]
330         float_value = float(float_string)
331         float_value=float_value*(2**-20)
332         data_usages.append(float_value)
```

شکل ۳۲- تبدیل‌سازی مقادیر

آرایه تاریخچه ورود با هر بار ورود اطلاعات کاربر را به کمک اتصالات انجام می‌دهد و اطلاعاتی مانند آی - پی از سوکت دریافت می‌شوند و در بانک داده ذخیره می‌شوند.

سپس این مقدار همراه با زمان ورود در جدول ذخیره می‌شود و به کمک لیست در روسازه نمایش داده می‌شوند.

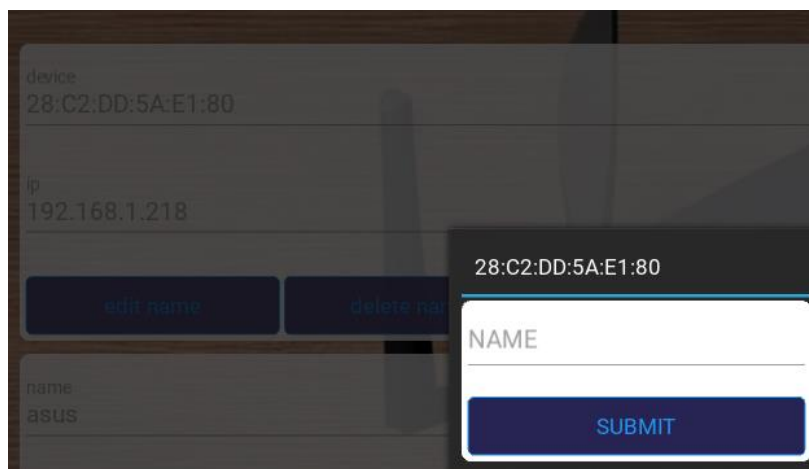
```
169     ip = "\"" + ip_address + "\""
170     date = "\"" + date_formatted + "\""
171     insert = "INSERT INTO loginInfo (ip, date) VALUES (" + ip + "," + date + ");"
172     command = "sqlite3 database.db \"" + insert + "\""
```

شکل ۳۳- ذخیره‌سازی اطلاعات ورود

۴-۴ نام‌گذاری دستگاه‌ها

برای این قسمت ما چند ویژگی موردنیاز از جمله نام‌گذاری، تغییر نام و حذف نام را اضافه کردیم و هر کاربر می‌تواند برای هر دستگاه این امکانات را استفاده کند. برای روسازه آن‌ها کار پیچیده‌ای انجام نشده

است و فقط در یک لایه دکمه‌ها قرار داده شده است و برای هر کدام پیام‌های مناسب به کاربر نمایش داده می‌شود. در تصویر پایین می‌توان یک نام دلخواه برای دستگاه انتخاب کرد.



شکل ۳۴- نام‌گذاری

در عکس زیر همان‌طور که مشخص است با کلیک بر روی دکمه تأیید اطلاعات به زیرسازه منتقل داده می‌شود و لیست به‌روزرسانی می‌شود.

```
1 usage  👤 sepehrmoghiseh
430     def editName(self, instance):
431
432         editName(self.user.text, instance.parent.parent.parent.children[2].text)
433         self.popup1.dismiss()
434
435         self.connected = what_device_is_connected(self.result_data)
436         self.addconnected(self.connected)
437
```

شکل ۳۵- روسازه تغییر نام

در زیر سازه اطلاعات برای آدرس مک مشخص شده از بانک داده گرفته می‌شود و باتوجه‌به اینکه از قبل برای این دستگاه نامی انتخاب شده و یا خیر، کد جلو پیش می‌رود و کوئری مناسب را انتخاب می‌کند. این کوئری می‌تواند قرارداد را برای بار اول و یا به‌روزرسانی مقادیر باشد که یکی از این دو برای سرور ارسال می‌شود و جواب را می‌گیرد و سپس در انتها مقادیر را از رشته حروف استخراج کرده و به روسازه پاسخ می‌دهد. این پیاده‌سازی در عکس زیر مشخص است.

```

192     if result["result"] == '':
193         insert = "INSERT INTO devices (mac_address, name) VALUES (" + mac + "." + name + ".)"
194         command = "sqlite3 database.db \"\" + insert + \"\""
195         rpc_request = {"method": "exec", "params": [command]}
196         requests.post(rpc_url, json=rpc_request).json()
197     else:
198         insert = "UPDATE devices SET name = " + name + " WHERE mac_address = " + mac + "."
199         command = "sqlite3 database.db \"\" + insert + \"\""
200         rpc_request = {"method": "exec", "params": [command]}
201         requests.post(rpc_url, json=rpc_request).json()

```

شکل ۳۶- زیرسازه تغییر نام

۴-۵ جمع‌بندی

در این فصل نحوه پیاده‌سازی برنامه را به صورت جزئی با هم مرور کردیم و توانستیم منطق اجرایی هر بخش را مشخص کنیم و تحلیل قابل قبولی را ارائه دهیم.

فصل پنجم

خروجی گرفتن و چالش‌های پیش‌آمده

۵-۱ مقدمه

در این فصل برنامه کدنویسی شده توسط پایتون را به کمک ابزار بیلدوزر خروجی گرفته و به کمک ابزار ای - دی - بی آن را بر روی تلفن اندرویدی نصب کرده؛ همچنین در ادامه نیز طریقه عیب‌یابی به کمک ابزار ای - دی - بی نیز توضیح داده می‌شود تا بتوان برنامه را در صورت نیاز تغییر داد و وقت کمتری صرف پیش‌بینی کرد و در انتها هم بعضی از چالش‌هایی که در توسعه برنامه در این راه اتفاق افتاد و روش‌های حل این چالش‌ها به اشتراک گذاشته شده است.

۵-۲ استفاده از ابزار بیلدوزر

زمانی که برنامه بر روی پایتون تست شد و تست‌ها با موفقیت اجرا شدند، نیاز به خروجی گرفتن از این برنامه در محیط بیلدوزر و نصب آن بر روی گوشی‌های اندرویدی فرامی‌رسد.

ابتدا در سی - ام - دی^{۳۵} این برنامه را به کمک ابزار پایتون (مانند پیپ^{۳۶}) نصب می‌کنیم و از دستور زیر استفاده می‌کنیم. نیاز به ذکر است برای خروجی اندروید نیاز به سیستم‌عامل لینوکس و یا سیستم توزیعی آن بر روی سیستم‌عامل ویندوز^{۳۷} هستیم چرا که ویندوز از ابزار مناسب اندروید پشتیبانی نمی‌کند. برای نصب سیستم توزیعی لینوکس بر روی ویندوز می‌تواند به قسمت پیوست‌ها مراجعه نمایید.

```
pip3 install --user --upgrade buildozer
```

شکل ۳۷- نصب بیلدوزر

سپس پیش‌نیازهای زیر را به کمک دستورات نشان داده‌شده نصب می‌کنیم. این پیش‌نیازها از ابزاری مانند جی - دی - کی باز^{۳۸}، و سایتون^{۳۹} تشکیل شده‌اند.

```
sudo apt update
sudo apt install -y git zip unzip openjdk-17-jdk python3-pip
pip3 install --user --upgrade Cython==0.29.33 virtualenv # t

# add the following line at the end of your ~/.bashrc file
export PATH=$PATH:~/.local/bin/
```

شکل ۳۸- نصب پیش‌نیازها

³⁵ CMD

³⁶ pip

³⁷ windows

³⁸ Open-JDK

³⁹ Cython

در ادامه در محیط تستی با وارد کردن دستور زیر فایل بیلدوزر - دات - اسپک را ایجاد می‌کنیم. این همان برنامه تنظیمات اجرایی است که در هنگام اجرا این برنامه خوانده شده و باتوجه به تنظیمات وارد شده برنامه خروجی گرفته شده است.

buildozer init

شکل ۳۹- ایجاد فایل تنظیمات

در ادامه قسمت‌های استفاده شده از فایل را مشخص کرده و آن‌ها را توضیح می‌دهیم.

```
# (str) Title of your application
title = My Application3

# (str) Package name
package.name = myapp3

# (str) Package domain (needed for android/ios packaging)
package.domain = org.packi

# (str) Source code where the main.py live
source.dir = .

# (list) Source files to include (let empty to include all the files)
source.include_exts = py,png,jpg,kv,atlas,gif,json

# (list) List of inclusions using pattern matching
source.include_patterns = pics/*,*.*py,garden_matplotlib/*.*py
```

شکل ۴۰- قسمت‌های تغییر داده شده از فایل تنظیمات

بخش اول برای تغییر نام برنامه، نام پکیج بندی و دامنه پکیجی است و می‌توان آن‌ها را به دلخواه تغییر داد. بخش دوم در خط دوم بعد از علامت مساوی باید تمامی نوع فایل‌هایی که در برنامه قرار است استفاده شود، مشخص شود و آن‌ها را لیست کنید. همچنین در خط آخر بعد از علامت مساوی باید الگو برنامه‌هایی که قرار است خوانده شوند را بیاورید. (به طور مثال اگر چند پوشه در قسمت پروژه خود دارید و در هر کدام فایل‌هایی دارید که باید خوانده شوند، باید آدرس آن را به صورت الگویی در اینجا بیاورید تا در هنگام خروجی گرفتن آن‌ها نیز در نظر گرفته شوند).

```
# (list) Application requirements
# comma separated e.g. requirements = sqlite3,kivy
requirements = python3,kivy,kivymd,requests,boto3,datetime,matplotlib,botocore,jmespath,s3transfer
```

شکل ۴۱- قسمت وارد کردن پیش‌نیازها

در این قسمت که می‌توان گفت مهم‌ترین تنظیمات برای خروجی گرفتن است، باید تمامی کتابخانه‌های مورد استفاده و وارد کرده در برنامه را در اینجا بیاورید تا برنامه هنگام خروجی گرفتن این کتابخانه‌ها را بارگیری کند و استفاده کند. اگر تمامی کتابخانه‌هایی که از قبل به صورت پیش فرض در داخل کتابخانه اصلی پایتون نباشد (مانند کتابخانه زمان)، در اینجا قرار داده نشوند، در هنگام اجرای برنامه خروجی گرفته شده، با خطا مواجه می‌شوید و امکان استفاده از برنامه را ندارید. پس برای جلوگیری از این اتفاق تمامی کتابخانه‌های خارجی را قرار دهید. نیاز به ذکر است وارد کردن نام آن‌ها کافیست و ابزار بیلدوزر به طور خودکار آن‌ها را تشخیص می‌دهد.

```
# (str) Icon of the application
icon.filename = pics/iygjuy.png
```

شکل ۴۲- تنظیمات لوگوی برنامه

در این قسمت می‌توان لوگوی مورد نظر برای برنامه انتخاب کرد. منظور از لوگو، همان تصویر برنامه بر روی تلفن‌های همراه است که در صفحه تلفن به نمایش داده می‌شود.

```
# (list) Permissions
# (See https://python-for-android.readthedocs.io/en/latest/buildoptions/#build-options-1 for all the supported syntaxes and properties)
android.permissions = android.permission.INTERNET, (name=android.permission.WRITE_EXTERNAL_STORAGE;maxSdkVersion=18)
```

شکل ۴۳- دسترسی‌های برنامه

در این قسمت لیست دسترسی‌های برنامه قرار دارد که ما فقط نیاز داریم دسترسی به اینترنت را برای برنامه برقرار کنیم چرا که واضحاً دستگاه نیاز به استفاده از اینترنت دارد تا با سرور ارتباط برقرار کند.

نیاز به ذکر است در این فایل هزاران تنظیمات مختلف وجود دارد که شما می‌تواند بسته به دلخواه و کارکرد برنامه خروجی آن‌ها را تنظیم کنید. این تنظیمات شامل تنظیمات برنامه برای خروجی آی - او - اس نیز هست که موضوع پروژه ما نیست. اما برای برنامه ما همین تنظیمات تغییر داده و استفاده می‌شوند و نیاز به استفاده از سایر تنظیمات نیست.

در ادامه در سی - ام - دی دستور زیر را در پوشه اصلی پروژه وارد می‌کنیم تا کار خروجی گرفتن شروع شود.

```
buildozer -v android debug
```

شکل ۴۴- دستور خروجی گرفتن

اگر برای بار اول است این دستور را اجرا می‌کنید ممکن است حتی ساعت‌ها هم خروجی گرفتن طول بکشد چرا که این ابزار ابتدا ابزار کیت توسعه نرم‌افزار^{۴۰} و سپس ابزار توسعه اندروید^{۴۱} و در آخر هم تمامی کتابخانه‌هایی که در قسمت ابزار پیش‌نیاز قرار دادیم، بارگیری می‌کند و تمامی اینها باتوجه به سرعت اینترنت ممکن است ساعت‌ها طول بکشد. بعد از این که تمامی این ابزار بارگیری شدند، شروع به خروجی گرفتن برنامه می‌کند و تمامی کد پایتون را به خروجی اندروید مناسب وصل می‌کند و فایل اجرایی اندروید را در اختیار ما می‌گذارد.

```
# Check application requirements
# Compile platform
# Run ['/usr/bin/python3', '-m', 'pythonforandroid.toolchain', 'create', '--dist_name=myapp', '--
requests,speedtest-cli', '--arch=arm64-v8a', '--arch=armeabi-v7a', '--copy-libs', '--color=alway
/build-arm64-v8a_armeabi-v7a', '--ndk-api=21', '--ignore-setup-py', '--debug']
# Cwd /mnt/d/android/.buildozer/android/platform/python-for-android
[WARNING]: prerequisites.py is experimental and does not support all prerequisites yet.
[WARNING]: Please report any issues to the python-for-android issue tracker.
[WARNING]: prerequisites.py is experimental and does not support all prerequisites yet.
[WARNING]: Please report any issues to the python-for-android issue tracker.
[INFO]: Will compile for the following archs: arm64-v8a, armeabi-v7a
[INFO]: Found Android API target in $ANDROIDAPI: 31
[INFO]: Available Android APIs are (31)
[INFO]: Requested API target 31 is available, continuing.
[INFO]: Found NDK dir in $ANDROIDNDK: /home/sepehr/.buildozer/android/platform/android-ndk-r25
[INFO]: Found NDK version 25b
[INFO]: Getting NDK API version (i.e. minimum supported API) from user argument
[INFO]: ccache is missing, the build will not be optimized in the future.
[DEBUG]: All possible dists: [<Distribution: name myapp with recipes (hostpython3, libffi, open
, setuptools, six, pyjnius, android, kivy, kivymd, urllib3, idna, datetime, certifi, chardet, req
[DEBUG]: Dist matching name and arch: [<Distribution: name myapp with recipes (hostpython3, lib
on3, sdl2, setuptools, six, pyjnius, android, kivy, kivymd, urllib3, idna, datetime, certifi, cha
[DEBUG]: Dist matching ndk_api and recipe: [<Distribution: name myapp with recipes (hostpython3
python3, sdl2, setuptools, six, pyjnius, android, kivy, kivymd, urllib3, idna, datetime, certifi
[INFO]: Of the existing distributions, the following meet the given requirements:
[INFO]: myapp: min API 21, includes recipes (hostpython3, libffi, openssl, sdl2_image, sd
pyjnius, android, kivy, kivymd, urllib3, idna, datetime, certifi, chardet, requests, speedtest-cl
[INFO]: myapp has compatible recipes, using this one
# Build the application #3
# Copy application source from /mnt/d/android
```

شکل ۴۵- تصویری از محیط بیلدوزر

```
pyjnius, android, kivy, kivymd, urllib3, idna, dateti
[INFO]: myapp has compatible recipes, using this o
# Build the application #3
# Copy application source from /mnt/d/android
# Create directory /mnt/d/android/.buildozer/android/
# Copy /mnt/d/android/main.py
# Copy /mnt/d/android/service.py
# Copy /mnt/d/android/test.py
```

شکل ۴۶- هنگام خروجی گرفتن برنامه

بعد از خروجی گرفتن برنامه در پوشه بین^{۴۲} در پوشه اصلی پروژه ذخیره می‌شود.

^{۴۰} SDK

^{۴۱} NDK

^{۴۲} bin

۵-۳ استفاده از ابزار آ-د-ب

در ادامه برای نصب برنامه و عیب‌یابی برنامه از این ابزار استفاده کرد. برای استفاده از این برنامه نیاز داریم تا تلفن همراه با کابل مناسب به کامپیوتر متصل شود و تمامی اجازه‌ها به رایانه برای دسترسی به تلفن داده شود و در انتها تلفن همراه را باید بر روی حالت عیب‌یابی بگذاریم که برای این کار باید در قسمت تنظیمات تلفن همراه، چند بار بر روی گزینه درباره تلفن بزنیم تا در حالت عیب‌یابی قرار بگیرد. با این کار یک سری تنظیمات با عنوان "تنظیمات عیب‌یابی" در تلفن همراه مشاهده می‌شود در این قسمت رفته و گزینه "اجازه عیب‌یابی به کمک حافظه خارجی"^{۴۳} را فعال کنید. با این کار می‌توان برنامه‌ها را از رایانه به گوشی منتقل کنیم و با کمک دستورات مناسب آن را عیب‌یابی کنیم.

نیاز به ذکر است از آنجاکه این برنامه به کمک سیستم توزیعی لینوکس از رایانه ایجاد شده است به کمک انتقال پورت مناسب بتوانیم دستگاه تلفن همراه را در سیستم توزیعی شناسایی کنیم تا امکان نصب و عیب‌یابی ایجاد شود. برای این کار مراحل زیر را انجام می‌دهیم. برای نصب ای - دی - بی می‌توان از لینکی که در قسمت پیوست قرار دادم برای نصب و نحوه استفاده کمک بگیرید.

بر روی سی - ام - دی ویندوز دستور زیر را می‌زنیم تا از اتصال دستگاه به رایانه مطمئن شویم.

```
adb devices_
```

شکل ۴۷- دستور لیست دستگاه‌های متصل

وقتی که دستگاه متصل را در این لیست دیدیم حالا دستور زیر را زده تا بتوانیم تمام اتصالات را به پورت مشخص منتقل کنیم.

```
>adb tcpip 5555_
```

شکل ۴۸- دستور انتقال پورت

حال بر روی سیستم توزیعی لینوکس این دستورات را وارد کرده تا بتوانیم دستگاه را در سیستم توزیعی نیز مشخص کنیم و در ادامه برنامه را نصب کنیم.

برای نصب این برنامه از همان دستورهای نصب برنامه لینوکس کمک می‌گیریم.

```
sudo apt install adb_
```

شکل ۴۹- دستور نصب برنامه بر روی لینوکس

⁴³ Allow USB debugging

سپس در ادامه باید برای اتصال به دستگاه دستور زیر را وارد کنید. توجه کنید باید هر دو رایانه و تلفن به یک اینترنت متصل باشند و آی - پی اختصاص داده شده به تلفن را در این دستور وارد کنید. می‌توانید آن را از تنظیمات تلفن خود پیدا کنید.

```
adb connect 192.168.1.101:5555
```

شکل ۵۰- دستور اتصال به دستگاه

همان‌طور که متوجه شدید این پورت همان پورتی است که پیش‌تر به آن منتقل کردیم. حال تمامی قدم‌ها برای نصب برنامه انجام شده است. ابتدا وارد پوشه‌ای شوید که برنامه خروجی قرارداد و در محیط توزیعی این دستور را برای نصب برنامه انجام دهید. با استفاده از دستور زیر می‌توان این کار را انجام داد.

```
adb -s PL2GAR9850504133 install myapp-0.1-arm64-v8a_armeabi-v7a-debug
```

شکل ۵۱- دستور نصب برنامه

به‌جای عبارت PL... نام اتصالی دستگاه خود که در لیست دستگاه‌های متصل مشاهده می‌کنید را قرار دهید و در آخر هم نام فایل اجرایی که خروجی گرفته شده است را همراه با پسوند *apk*. قرار دهید. بعد از نصب برنامه با پیام موفقیت مواجه می‌شوید.

۵-۳-۱ استفاده از ابزار عیب‌یابی

ممکن است در هنگام اجرای برنامه بر روی تلفن همراه و بررسی قابلیت‌های مختلف با خطاهایی مواجه شوید و به هر دلیلی برنامه به‌درستی عمل نکند. ابزار ای - دی - بی امکان عیب‌یابی را برای ما فراهم کرده است و می‌توانیم به کمک دستور زیر عیب‌های ایجاد شده را پیدا کنیم و آن‌ها را حل کنیم.

```
adb -s PL2GAR9850504133 logcat *:S python:D
```

شکل ۵۲- دستور عیب‌یابی

به‌جای PL... نام دستگاه خود را قرار دهید. با این دستور وارد حالت عیب‌یابی شده و می‌توانیم کد پایتونی که پیش‌تر خروجی گرفتیم را بررسی کنیم و قسمت‌های ایراد دار را اصلاح کنیم. به طور مثال در تصویر پایین با این دستور متوجه شدیم یک سری کتابخانه‌ها پیدا نشده‌اند و دلیل آن این است که در فایل تنظیمات نام این کتابخانه را به‌عنوان ابزار پیش‌نیاز پیش‌تر وارد نکرده‌ایم.

```
ModuleNotFoundError: No module named 'kivy_garden'
Python for android ended.
```

شکل ۵۳-دلیل عیب برنامه

برای این پروژه در هنگام عیب‌یابی با خطاهای متعددی روبه‌رو شدیم که به کمک روش‌هایی که در قسمت بعدی مشاهده می‌کنیم توانستیم آن‌ها را حل کنیم و خروجی صحیح از برنامه بگیریم.

۴-۵ چالش‌های مطرح شده

در هنگام توسعه این برنامه با چالش‌های بسیاری روبه‌رو شدیم که عمده آن‌ها مربوط به مشکلات برنامه بعد از خروجی گرفتن مربوط می‌شد. این مشکلات شامل دلایلی قبیل از عدم نصب پیش‌نیازهای موردنیاز به‌صورت خودکار و یا عدم پشتیبانی از کتابخانه‌های موجود در پایتون در بیلدور می‌شود. یکی از مشکلات اصلی در هنگام توسعه نرم‌افزار مربوط به قسمت آزمون تست سرعت به دست کاربر بود که ابتدا از کتابخانه آماده آزمون سرعت پایتون استفاده شده بود که به کمک سایت آزمون سرعت معروف پشتیبانی می‌شد؛ اما هنگامی که خروجی برنامه به کمک ابزار عیب‌یابی بررسی شد، مشخص شد که این کتابخانه را بیلدور نتوانسته به‌درستی به خروجی اندروید تبدیل کند و خطاهای بسیاری ایجاد کرده است. یکی از این خطاها مربوط به " خطای مسیر فایل "۴۴" مربوط می‌شد. به همین دلیل روشی جایگزین کردیم و تمام قسمت‌های آزمون سرعت به‌صورت دستی زده شد. به این صورت که برای سرعت بارگیری یک فایل به حجم ده مگابایت با ابزار کتابخانه *request* درخواست داده شد و زمان شروع و پایان بارگیری را گرفته و با محاسبه سرعت را پیدا کردیم و برای سرعت بارگذاری از صندوقچه‌های ابری شرکت ابرآوان استفاده شد و با انتقال این فایل به سامانه S3 آن‌ها زمان شروع و پایان را گرفته و محاسبه کردیم.

مشکل دوم مربوط به عدم پشتیبانی و مشکل در کتابخانه‌های مت‌پلات‌لیب مخصوص کیوی می‌شد که با جست‌وجو در اینترنت متوجه شدیم سایر کاربران هم با این مشکل روبه‌رو شدند و برای حل مشکل باید تمام این کتابخانه به‌صورت محلی در پوشه پروژه ریخته می‌شد و تمامی استفاده‌های آن در پروژه به‌صورت محلی انجام می‌شد. برای این کار فایل پروژه از سایت گیت‌هاب^{۴۵} بارگیری شد و در پوشه پروژه قرار داده شد. سپس در فایل تنظیمات بیلدور در قسمت الگوی فایل‌هایی که قرار است خوانده شود، آدرس پوشه این پروژه قرار داده شد و بیلدور در هنگام خروجی گرفتن مسیر این پوشه را جایگزین کرد و در کد هم از آن استفاده شد و دیگر از ابزار خارجی برای ابزار نمودارسازی استفاده نشد. شکل زیر اشاره به خطای پیش آمده دارد.

⁴⁴ Is directory

⁴⁵ Git hub


```
09-14 11:32:56.530 17786 18857 I python : KeyError: 'kivy.garden.matplotlib'
09-14 11:32:56.531 17786 18857 I python : Python for android ended.
```

شکل ۵۴- روش ایجاد شده

در ادامه عکس گفت‌وگوی کاربران گیت‌هاب^{۴۶} برای این مشکل قرار داده شده است تا بیشتر با این قضیه آشنا شوید.

to me it looks like `garden.matplotlib` has bit rot.
 In <https://buildozer.readthedocs.io/en/latest/specifications.html> in the "garden_requirements" section it says:

Please note that if it doesn't work, it might be because of the garden package itself. Refer to the author of the package if h already tested it on your target platform, not us.

Which to me looks like finger pointing, and was not helpful to me.

You can however make a local copy of `garden.matplotlib` and handcraft.... here are my notes from a previous project.

```
# 0) It appers garden.matplotlib has bit rot
# See https://buildozer.readthedocs.io/en/latest/specifications.html
# "garden_requirements"
#
# 1) copy the garden.matplotlib source to a folder
# say <project>/garden_matplotlib
#
# 2) garden.matplotlib expects an old version of matplotlib
# Comment lines 255,256 in garden_matplotlib/backend_kivy.py
# from matplotlib import _png
# from matplotlib import _path
#
# 3) in garden_matplotlib/__init__.py
# change 2 instances import kivy.garden.matplotlib to import garden_matplotlib
#
# 4) in garden_matplotlib/backend_kivyadd.py line 95
# change import kivy.garden.matplotlib to import garden_matplotlib
#
# 5) import into the app as garden_matplotlib, for example:
# from garden_matplotlib.backend_kivyagg import FigureCanvasKivyAgg
#
# 6) no extra buildozer requirements except for the usual matplotlib dependency
#
```

شکل ۵۵- راه حل ارائه شده توسط یکی از کاربران

این کاربر به مشکلات این کتابخانه اشاره کرده و گفته که به‌درستی در بیلدووز تبدیل نمی‌شود. راه حل ارائه داده شده‌ی این کاربر این است که کتابخانه را به‌صورت محلی در پوشه پروژه قرار دهیم و با کامنت کردن

⁴⁶ <https://github.com/kivy/buildozer/issues/۱۲۲۶>:مراجعه به سایت

یک سری از خطها و تغییر آنها به صورت محلی از این کتابخانه استفاده شود. با انجام این کار مشکل برنامه ما نیز حل شده و به درستی کار می کند.

۵-۶ جمع بندی

در این فصل کار با ابزار بیلدوزر و ای - دی - بی را با هم مرور کردیم و توانستیم ابزار عیب یابی را به کار بگیریم، برنامه را نصب کنیم، یک توضیح مختصری از چالش هایی که با آنها روبه رو شدیم و راه حل آنها ارائه دهیم.

فصل ششم

نظرسنجی و تست

۶-۱ مقدمه

در این فصل یک سری نظرسنجی‌هایی که از چند تن دانشجویان امیرکبیر گرفته شده است را همراه با نقد آنان فراهم کردیم و در آخر هم مروری بر کلیات این پروژه داریم و هدف‌های آینده و برنامه‌هایی که برای توسعه قوی‌تر این پروژه و یا پروژه‌های مشابه در ذهن داریم را به نمایش می‌کشیم. امیدوارم که فصل‌های قبل برای خوانندگان این مقاله مفید بوده باشد و چیزی بر دانش آنان افزوده باشد.

۶-۲ نظرسنجی

جدول ۱- نظرسنجی کاربران

سؤالات / کاربران	فرد یکم	فرد دوم	فرد سوم	فرد چهارم
سرعت سامانه تا چه حد قابل قبول بود؟	۶	۵	۷	۷
تا چه حد این برنامه در زندگی روزمره کارایی دارد و یا یک برنامه بی‌استفاده خواهد بود	۵	۸	۷	۷
تا چه حد برنامه ظاهری دارد که افراد را به استفاده تشویق کند؟	۵	۶	۶	۷
تا چه حد به راحتی می‌توان با سامانه کارکرد و تمامی قسمت‌ها مشخص هستند؟	۹	۹	۸	۹
تا چه حد برنامه ویژگی‌های برنامه مورد استفاده قرار می‌گیرد؟	۸	۷	۹	۶
آیا آن را به دیگران پیشنهاد می‌کنید؟	بله	بله	در حال حاضر خیر	بله
میزان رضایت کلی شما	۷	۸	۷	۷
میانگین	۶.۶	۷.۱	۷.۱	۷.۱

قبل از این که نقدهای افراد را اینجا بگذاریم نیاز است جدول را تحلیل کنیم. همان‌طور که مشخص است کمترین امتیاز برای ردیف سرعت است و مشخصاً تمامی افراد رضایت کافی از سرعت نداشتند و باید سرعت برنامه بهبود بخشد. ردیف دوم مربوط به ظاهر برنامه است که نسبت به سایر برنامه‌ها شمایل ساده‌تری

دارد و ممکن است افراد به استفاده از این برنامه ترغیب نشوند و می‌توان در ادامه با تمرکز بیشتر بر روی روسازه شمایل و ارتباط بین قسمت‌های روسازه بهبود یابد. تمای افراد بر سر راحتی کار با برنامه توافق دارند و اعتقاد دارند که به راحتی می‌توان قسمت‌های مختلف برنامه را به کار گرفت و نیاز به استفاده از مستند خاصی ندارد. همچنین سه نفر از چهار نفر این برنامه را به دیگران معرفی می‌کنند و یک نفر معتقد است برای معرفی، برنامه نیاز به توسعه بیشتری دارد.

۶-۲-۱ نقدهای کاربران

افراد هر کدام نظرات متفاوتی داشتند که در اینجا عیناً نقدهای این افراد آورده شده است تا خواننده این مقاله بتواند برای دید بهتر نسبت به برنامه آن‌ها را مطالعه نماید.

نقد فرد اول: "سامانه نیاز به سرعت بیشتری دارد و درعین حال بسیار راحت می‌توان با قسمت‌های مختلف برنامه کارکرد و نیاز به آموزش و توضیحات خاصی ندارد. اما مشکل اصلی این است که این برنامه در حال حاضر ساده‌تر از سایر برنامه‌ها است و ممکن است بسیاری از افراد به دلیل شکل ساده‌ای که دارد به آن توجهی نکنند."

نقد فرد دوم: "درعین حال که این برنامه می‌تواند به خوبی استفاده شود؛ اما ممکن است به دلیل سرعت پایین بسیاری از افراد بعد از مدتی از استفاده از آن خسته شوند و شکل و شمایل ساده آن نیز به سرعت این قضیه کمک می‌کند و با سرعت بیشتری ممکن است افراد برنامه‌های دیگر را به این برنامه ترجیح دهند. اما خب هر برنامه نیاز به استفاده و بازبینی از سمت کاربران دارد تا در شاخه‌های مختلف بهبود یابد پس من این برنامه را به دوستانم پیشنهاد می‌دهم."

نقد فرد سوم: "این برنامه برای نسخه اولیه از سرعت خوب و منطقی ای برخوردار است و برای ورژن اولیه دارای شکل و شمایل خوبی است؛ اما بهتر است این برنامه فعلاً در میان افراد توسعه‌دهنده و آشنا به برنامه‌نویسی استفاده شود و نمی‌توانم آن را به دوستانم که دانشی از برنامه‌نویسی ندارند معرفی کنم چرا که قطعاً بازخورد خوبی نمی‌گیریم."

نقد فرد چهارم: "این سامانه نمایی عالی از یک نسخه اولیه است و تمامی امکانات و حتی عیب‌هایی که یک نسخه اولیه می‌تواند داشته باشد را در خود جای داده است. نیاز است این برنامه زیر دست تعدادی از افراد رود و با آزمون و خطا بسیاری از مسیرهای احتمالی که در پیشروی برنامه قرار دارد مشخص شود و بهترین جهت برای توسعه انتخاب شود. این برنامه شاید در حال حاضر مورد استفاده نباشد؛ اما امکانات جدید و خلاقیتی را مانند آزمون سرعت و نمودارهای متعدد به برنامه‌های این بازار اضافه کرده است و شرکت‌های بزرگ می‌توانند از این گزینه برای نوآوری در برنامه‌های خود استفاده کنند."

این چهار نقد به برنامه موردنظر از جوانب مختلفی به آن نگاه کرده‌اند و نکات مهمی را مطرح کرده‌اند. در اینجا، یک جمع بندی کلی از این چهار نقد ارائه می‌شود:

نیاز به بهبود سرعت و شکل ظاهری: همه چهار نقد اشاره کرده‌اند که برنامه نیاز به بهبود سرعت دارد. نقدهای دوم و سوم به این موضوع اشاره کرده‌اند که سرعت کم و شکل ظاهری ساده ممکن است باعث خستگی کاربران شود و آنها را به برنامه‌های دیگر سوق دهد. نقد اول نیز به شکل ساده برنامه اشاره کرده و احتمال دارد که کاربران به دلیل این سادگی به آن توجه نکنند.

نیاز به بازخورد کاربران: نقد دوم و چهارم تاکید کرده‌اند که نیاز به بازخورد و بازبینی از سمت کاربران وجود دارد تا برنامه در جهت بهبود پیش برود. نقد سوم هم به این موضوع اشاره کرده که برنامه ممکن است فعلاً برای افراد غیرتوانای برنامه‌نویس مورد استفاده قرار نگیرد.

آینده و نقطه قوت‌ها: نقد چهارم به این موضوع اشاره کرده که این برنامه امکانات جدید و خلاقیتی را در اختیار شرکت‌ها و توسعه‌دهندگان قرار می‌دهد و می‌تواند به عنوان یک ابزار نوآوری مورد استفاده قرار گیرد. همچنین، این نقد اشاره کرده که برنامه بهتر است زیر نظر توسعه‌دهندگان تجربه‌ای توسعه یابد و از طریق آزمون و خطا بهترین راه برای پیشروی را انتخاب کند.

در نهایت، می‌توان گفت که این برنامه در حال حاضر نیاز به بهبوداتی دارد از جمله افزایش سرعت، بهبود ظاهری، و دریافت بازخورد کاربران. اما با استفاده از امکانات جدید و بهره‌گیری از نقدها و بازخوردهای مطرح شده، می‌توان به یک نسخه بهتر و جذاب‌تر از این برنامه دست یافت که از آن به عنوان یک ابزار نوآوری و توسعه استفاده شود. همچنین در فصل بعد، برنامه‌هایی که با توجه به این انتقادات برای آینده شکل گرفته شده است را، بیان می‌کنیم.

۶-۳ نتایج تست برنامه

ما یک سری تست واحد و تست‌های ادغامی طراحی کردیم که زیرسازه این برنامه را مورد آزمون قرار دهیم و نتایج را با شما در پایین به اشتراک گذاشت.

```

✓ hm 16 sec 277 ms
  ✓ openWRTTest 16 sec 277 ms
    ✓ test_add_login_info_integration 4 ms
    ✓ test_change_password_service 2 ms
    ✓ test_change_password_service_integration 1 ms
    ✓ test_delete_name_integration 1 ms
    ✓ test_edit_name_integration 1 ms
    ✓ test_find_name_integration 2 ms
    ✓ test_get_user 3 ms
    ✓ test_get_user_integration 0 ms
    ✓ test_login 3 ms
    ✓ test_logread_integration 1 ms
    ✓ test_report_login_integration 2 ms
    ✓ test_run_speed_test_and_InsertIntoUserTest 7 ms
    ✓ test_run_speed_test_integration 16 sec 242 ms
    ✓ test_speedtest_gateway_integration 6 ms
    ✓ test_userSpeedtest_gateway_integration 1 ms
    ✓ test_what_device_is_connected_integration 1 ms

✓ Tests passed: 16 of 16 tests - 16 sec 277 ms
C:\Users\ASUS\AppData\Local\Programs\Python\Python39\pyt
Testing started at 12:20 PM ...
Launching unittests with arguments python -m unittest C:\
C:\Users\ASUS\PycharmProjects\pythonProject10\hm.py:40:
download_speed, upload_speed = run_speed_test()
ResourceWarning: Enable tracemalloc to get the object all
[50.0, 60.0]

Ran 16 tests in 16.287s

OK

Process finished with exit code 0

```

شکل ۵۶- نتایج تست برنامه

همان‌طور که در تصویر بالا مشخص است ما آزمون‌هایی برای تست این برنامه نوشتیم و توابع را به‌صورت واحد و ادغامی مورد تست قرار دادیم. در ادامه چند تا از این تست‌ها را با هم می‌بینیم. همچنین، در تصویر پایین نحوه پیاده سازی تست را مشاهده می‌کنیم.

```

7 @patch('service.boto3.resource')
8 @patch('service.requests.get')
9 @patch('service.requests.post')
10 @patch('service.time.time')
11 def test_run_speed_test_and_insertIntoUserTest(self, mock_time, mock_post, mock_requests_get, mock_boto3_resource):
12     # Mocking the response from S3 bucket
13     mock_bucket = MagicMock()
14     mock_boto3_resource.return_value.Bucket.return_value = mock_bucket
15     # Mocking response content from requests.get
16     mock_response = MagicMock()
17     mock_response.content = b'fake_response_content'
18     mock_requests_get.return_value = mock_response
19
20     # Mocking time measurements
21     mock_time.side_effect = [0, 1, 2, 3] # start_time, end_time (upload), start_time, end_time (download)
22     # Call the function to test
23     download_speed, upload_speed = run_speed_test()
24
25     # Assert the results
26     self.assertEqual(download_speed, second="8.00 Mbps")
27     self.assertEqual(upload_speed, second="8.00 Mbps")
28     mock_bucket.put_object.assert_called_once_with(
29         ACL='private',
30         Body=b'fake_response_content',
31         Key="s.db"
32     )
33
34     # Assert insertIntoUserTest
35     mock_post.assert_called_once()
36

```

شکل ۵۷- آزمون تابع تست سرعت کلاینت

همان‌طور که مشخص است در تست بالا ما داده‌های غیرواقعی و یا به‌اصطلاح ماک^{۴۷} استفاده کردیم و برای این که متغیرها رفتارهایی که در کد واقعی رخ می‌دهد را داشته باشند باید به کمک ابزار برچسب^{۴۸} آن‌ها را از فایل اصلی صدا بزنیم و سپس رفتاری که مورد انتظار ما است را برای نتیجه شرح دهیم و در انتها تابع را صدا بزنیم. در کد بالا ما برای اینکه بتوانیم تست سرعت را به همراه بوتو-۳ تست کنیم باید رفتارهای مشابه را از فایل سرویس برایشان تعریف کنیم و در ادامه یک پاسخ غیرواقعی را برای نتیجه در نظر بگیریم و با مقدار ثانیه‌هایی که در خط ۲۱ در نظر می‌گیریم با توجه به محاسباتی که رخ می‌دهد نتیجه را چک کنیم. به طور مثال در حال حاضر با داده‌های زمانی که داریم جواب هر دو خروجی بارگیری و بارگذاری ۸ مگابیت می‌شود.

```

159 @patch('service.requests.post')
160 def test_add_login_info_integration(self, mock_post):
161     # Mock response from the server
162     mock_response = MagicMock()
163     mock_response.json.return_value = {"result": 11}
164     mock_post.return_value = mock_response
165
166     # Call the function to test
167     addLoginInfo(ip_address='fake_ip_address', date_formatted='fake_formatted_date')
168
169     # Assert that the function made the expected requests to the server
170     mock_post.assert_any_call(
171         "http://192.168.1.1/cgi-bin/luci/rpc/sys?auth=" + result_data,
172         json={"method": "exec", "params": ["sqlite3 database.db \"SELECT COUNT(*) FROM loginInfo\""]}
173     )
174     mock_post.assert_any_call(
175         "http://192.168.1.1/cgi-bin/luci/rpc/sys?auth=" + result_data,
176         json={"method": "exec", "params": [
177             "sqlite3 database.db \"DELETE FROM loginInfo WHERE id = (SELECT MIN(id) FROM loginInfo)\""]}
178     )
179     mock_post.assert_any_call(
180         "http://192.168.1.1/cgi-bin/luci/rpc/sys?auth=" + result_data,
181         json={"method": "exec", "params": [
182             "sqlite3 database.db \"INSERT INTO loginInfo (ip, date) VALUES ('fake_ip_address', 'fake_formatted_date');\""]}
183     )
184

```

شکل ۵۸- تست تابع ثبت اطلاعات ورود در بانک داده

با توجه به تصویر بالا، برای این تست ما می‌خواهیم سرویس ثبت اطلاعات ورود را تست کنیم. اول یک داده ماک ایجاد می‌کنیم و در ادامه از آنجایی که در سرویس اصلی ما شرطی برای بزرگ‌بودن نتیجه (بیشتر از ده) گذاشته بودیم، نتیجه را برابر با ۱۱ می‌گذاریم و در ادامه با صدازدن تابع به همراه اطلاعات نادرست و اطمینان حاصل کردن از انجام تماس‌های موردنیاز با بانک داده در خط‌ها ۱۷۰ تا ۱۷۸، تابع را مورد تست قرار می‌دهیم.

⁴⁷ Mock

⁴⁸ patch


```

265 s.Moghiseh
266 @patch('service.requests.post')
267 def test_speedtest_gateway_integration(self, mock_post):
268     # Mock the response for the first RPC call to get speedtest count
269     mock_post.return_value.json.side_effect = [{'result': '15'}, {
270         'result': '15'}, {
271             'result': '1|2024-02-14 12:00:00|100|50\n2|2024-02-14 12:10:00|120|60\n'}]
272
273     # Call the function to test
274     times, speeds, uploads = speedtestGateway()
275
276     self.assertIsInstance(times, list)
277     self.assertIsInstance(speeds, list)
278     self.assertIsInstance(uploads, list)
279     self.assertEqual(len(times), 2) # Assuming there are 2 records in the response
280     self.assertEqual(len(speeds), 2) # Assuming there are 2 records in the response
281     self.assertEqual(len(uploads), 2) # Assuming there are 2 records in the response
282
283     # Check the values of the first record
284     self.assertEqual(times[0], datetime(year=2024, month=2, day=14, hour=12, minute=0, second=0))
285     self.assertEqual(speeds[0], 100)
286     self.assertEqual(uploads[0], 50)
287
288     # Check the values of the second record
289     self.assertEqual(times[1], datetime(year=2024, month=2, day=14, hour=12, minute=10, second=0))
290     self.assertEqual(speeds[1], 120)
291     self.assertEqual(uploads[1], 60)

```

شکل ۵۹- تست آزمون سرعت سرور

برای این آزمون مشخص شده در تصویر بالا، از آنجایی که اطلاعات خودکار در بانک داده ذخیره می شوند نیاز داریم که داده های دیتابیس را نیز ماک کنیم. در خط ۲۶۸ این اتفاق افتاده است و داده های بانک داده به صورت ماک آماده شدند و در ادامه با استفاده از این داده به عنوان داده اصلی تابع و تست آن در خط های بعدی تعداد خانه های خروجی ها را به همراه نوع آن ها و خود داده تست کردیم تا از صحت داده ها مطمئن شویم.

۴-۶ جمع بندی

در این فصل نتایج نظرسنجی را به همراه نقد و بررسی هر فرد به اشتراک گذاشتیم و تحلیل خوبی از نظرسنجی آماده کردیم. همچنین نتایج تست زیرسازه را به اشتراک گذاشتیم و نحوه تست نویسی برای سه تابع را به صورت جزئی بررسی کردیم.

فصل هفتم

جمع‌بندی، نتیجه‌گیری و هدف‌های آینده

۷-۱ جمع‌بندی اجمالی

ما سعی کردیم پروژه را در ۵ فصل تشریح کنیم و نتایج حاصله را در یک فصل قرار دهیم. در فصل اول با مقدمات و کلیات ابزار مورد استفاده آشنا شدیم و کارهای مشابه را مرور کردیم و دلایلی آوردیم که چرا برنامه ما در یک سری ابعاد بهتر و در یک سری ابعاد ضعیف‌تر عمل خواهد کرد.

در ادامه در فصل دوم تمامی ابزار مورد استفاده را در قالب سه بخش زیرسازه، روسازه و ابزار خروجی گرفتن بررسی کردیم و شرح کاملی و دلایل استفاده از این ابزار را مشخص کردیم تا در آینده اگر فردی خواست پروژه‌ای مشابه را پیاده‌سازی کند بتواند یک دید کلی نسبت به این ابزار داشته باشد و آنی را انتخاب کند که مناسب‌ترین باشد. نیاز به توجه است که این ابزار به دلیل جدید بودن می‌تواند برای هر توسعه‌ای مناسب نباشد و با احتیاط وارد عمل شوید.

در فصل سه جزئیات کارکرد هر قسمت از سامانه را مرور کردیم و تمامی قسمت‌ها را تحلیل و حالت‌های مختلف را مشخص کردیم. همچنین یک توضیح کلی ای در رابطه با حالت‌های خطا داشتیم و این که در زمان رخداد خطا چه اتفاقی می‌افتد. همچنین نمودارهای مختلف و این که هر کدام مربوط به کدام بخش است را نیز تحلیل کردیم و در هر قسمت عکس‌های مشخصی از سامانه ارائه دادیم.

در فصل چهار پیاده‌سازی کد به صورت کامل و اصولی توضیح داده شد و تمامی کدهایی که نیاز به تحلیل بیشتری داشتند، بر روی آنها بیشتر وقت گذاشتیم و نحوه پیاده‌سازی روسازه، زیرسازه و نحوه مرتب کردن اطلاعات نمودارها را توضیح دادیم و توانستیم یک آشنایی کلی ای با این ابزار ارائه دهیم تا در مرحله بعد امکان خروجی گرفتن از برنامه را فراهم سازیم.

فصل پنجم مربوط به خروجی گرفتن از برنامه، نحوه نصب و عیب‌یابی برنامه بود که کاملاً دلایل نحوه استفاده (به طور مثال دلیل اینکه مجبوریم از سیستم توزیعی استفاده کنیم) از محیط‌های مختلف را بررسی کردیم. همچنین پله به پله از نصب تا استفاده از ابزار بیلدور برای خروجی گرفتن را بررسی کردیم و برای آخرین ابزار مورد استفاده، ای - دی - بی را توضیح دادیم و روش عیب‌یابی و نصب را به کمک این ابزار شرح دادیم. در پایان هم چالش‌های پیش آمده و این که چطور آن‌ها را حل کردیم توضیح دادیم و روش‌های مقابله با چالش‌های احتمالی را قرار دادیم.

در این فصل نظرسنجی‌های متعددی را که انجام داده بودیم قرار دادیم و آن‌ها را تحلیل کردیم و توانستیم نظرات اجمالی کاربران مسئول آزمون برنامه را بدست آوریم و آن‌ها را با یک دیگر مقایسه کنیم و در کنارشان نقدهای مستقیم این افراد را نیز قرار دادیم. حال می‌خواهیم در ارتباط با نتیجه ای که از این پروژه بدست آمد صحبت کنیم و در آخر هم هدف‌های پیش رو و کارهایی که در آینده می‌توانیم در ارتباط با توسعه این برنامه و پروژه‌های مشابه داشته باشیم را با شما در میان بگذاریم.

۷-۲ نتیجه‌گیری

در این مقاله، تأکید بر اهمیت ارائه یک اپلیکیشن مخصوص برای تنظیمات اوپن دبلیو - آر - تی به منظور سهولت و دسترسی بهتر به تنظیمات این سیستم عامل شده است. این اپلیکیشن با رابط کاربری کارآمد و قابل فهم، به کاربران امکان مدیریت و تنظیمات دقیق تر را فراهم می کند. باتوجه به مزایای این اپلیکیشن شامل افزایش سرعت و دقت در تنظیمات، کاهش خطاها و اشکالات ممکن، و افزایش کارایی، انتظار می رود که بهبودها بر روی تجربه کاربری و عملکرد شبکه های بر پایه اوپن دبلیو - آر - تی تأثیر مثبتی داشته باشد. امیدواریم که این تحقیق نقطه شروع مؤثری برای توسعه و بهبود سیستم های تنظیمات اوپن دبلیو - آر - تی با استفاده از اپلیکیشن های هوشمندتر باشد. باتوجه به پتانسیل بالقوه ای این اپلیکیشن برای بهبود عملکرد و کارایی شبکه های اوپن دبلیو - آر - تی، پیشنهاد می شود که توجه بیشتری به این حوزه داده شود و منابع لازم برای توسعه و بهبودها آینده ای این اپلیکیشن در نظر گرفته شود. با استفاده از فناوری های نوین و رویکردهای خلاقانه، می توان نقاط ضعف موجود را به فرصت های جدید برای پیشرفت تبدیل کرد و به عملکرد شبکه های اوپن دبلیو - آر - تی افزوده قابل توجهی بخشید.

با انجام این پروژه و نوشتن این مقاله متوجه شدیم که برنامه های کنونی تنظیمات مودم بسیار تکراری و تقریباً تمامی آنها مشابه هم هستند و نیاز به یک نوآوری برای جذب کاربران دارند. همچنین بسیاری از ویژگی هایی متشکل از آزمون سرعت، نمودارها مختلف و نام گذاری در برنامه های کنونی مشاهده نمی شود و توسعه این ویژگی ها و جای گذاری آنها در برنامه به شدت دیده می شود. از سویی توانستیم با ابزاری برای توسعه برنامه که شاید درصد کمی از مردم به فکر استفاده از آن بیافتند، به صورت نو و جدیدی استفاده کنیم و با ترکیبات روش های مختلف کدنویسی و ایجاد نوآوری این ارتباطات را به هم پیوند دهیم و باعث ایجاد یک میکروسرویس شویم که هر قسمت آن به کمک زیرسازهای مختلف مدیریت شوند و نتایج آن ها با هم همگام شوند و بتوانیم ارتباطی میان آنها گسترش دهیم. این کار نه تنها باعث می شود تا این ابزار بیشتر دیده شوند و مردم به استفاده از آنها رو بیاورند، بلکه توسعه دهندگان این ابزار را تشویق به افزودن امکانات و متدهای جدیدتر به ابزار مورد توسعه خود کرده و باعث می شود این افراد به فکر تبلیغات بیشتری در جهت توسعه آگاهی عمومی برای استفاده از ابزار و سایر خدماتشان بیافتند.

معتقدم با توسعه قوی تر این برنامه به صورت گروهی و بیشتر از یک نفر، قطعاً می توان به عنوان یک برنامه جذاب که مردم را به فکر نصب آن می اندازد، طراحی کرد و در بازارهای رقابتی این نوع خدمات برگ برنده ای را در دست بگیرد.

۳-۷ هدف و کارهای آینده

در این قسمت می‌توانیم برای خود هدف‌گذاری کنیم و کارهای احتمالی آینده را در برنامه بلند مدت خود قرار دهیم. در هنگام توسعه این برنامه، ویژگی‌ها و هدف‌هایی که می‌توان در آینده آن‌ها را اجرا کرد به ذهن من آمد که با شما به اشتراک می‌گذارم.

اولین هدفی که برای آینده داریم تمرکز بر افزایش سرعت این سامانه دارد به‌طوری‌که بتوانیم از زیرسازهای استانداردتر و الگوریتم‌های بهینه‌تر استفاده کنیم تا در هنگامی که انتظار پاسخ از سامانه هستیم، زمان زیادی را صرف نکنیم و قدرت پردازش را بهتر کنیم. هدف بعدی نیز بر سرعت دلالت دارد که به بهینه‌سازی روسازه به‌طوری‌که هم از لحاظ زیبایی و چشم‌نواز بودن و هم سرعت پردازش گرافیکی، اشاره دارد. این هدف هم وابسته به توسعه بهتر کتابخانه کیوی و مشتقات و هم آموزش و صرف زمان بیشتر در جهت کار با این کتابخانه هست.

یکی از مهم‌ترین کارهای پیشرو، اضافه کردن ویژگی‌های بیشتری است که این ویژگی‌ها می‌توان شامل اضافه کردن نمودارهای پویاتر با قابلیت‌هایی مانند زوم کردن و هدایت جهتی توسط کاربر باشد. همچنین قابلیت‌هایی مانند انتخاب زمان برای قطع اتصال یک دستگاه به مودم، تنظیمات آی - پی و دی - ان - اس نیز هست چرا که از مهم‌ترین و پرکاربردترین ابزارهای که در تنظیمات مودم ممکن است بسیاری از کاربران استفاده کنند، شامل همین ویژگی‌ها می‌شود.

به‌عنوان آخرین برنامه‌ای که برای آینده خواهیم داشت می‌توان به اضافه کردن ویژگی‌های بیشتری به قسمت آزمون سرعت اشاره کنیم چرا که در حال حاضر این تست سرعت بسیار ساده‌تر از سایر ابزارهایی که این خدمات را ارائه می‌دهد واقع بوده است و می‌توان در شاخه‌های زیادی گسترش یابد.

منابع و مراجع

[1] OpenAI, "ChatGPT," [Computer software] , 2024. [Online]. Available: <https://www.openai.com/chatgpt>

[2] Korsun, Julia. Jul 12, 2023. "The 16 Most Important Pros and Cons of using Python for Web Development", <https://djangostars.com/blog/python-web-development/>

[3] OpenWrt, "OpenWrt Wiki," [Online]. Available: <https://openwrt.org>.

[4] S. Smith, "Optimizing Network Performance with OpenWrt," *IEEE Network*, vol. 30, no. 2, pp. 54-60, Mar./Apr. 2016.

[5] M. F. Hussain et al., "Performance Evaluation of OpenWrt as a Virtual Network Function," in *IEEE Transactions on Network and Service Management*, vol. 14, no. 2, pp. 521-532, Jun. 2017.

[6] R. Zhang, Y. Zhang, and Z. Zhang, "Integration of OpenWrt and SDN for Dynamic Network Management," in *IEEE Access*, vol. 6, pp. 18759-18767, Apr. 2018.

[7] OpenWrt Forum, "OpenWrt Forum - Community Support," [Online]. Available: [Insert OpenWrt Forum URL].

[8] Buildozer Documentation, [Online]. Available: <https://buildozer.readthedocs.io/>

[9] Kivy, "Kivy," [Online]. Available: <https://kivy.org/>

[10] Smith, J., "BackupScript.sh," [Computer software] , 2022.

[11] ActiveState, "What Is Matplotlib in Python? How to Use It for Plotting," [Online]. Available: <https://www.activestate.com/resources/quick-reads/what-is-matplotlib-in-python-how-to-use-it-for-plotting/>

[12] Wikipedia, "JSON-RPC," *Wikipedia*, [Online]. Available: <https://en.wikipedia.org/wiki/JSON-RPC/>

ضمیمه ۱- آموزش نصب اوپن دبلیو - آر - تی بر روی مودم

برای نصب این سیستم عامل بر روی مودم بسته به نوع مودم دارد؛ ولی گام‌های کلی این نصب را در ادامه به اشتراک می‌گذاریم:

۱. وارد سایت سیستم عامل شوید و داخل قسمت دانلودها شوید و به قسمت سخت افزارها همانند شکل پایین بروید.

Downloads

Browse the OpenWrt firmware repository

These links take you to the Downloads directory for the current hardware, grouped by processor type of the devices. OpenWrt software has two distinct branches: a stable **Release** build that is suitable for production use, and a **Development** build that contains an ever-evolving set of enhancements.

Stable Release builds	Development Snapshot builds
The Release builds have had significant testing. Use them for production, or for your home where your family will rely on a functioning router. More...	Get the latest with a Development build. These contain the latest technology, but may not work well, or at all. Be prepared to supply bug reports, etc. More...

Download OpenWrt firmware specific for your device

Go to the **Table of Hardware** or use [firmware selector](#) to locate the latest official release firmware for your device hardware.

- Refer to the [file signing documentation](#) to learn how to verify the integrity of the firmware downloads.
- Refer to the [Quick Start](#) or [User Guide](#), to install the firmware on your device
- GL.iNet, Turris and some others routers already have an OpenWrt based firmware so you may not need to install it manually.

شکل ۶۰-صفحه موردنظر

۲. مدل خود را پیدا کنید و باتوجه به آموزش جلو بروید. در ادامه آموزش برای مودم ASUS RT-N56U قرار داده شده است.

۳. وارد صفحه مودم موردنظر در سایت شوید و از قسمت نصب^{۴۹} فایل موردنظر را دانلود کنید.

ASUS RT-N56U A1



Note: There appears to be a problem with 5 GHz spontaneously turning off after a day or so if channel set to "auto". Reported workaround is to manually set the channel. See the following

- <https://dev.openwrt.org/ticket/22540>
- <https://forum.lede-project.org/t/n56u-wifi-problems/2056>
- <https://forum.lede-project.org/t/n56-no-reactivation-of-5ghz-except-reboot/12244>

Supported Versions

Brand	Model	Version	Current Release	OEM Info	Forum Search	Technical Data
ASUS	RT-N56U	A1	23.05.0	https://www.asus.com/us/networking/rt-n56u/	RT-N56U	View full data

Hardware Highlights

Model	Version	SoC	CPU MHz	Flash MB	RAM MB	WLAN Hardware	WLAN2.4	WLAN5.0	100M ports	Gbit ports	Modem	USB
RT-N56U	A1	Ralink RT3662	500	8	128	Ralink RT3092, Ralink RT3662	big/n	a/n	-	-	-	2x 2.0

Installation

Model	Version	Current Release	Firmware OpenWrt Install	Firmware OpenWrt Upgrade
RT-N56U	A1	23.05.0	https://downloads.openwrt.org/releases/23.05.0/targets/armv8l/rt3662/openwrt-23.05.0-armv8l-rt3662-secure-rt-n56u-arguswrt-viya.upgrade.bin	https://downloads.openwrt.org/releases/23.05.0/targets/armv8l/rt3662/openwrt-23.05.0-armv8l-rt3662-secure-rt-n56u-arguswrt-viya.upgrade.bin

شکل ۶۱- دانلود فایل موردنظر

۴. سپس از سایت رسمی ASUS نرم افزار ASUS Firmware Restoration tool را دانلود کرده

۵. بعد از دانلود مودم را در حالت بازیابی قرار می دهیم:

- مودم را از برق می کشیم
- دکمه بازیابی^{۵۰} مودم را فشار می دهیم
- سپس مودم را به برق می زنیم

۶. در تنظیمات دی-ان-اس های مودم این اطلاعات را قرار می دهیم:

IP address: 192.168.1.10

Subnet mask: 255.255.255.0

۷. سپس فایل موردنظر را با نرم افزار دریافت شده به مودم بارگذاری می کنیم. بعد از بارگذاری موفق مودم خودبه خود تنظیمات را انجام می دهد حالا با مراجعه به صفحه تنظیمات مودم که در حالت اولیه همان آدرس ۱۹۲.۱۶۸.۱.۱ است. مراجعه کنیم با این صفحه روبه رو می شویم.

No password set!
There is no password set on this router. Please configure a root password to protect the web interface.

Status

System

Hostname	OpenWrt
Model	LoraWan Gateway Ethernet
Architecture	ARMv7 Processor rev 5 (v7l)
Firmware Version	OpenWrt 19.07-SNAPSHOT r0-eaf4c87 / LuCI openwrt-19.07 branch git-21.086.32701-7456e2a
Kernel Version	4.14.199
Local Time	2021-04-04 18:20:14
Uptime	0h 4m 16s
Load Average	0.22, 0.44, 0.23

Memory

Total Available	198.88 MB / 245.15 MB (81%)
Used	28.19 MB / 245.15 MB (11%)
Buffered	2.43 MB / 245.15 MB (0%)
Cached	6.90 MB / 245.15 MB (2%)

شکل ۶۲- صفحه تنظیمات مودم

ضمیمه ۲- آموزش اجرای بش اسکریپت

برای اجرای یک اسکریپت بش، ابتدا باید اطمینان حاصل کنید که دسترسی اجرایی برای فایل اسکریپت وجود دارد. سپس با استفاده از ترمینال یا پنجره فرمان، به مسیر فایل اسکریپت بروید. سپس اسکریپت را اجرا کنید با استفاده از دستور زیر:

```
./name-of-file.sh
```

در اینجا، name-of-file.sh نام فایل اسکریپت شما است. توجه داشته باشید که باید فایل اسکریپت دارای دسترسی اجرایی باشد تا بتوانید آن را اجرا کنید. اگر دسترسی اجرایی بر روی فایل تنظیم نشده است، می توانید از دستور `chmod` برای تنظیم دسترسی مناسب استفاده کنید. به عنوان مثال:

```
chmod +x name-of-file.sh
```

بعد از اجرا کردن این دستور، اسکریپت بش شما آماده برای اجرا خواهد بود و می‌توانید آن را با استفاده از دستور `./name-of-file.sh` اجرا کنید.



Amirkabir University of Technology
(Tehran Polytechnic)

Department of Computer Engineering
BSc Thesis

Mobile Application Modem Configuration Project for Users

By
Sepehr Moghiseh

Supervisor
Sajad Shirali-Shahreza

Winter 2024