

به نام خدا



گزارش عمل کرد فاز اول پروژه طراحی و پیاده سازی سامانه تشخیص و مدیریت ورود و خروج

آزمایشگاه سخت افزار
دکتر اجلالی

وحید زهتاب

۹۶۱۱۰۰۶۷

آروین آذرمینا

۹۶۱۰۵۵۴۲

کوروش شریعت

۹۶۱۰۹۷۱۴

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی شریف

بهار ۱۴۰۰

۱ معرفی

هدف اصلی این پروژه پیاده سازی سامانه ای یک پارچه برای تشخیص عبور مرور افراد به کمک سخت افزار طراحی شده (شبیه سازی با کمک raspberry-pi) با قابلیت ضبط تصویر و تشخیص جهت حرکت به کمک سنسورهای حرکتی و همچنین دارای قابلیت اجرای محاسبات مربوط به پردازش تصاویر و تشخیص چهره بر مدار خود می باشد. به بیان دیگر تصور کنید دوربین تحت شبکه ای در اختیار داشته باشید که توانایی تشخیص و مقایسه ی چهره ی افراد را داشته و همچنین به کمک سنسورهای حرکتی خود می تواند با دقت بالایی جهت حرکت افراد را تشخیص دهد. با نصب همچنین دستگاهی بر سردر اتاق های یک موسسه ی بزرگ، بخش های مختلف یک فروشگاه، کلاس های مدارس و مکان های عمومی و یا خصوصی و دیگر کاربردهای مختلفی می تواند برای استفاده از از توانایی های چنین دستگاهی فراهم شود. به عنوان مثال، دستگاه مذکور به راحتی توانایی تشخیص ورود و خروج افراد به اتاقی که بر سردر آن نصب شده باشد را خواهد داشت بنابراین به راحتی می توان از آن به عنوان سامانه ی حضور و غیاب خودکار در مدارس بهره برد.

حال تعدادی از این دستگاه ها را تصور کنید که به کمک شبکه به یک دیگر متصل شده و اطلاعات خود را با یک دیگر به اشتراک گذارند. دستگاه های مختلف در شبکه، گرافی اطلاعاتی تشکیل خواهند داد که محتوای هر بخش آن می تواند به تکمیل مطلب ضبط شده توسط دستگاهی دیگر در شبکه کمک کرده و نگاهی بزرگ تر از مسائل در اختیارمان گذارد. به همین منظور در این پروژه به تحلیل و طراحی و پیاده سازی چنین دستگاه و سامانه ای از اینترنت اشیا^۱ می پردازیم.

در نگاه کلی شبیه سازی چنین سامانه ی سخت افزاری/نرم افزاری می تواند به شبیه سازی دو محصول (نسبتاً) جداگانه ی سامانه ی مدیریت مرکزی دستگاه ها و مجموعه ی دستگاه های دوربین های به اصطلاح هوشمند تقسیم شود که توسط قراردادهایی ارتباطی و برنامه نویسی^۲ به یک دیگر مرتبط می شوند. در ادامه به شرح فعالیت های صورت گرفته برای توسعه و بررسی هر یک از بخش های یاد شده می پردازیم. لازم به ذکر است که تمامی کدهای پیاده سازی شده به صورت مستند شده و آماده ی اجرا از این مخزن گیت هاب قابل دسترسی می باشند.

۲ طراحی و پیاده سازی منطق سامانه ی مدیریت مرکزی

۱.۲ نیازمندی ها و ساختار رابط کاربری

۱.۱.۲ نیازمندی ها

در این سامانه، چندین کاربر داریم، که هر کاربر، دارای چهار سری اطلاعات است. این اطلاعات شامل اطلاعات شخصی آن کاربر، دستگاه های آن کاربر، چهره های آن کاربر و مشاهدات دستگاه های آن کاربر است.

دستگاه های هر کاربر توسط شماره ی مخصوصی به آن کاربر اختصاص داده می شود، و کاربر می تواند چهره های مشاهده توسط آن کاربر را مشاهده نماید.

چهره های هر کاربر شامل دو سری چهره هستند، سری اول شامل چهره هایی است که خود کاربر ثبت کرده است، و سری دوم شامل چهره هایی است که دستگاه های کاربر آن ها را مشاهده کرده اند.

مشاهدات هر کاربر نیز شامل تمامی مشاهداتی است که دستگاه های کاربر انجام داده اند، و چه در صورت

^۱ Internet of Things
^۲ API (Application Programming Interface)

ثبت بودن چهره و چه در صورت جدید بودن چهره، آن چهره در مشاهدات ثبت شده و چهره های جدید در چهره های کاربر اضافه می شوند.

در ادامه به توضیح ساختار رابط کاربری و چگونگی اجرای هر کدام از بخش های بالا در این رابط می پردازیم.

۲.۱.۲ ساختار رابط کاربری

برای این سامانه، رابط کاربری ای در صورت Web Application طراحی نموده ایم، که تمامی نیازمندی های سامانه را در آن شامل است. در ادامه به توضیح صفحات این رابط خواهیم پرداخت.

● صفحه ی ورود و ثبت نام

در این صفحه کاربر امکان ورود و یا ثبت نام در سامانه را دارد. در صورت ورود، کاربر با وارد کردن نام کاربری و رمز ورود خود، وارد صفحه ی اصلی رابط کاربری می شود.

در صورت نیاز به ثبت نام، کاربر بر روز دکمه ی Log On کلیک کرده، و به صفحه ی ثبت نام وارد می شود. در این صفحه با وارد کردن اطلاعات اجباری نام کاربری، رمز ورود حساب کاربری خود را ایجاد کرده، و به صفحه ی قبلی باز می گردد و سپس امکان ورود به سامانه را خواهد داشت.

● صفحه ی اصلی

در این صفحه، کاربر امکان رفتن به صفحات دیگر را دارد، و این صفحه در حکم فهرستی از اختیارات کاربر است.

صفحاتی که کاربر امکان مراجعه به آن ها را در این صفحه دارد، شامل صفحه ی دستگاه ها، صفحه ی چهره ها، صفحه ی مشاهدات و صفحه ی تغییر اطلاعات کاربری است. همچنین کاربر در این صفحه امکان خروج از سامانه را دارد.

● صفحه ی دستگاه ها

در این صفحه، کاربر امکان مشاهده، تغییر اطلاعات، و اضافه کردن دستگاه را دارد.

در این صفحه فهرستی از دستگاه های کاربر مشاهده می شود، که شامل اطلاعاتی در مورد هر دستگاه است. این اطلاعات شامل نام و شماره ی دستگاه و زمان آخرین مشاهده ی دستگاه است.

کاربر برای هر دستگاه در این فهرست، امکان تغییر نام آن را دارد. همچنین کاربر امکان حذف دستگاه را دارد.

کاربر همچنین در این صفحه امکان اضافه کردن دستگاه جدید را دارد، که در صورت کلیک بر دکمه ی اضافه کردن دستگاه، کاربر به صفحه ی دیگری وارد می شود که با گرفتن شماره ی دستگاه و نامی اختیاری برای آن، می تواند دستگاه را به دستگاه های خود اضافه کند.

● صفحه ی چهره ها

در این صفحه، کاربر امکان مشاهده ی فهرستی از چهره های ثبت شده و یا چهره های ثبت نشده ای که توسط دستگاه های خود مشاهده شده اند را مشاهده نماید. همچنین کاربر امکان تغییر اطلاعات هر چهره و یا ایجاد چهره ای جدید دارد.

در فهرست چهره های کاربر، برای هر چهره اطلاعاتی مانند شماره ی آن چهره، نام آن، تصویری از آن، و زمان و دستگاه و نوع آخرین مشاهده ی آن موجود است. کاربر امکان تغییر نام و یا تصویر آن را تغییر دهد، و یا آن را حذف نماید.

در صورت انتخاب دکمه‌ی ایجاد چهره‌ای جدید، کاربر به صفحه‌ای برای اضافه کردن چهره منتقل می‌شود، که در این صفحه با وارد کردن نام و تصویر، می‌تواند چهره‌ی جدید را به چهره‌های خود اضافه نماید. در این بخش در صورتی که تصویر چهره از چهره‌ای باشد که قبلاً در چهره‌های کاربر بوده است، این چهره در چهره‌های کاربر به عنوان چهره‌ای جدید وارد نخواهد شد.

همچنین در زمان اضافه کردن چهره، اگر در تصویر چندین چهره وجود داشته باشد، سامانه تمامی چهره‌های جدید با به چهره‌های کاربر اضافه می‌نماید و برای هر چهره بخشی از تصویر که شامل آن چهره است را اختصاص می‌دهد.

● صفحه‌ی مشاهدات

در این صفحه، کاربر تمامی مشاهداتی که دستگاه‌های ثبت شده‌ی او کرده‌اند را در فهرستی مشاهده می‌نماید. در این فهرست، برای هر مشاهده اطلاعاتی مانند نام چهره، دستگاه آن، زمان مشاهده، نوع مشاهده (ورود یا خروج) وجود دارد. همچنین برای هر مشاهده تصویری از آن مشاهده وجود دارد. در صورت مشاهده‌ی یک دستگاه، در صورتی که چهره‌ای مشاهده شود که در چهره‌های ثبت شده‌ی کاربر نیست، این چهره با نامی کاملاً رندوم به چهره‌های کاربر اضافه می‌شود.

۲.۲ معماری

مهم‌ترین هدف در طراحی بخش نرم‌افزاری پروژه آن است که تا حد ممکن بار محاسباتی بر عهده‌ی سیستم‌های نهفته کاهش یابد تا محصول سخت‌افزاری نهایی نیازمند توان محاسباتی معقولی شود. معماری سامانه به گونه‌ای خواهد بود که تمامی اطلاعات تصویری در بخش نرم‌افزار مرکزی ذخیره شده و هر دوربین تحت شبکه تنها وظیفه‌ی نگهداری تعدادی محدود از مشخصه‌های برداری تصاویر را در حافظه‌ی اصلی خود دارد. بدین طریق سرعت پاسخ‌گویی دستگاه‌های تشخیص چهره بالا رفته و همچنین با کاهش توان محاسباتی مورد نیاز در هر دستگاه، هزینه‌ی ساخت آن نیز می‌تواند کاهش یابد.

مسئله‌ی دیگری که در طراحی بخش نرم‌افزاری سامانه حائز اهمیت است، نحوه و فواصل رد و بدل اطلاعات میان سرور و هر دوربین برای ارسال اطلاعات نسبتاً زیادتر است (برای مثال به روزرسانی لیست بردارهای موجود بر هر دوربین). چرا که ضمن امکان‌کند بودن ارتباط شبکه، سخت‌افزار لازم برای پردازش و انتقال سریع‌تر اطلاعات می‌تواند هزینه‌ی بیشتری به دنبال داشته باشد. به همین منظور و همچنین از جهت مسائل امنیتی، کلیددسترسی هر دوربین پس از مدت محدودی (در این شبیه‌سازی این مقدار معادل یک‌ساعت در نظر گرفته شده است). باطل شده و پس از درخواست برای کلید دسترسی جدید لیست به روز شده‌ای از بردارهای نمایانگر چهره‌ی افراد مشاهده‌شده توسط تمامی دوربین‌های کاربر به هر دوربین ارسال می‌شود. لازم به ذکر است که معماری و نحوه‌ی پیاده‌سازی این بخش از پروژه به صورتی انجام شده است که در صورت علاقه امکان افزایش قابلیت‌هایی چون افزودن رباط تلگرام برای پیام‌رسانی اعلام هشدار ورود و خروج افراد به مکان‌های مختلف وجود دارد.

۱.۲.۲ قرارداد ارتباطی (APIs)

تمامی ارتباطات API توسط درخواست‌های JSON HTTP انجام می‌شود. در ادامه لیستی از درگاه‌های API آمده است و در مورد هر کدام توضیح مختصری داده شده است. قابل ذکر است که تمامی درگاه‌ها با جزئیات فنی‌تر در این آدرس موجود است.

● **درگاه hello** هنگام اتصال دستگاه‌ها به اینترنت، هر دستگاه از این درگاه برای اعلام حضور به سامانه‌ی مدیریت مرکزی با ارسال شماره‌ی خاص آن دستگاه ۳ استفاده می‌نماید. در پاسخ سرور یک توکن دسترسی ۴ موقت را برای آن دستگاه ارسال می‌نماید. هر دستگاه توسط توکن خود می‌تواند با نقاط سامانه ارتباط برقرار کند.

هر توکن به دلایل امنیتی در صورت تغییر IP دستگاه و یا پس از گذشت یک روز از ایجاد آن بی‌اعتبار خواهد شد. دستگاه‌ها پس از بی‌اعتباری توکنشان با درگاه hello باید دوباره اقدام به ساخت توکن جدیدی کنند.

قابل ذکر است که برای هر دستگاه تا کاربری اقدام به اضافه کردن آن نکند، توکنی برای آن ایجاد نمی‌شود.

پس از ایجاد موفق توکن برای دستگاه، آن دستگاه توسط درگاه fetch اقدام به به‌روزرسانی اطلاعات چهره‌های ۵ خود می‌کند.

● **درگاه fetch** توسط این درگاه، دستگاه‌ها می‌توانند آخرین اطلاعات چهره‌ها را دریافت نمایند. برای دور زدن نیاز اتصال مدام به اینترنت و سرور و همچنین سامانه‌های پیچیده به‌روزرسانی، هر دستگاه موظف به‌روزرسانی جدول چهره‌های خود است.

● **درگاه introduce** در صورت مشاهده‌ی چهره‌های ناشناخته توسط دستگاه، با استفاده از این درگاه، دستگاه به سیستم مدیریت مرکزی تصویر و اطلاعات آن را ارسال می‌نماید تا سیستم مدیریت مرکزی آن را با چهره‌های شناخته شده مقایسه نماید و در صورت نیاز چهره‌ی جدیدی ایجاد نماید. در پاسخ به این درگاه نیز سرور id آن چهره، چه جدید و چه شناخته شده، را به دستگاه ارسال می‌نماید.

● **درگاه log** توسط این درگاه دستگاه‌ها در صورت مشاهده‌ی چهره‌ای به سامانه‌ی مدیریت مرکزی اطلاعات آن را ارسال می‌نمایند.

قابل ذکر است که در صورت مشاهده چهره‌ای جدید برای دستگاه، دستگاه ابتدا درگاه introduce را اجرا می‌کند و سپس log.

۳.۲ تکنولوژی

همانطور که مشخص است، نیازمندی سامانه‌ی یادشده و همچنین پیکربندی آن مشابه بسیاری از پروژه‌های شرکتی و وب و یا محصول محوری است. بنابراین همانند این نوع پروژه‌ها بهره‌گیری از ابزارهای موجود برای گسترش نرم‌افزارهای تحت وب می‌تواند کیفیت محصول نهایی را بهبود بخشیده و زمان پیاده‌سازی آن را نیز بکاهد. در نگاه ساده‌انگارانه برای پیاده‌سازی منطق تحت وب این سامانه تنها نیازمند ایجاد چند صفحه‌ی مدیریت و مشاهده‌ی مستندات ضبط شده در پایگاه داده‌های متداولی چون پایگاه داده‌های رابطه‌ای می‌باشد.

نرم‌افزار مورد نظر از سه جهت با کاربران و اطلاعات در ارتباط خواهد بود، یکی از طریق سامانه‌ی گرافیکی که کاربران در آن وضعیت اطلاعات را مشاهده و تنظیمات خود را اعمال می‌کنند. یکی از طریق API قرارداد شده با تعداد (احتمالا) زیادی از دستگاه‌های ضبط و تشخیص حرکت و چهره، و در پایان ارتباط مستقیم یا غیر مستقیم با پایگاه اطلاعاتی.

device_id^۳
access_token^۴
face_embeddings^۵

با در نظر گرفتن این موضوع تصمیم بر آن شد که از framework قدرتمند و کامل django برای پیاده سازی این بخش از سامانه بهره گیریم. این framework علاوه بر تسهیل فراوان طراحی و ارتباط با پایگاه های داده، به کمک ابزارهای مدیریتی و قابل تنظیم و توسعه ی خود چون django-admin و همچنین امکانات فراوان برای طراحی ارتباطات پیچیده تحت شبکه به گزینه ای ایده آل برای توسعه ی محصول مورد نظرمان بدل می شود. به صورتی که تمامی بخش گرافیکی و رابط کاربری مورد نظرمان با توسعه بر سطح django-admin صورت گرفته است.

۴.۲ بررسی عمل کرد سامانه ی پیاده شده

در این قسمت به صورت گام به گام نمونه ی استفاده از سامانه و عمل کرد آن را مرور می کنیم. برای اجرای قطعه کد این بخش مطابق توضیحات شرح شده در [مستندات پیاده سازی سرور](#) می توان عمل کرد. تصور کنید کاربری دستگاه جدیدی از دوربین های هوشمند ما را خریداری کرده باشد، هر دستگاه دارای یک شماره ی شناسایی یکتا می باشد که کاربران با استفاده از آن می توانند مالکیت خود بر دستگاه خویش را در سامانه مطرح کنند.

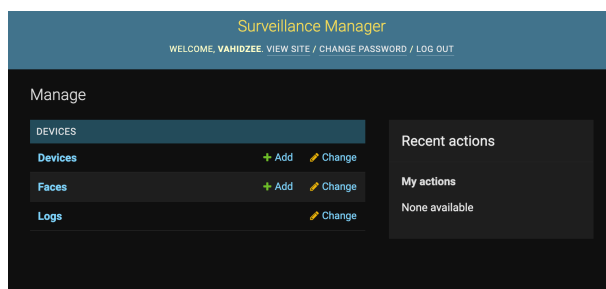
با وصل کردن دوربین هوشمند به اینترنت، دوربین به صورت خودکار از طریق Hello API شروع فعالیت خود را به سامانه اعلام می کند. پس از آن ذکر شماره ی شناسایی دستگاه توسط اولین کاربر در سامانه به معنای مالکیت وی بر دستگاه تازه فعال شده است. (به منظور جلوگیری از تلاقی های ناخواسته میان کاربران، شماره ی شناسایی دستگاه ها به می تواند به صورت مخفی بر بدنه ی دوربین ها نوشته شده باشد) تصاویر ۱، ۱ب، ۲ نمایش گر نحوه ی ثبت نام و ورود به سامانه و صفحه ی اصلی سامانه پس از ورود را نمایش می دهند:

شکل ۱: صفحات عمومی

(ب) صفحه ی ورود

(آ) صفحه ی ثبت نام

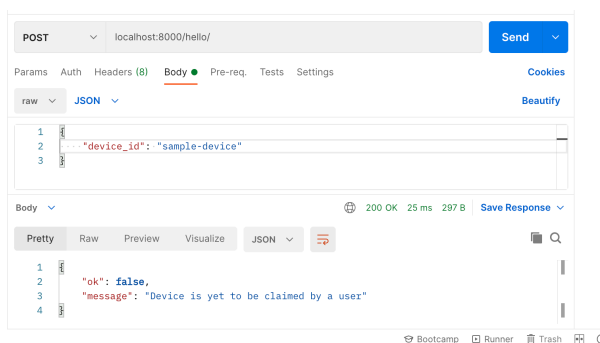
همانطور که توضیح داده شد پیش از ثبت دستگاه جدید خریداری شده نیازمندیم که دستگاه به اینترنت وصل شده بوده و فعالیت خود را به سامانه اعمال کرده باشد، برای شبیه سازی این رویداد به کمک نرم افزار [postman](#) مکالمات صورت گرفته در شبکه را شبیه سازی می کنیم. همانطور که در تصویر ۳آ مشاهده می کنید در صورتی که دستگاه فعالیت خود را اعلام نکرده باشد امکان افزودن آن وجود ندارد. از طرف دیگر تا هنگامی که دستگاه به کاربری تعلق گرفته نشود کلید دسترسی به آن تعلق نخواهد گرفت، تصویر ۳ب نمونه ای از این



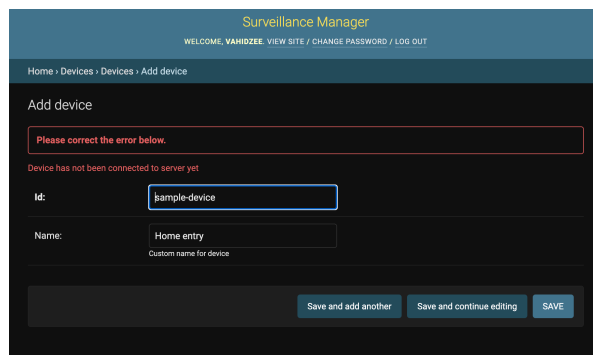
شکل ۲: صفحه اصلی پس از ورود

مسئله می باشد. پس از اعلام فعالیت (در تصویر ۳ب از طریق فهرست Devices می توانیم دستگاه جدید خود را با شناسه اش ثبت کنیم، تصویر ۴آ نمایانگر این مسئله است. پس از اعلام مالکیت کاربر نیز درگاه Hello API کلید دسترسی یک ساعته را در اختیار دستگاه می گذارد.

شکل ۳: مراحل اولیه فعال سازی دستگاه ها

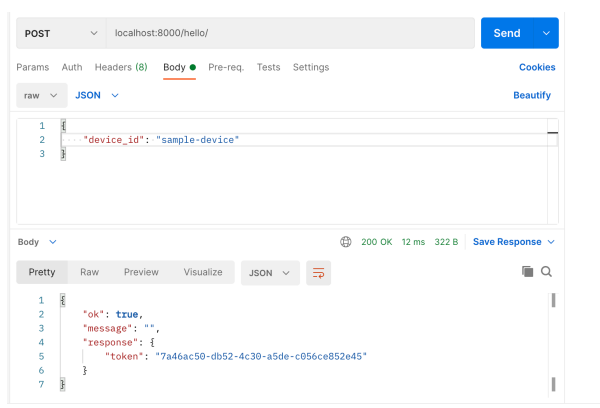


(ب) شبیه سازی اولین اعلام فعالیت دستگاه به سامانه

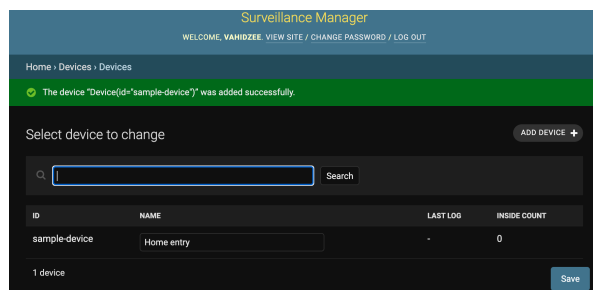


(آ) عدم امکان افزودن دستگاه جدید بدون اعلام فعالیت دستگاه به سامانه

شکل ۴: فعال سازی دستگاه ها و دریافت کلید دسترسی



(ب) دریافت کلید دسترسی توسط دستگاه پس از اعلام مالکیت یک کاربر بر دستگاه

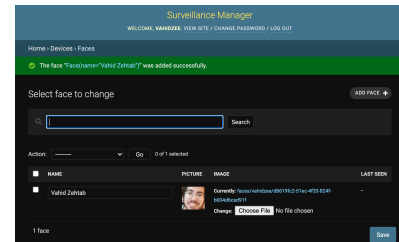
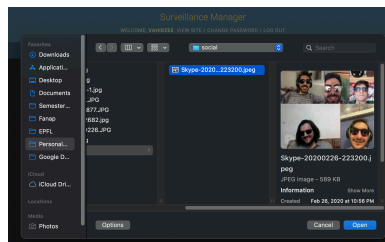
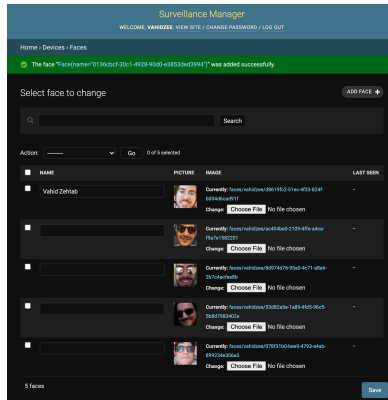


(آ) امکان افزودن دستگاه جدید پس از اعلام فعالیت دستگاه به سامانه

پس از ثبت دستگاه ها کاربران می توانند از طریق فهرست Faces تصاویر جدیدی به سامانه بی افزایند. تصویر ۵آ نمایش نتیجه ی افزودن عکسی تک نفره از این فهرست به سامانه می باشد. همانطور که مشاهده می کنید به صورت خودکار تنها صورت فرد افزوده شده استخراج و نگهداری می شود. حال در صورت اضافه کردن تصویری شامل جمعی از حضار به صورت خودکار صورت ها استخراج و به صورت جداگانه در صورتی

که قبلا در سامانه افزوده نشده باشند، در سامانه ذخیره می شوند. تصویر ۵ ج نتیجه ی افزودن تصویر ۵ ب به سامانه را نمایش می دهد. همانطور که مشاهده می کنید وحید که پیشتر تصویر آن به سامانه در ۵ آ افزوده شده بود به عنوان فرد تکراری در تصویر جدید مجددا حضور دارد اما نسخه ی دیگری از صورت وی به سامانه افزوده نشده است.

شکل ۵: افزودن چهره به سامانه

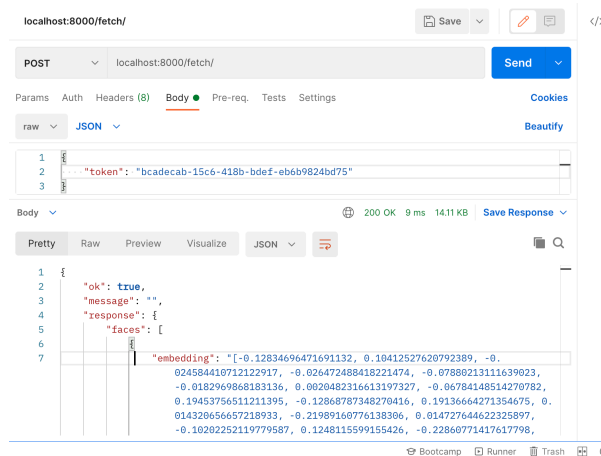


(ج) نتیجه ی افزودن تصویر حاوی چند صورت به سامانه (تشخیص خودکار صورت ها و ذخیره ی آنها یی که در دادگان از پیش وجود نداشته اند)

(ب) نمونه ی تصویر حاوی چند صورت برای افزودن به سامانه

(آ) نتیجه ی افزودن تصویر یک فرد به سامانه و نام گذاری وی

حال که مجموعه ی چهره های داخل سامانه به تعداد قابل قبولی رسیده است می توانیم عمل کرد دریافت لیست نمایش برداری صورت ها و شناساگر آنها را (یعنی Fetch API) را بررسی کنیم. تصویر ۶ نمونه ی عمل کرد این درگاه برای ارسال لیست به روز شده ی اطلاعات چهره های شناخته شده توسط کاربر را نشان می دهد.

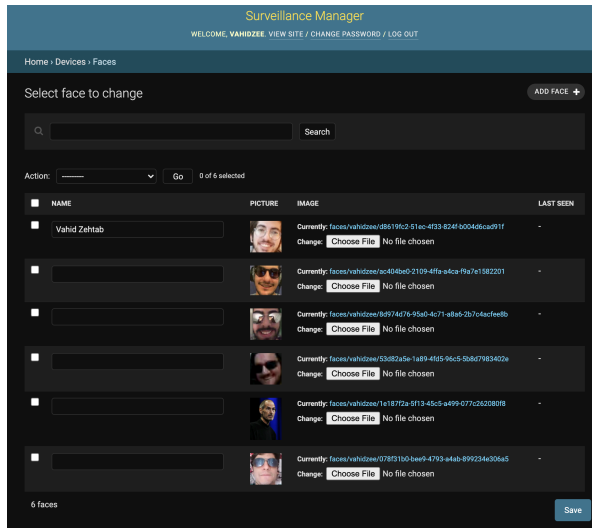


شکل ۶: شبیه سازی واکشی اطلاعات چهره ها از سامانه توسط دستگاه ها

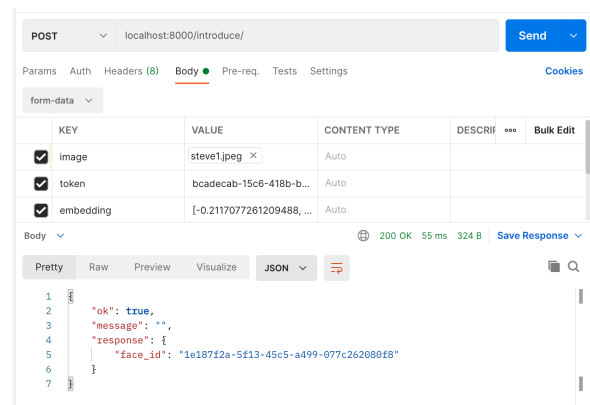
هنگامی که دستگاه با چهره ای جدید روبرو می شود که در حافظه ی خود مشخصات مشابه آن ندارد، تصویر فرد جدید به همراه مشخصات آنرا از طریق درگاه Introduce به سامانه شناسایی می کند. سامانه ابتدا بررسی می کند که آیا این فرد جدید است یا خیر، در صورت وجود مشخصات چهره ی فرد در میان مستندات پیشین، شماره ی شناسایی چهره ی فرد را که از پیش وجود داشته است باز می گرداند. در غیر این صورت پرونده ی چهره ی جدیدی برای وی باز کرده و شماره ی آنرا در پاسخ خود به دستگاه ارسال می کند. تصویر ۷ حاصل معرفی تصویر فردی جدید (تصویر استیو جابز) به سامانه است. لازم به ذکر است که سخت افزار هر دوربین خود توانایی محاسبه ی بردار ویژگی های چهره ها را دارد و برای محاسبه ی این مشخصات به صورت جداگانه

الگوریتم تشخیص ویژگی های چهره بر تصویر مورد بحث اعمال شده است، در حالی که در عمل این کار به صورت خودکار بر دوربین ها صورت خواهد پذیرفت. تصویر ۷ ب هم وضعیت سامانه پس از معرفی تصویر جدید را نمایش می دهد. عمل کرد سامانه حین دریافت تصاویر از پیش شناخته شده نیز مطابق توضیحات می باشد.

شکل ۷: وضعیت سامانه پس از معرفی چهره ی جدید

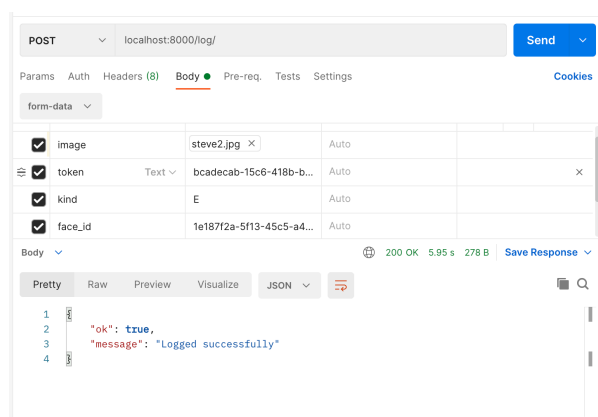


(ب) وضعیت لیست تصاویر کاربر پس از ارسال تصویر جدید (تصویر استیو جابز) به وسیله ی درگاه معرفی چهره ها



(آ) نمونه ی عملکرد درگاه Introduce و دریافت کلید شناسایی مربوط به چهره ی جدید ارسالی

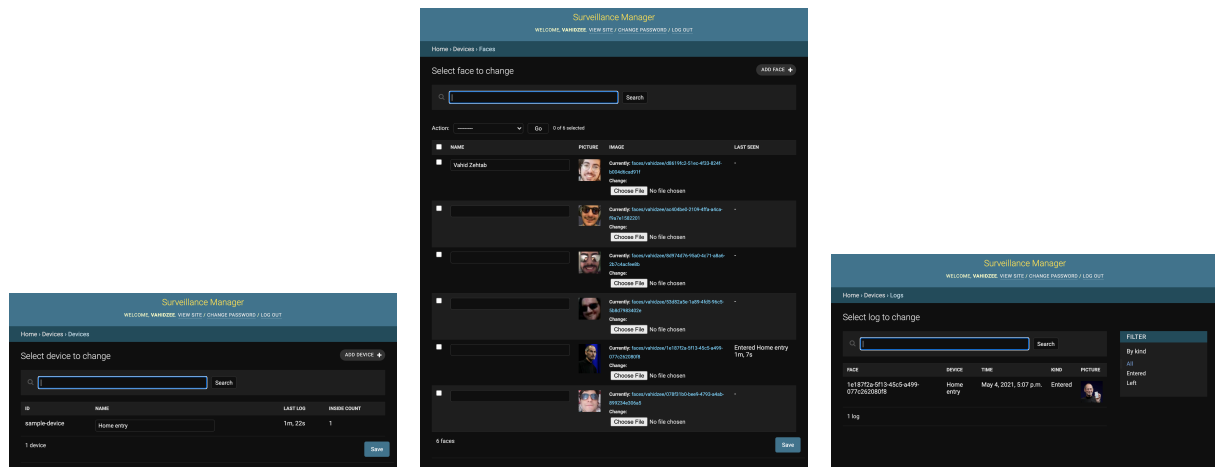
در نهایت عمومی ترین درگاه کماله با سامانه، مربوط به درگاه Log ها می باشد. تصویر ۸ نمونه از ارسال Log اطلاعاتی در باره ی ورود استیو جابز (با تصویری دیگر) به منطقه ی تحت بررسی بوده و تصاویر ۹، ۱۰ و ۱۱ به ترتیب بازتاب اعمال تغییر نام برده بر دادگان سامانه از درگاه Log می باشد. همانطور که مشاهده می کنید آماره های کلی بخش های مختلف بر حسب Log های سامانه همگی تنظیم شده و تغییر می کنند.



شکل ۸: نمونه ی ثبت اطلاعات عبور و مرور از طریق درگاه API Log

هرچند نمونه های بررسی شده پوشای عمل کرد کلی سامانه و ارتباطات آن با دستگاه ها خواهد بود، بررسی و ارزیابی حالات خاص و نحوه ی رسیدگی به خطاهای ممکن در پیاده سازی صورت گرفته خارج از حوصله ی فضای کم این گزارش است با این حال کد پیاده سازی سامانه موجود بوده و بررسی این مسائل خالی از لطف نخواهد بود.

شکل ۹: وضعیت سامانه پس از ثبت یک Log



(آ) وضعیت فهرست Log پس از افزودن نمونه (ب) وضعیت فهرست Face پس از افزودن (ج) وضعیت فهرست Device پس از افزودن
اطلاعات عبور و مرور نمونه اطلاعات عبور و مرور نمونه اطلاعات عبور و مرور

۵.۲ گام بعدی

بخش نرم‌افزاری این سامانه به صورت کامل پیاده‌سازی و ارزیابی شده و در گام بعدی صرفاً باید مستقیماً به منطق شبیه‌سازی شده برای بخش سخت‌افزاری سامانه متصل شود. در نهایت هم در شرایط کنونی امکان انتشار سامانه بر سرور جهانی وجود دارد که می‌تواند صورت پذیرد.

۳ پیاده‌سازی بخش سخت‌افزار

در این بخش سامانه‌ی کنترل‌کننده‌ی هرکدام از واحدهای سخت‌افزاری طراحی و پیاده‌سازی شد. در این فاز فرض می‌شود که لیست افراد مجاز در دستگاه موجود و به‌روز است. در فاز بعدی با یکپارچه‌سازی سخت‌افزار و مدیریت، به‌روزرسانی این لیست و ارتباط کامل میان سامانه‌ای صورت می‌گیرد.

۱.۳ سنسورهای به‌کار رفته

سنسورهای به‌کار رفته در این سامانه همانطور که در پروپوزال اشاره شد، دو سنسور PIR و یک دوربین رزبری پای هست. سنسورها با فاصله‌ای افقی به اندازه ۳۰ سانتی‌متر قرار گرفته تا در محدوده یکدیگر تداخل ایجاد نکنند.

در حلقه اصلی کد سخت‌افزار (main loop) هنگامی که سنسور سمت چپ فعال شد، یک interrupt رخ می‌دهد و دوربین عکس می‌گیرد. از آنجایی که interrupt بر روی سنسور سمت چپ رخ داده، با دانستن وضعیت سنسور راست می‌توان جهت حرکت شخص را تعیین کرد (ورود یا خروج). از آنجایی که خروجی هر سنسور پس از فعال شدن تا چندین ثانیه غیرقابل اطمینان است، ۱۰ ثانیه bounce time برای آن در نظر گرفته شده است. این محدودیت در واقع محدودیت سخت‌افزار کنونی ماست و با استفاده از سنسورهای باکیفیت‌تر قابل رفع است.

۲.۳ تشخیص چهره

تمامی عملیات ها و محاسبات تشخیص چهره بر روی رزبری پای انجام شده تا از تاخیر ارتباط با سرور اجتناب شود. با توجه به قدرت بالای پردازشی این دستگاه، انجام این محاسبات سرباری بر سامانه اضافه نمی کند. در این بخش دو الگوریتم تشخیص چهره آزموده شدند.

۱.۲.۳ الگوریتم Haar Cascade

در این الگوریتم، تصویر از تعداد زیادی فیلتر ساده که قبلاً آموزش دیده اند عبور کرده (cascade شده) و در نهایت محل صورت ها در تصویر مشخص می شود. ابتدا قصد بر استفاده از این الگوریتم چرا که الگوریتمی ساده و سبک است. اما از آنجایی که قابلیت مقایسه صورت ها در این الگوریتم وجود ندارد، از الگوریتم بعدی استفاده شد. در ادامه یک نمونه کد ساده از نحوه کاربرد این الگوریتم در کتابخانه opencv آمده است.

```

1 cap = cv2.VideoCapture(0)
2 casc = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
3 ret, frame = cap.read()
4 if ret:
5     print(int(cap.get(3)))
6     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
7
8     faces = casc.detectMultiScale(
9         gray,
10        scaleFactor=1.1,
11        minNeighbors=5,
12        minSize=(30, 30)
13    )
14    for (x, y, w, h) in faces:
15        cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2) # draw a rectangle around
16                                the detected faces
17    cap.release()

```

یکی دیگر از چالش های این الگوریتم نیز تعداد hyperparameter ها مانند حداقل همسایه هاست که دقت مدل را کنترل می کند.

۲.۲.۳ الگوریتم شبکه عصبی

در این الگوریتم ها در ابتدا مکان صورت ها محاسبه می شود (این مرحله می تواند مانند بخش قبل رخ دهد) سپس هر صورت وارد یک شبکه عصبی شده و به یک بردار ویژگی encode می شود. این الگوریتم بار محاسباتی سنگین تری دارد اما افراد داخل عکس را نیز تشخیص می دهد. در نهایت از پیاده سازی این الگوریتم در کتابخانه پایتون face_recognition استفاده شد که در ریپازیتوری پروژه قابل مشاهده است. در ادامه نیز چند مثال از عملکرد این دو الگوریتم آمده است.

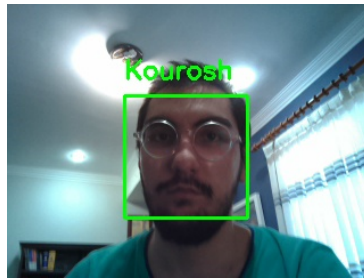
۳.۳ گام بعدی

در گام بعدی بین سخت افزار و سامانه مدیریت ارتباط کامل برقرار خواهد شد. همچنین یک LCD ساده برای اطلاع رسانی افراد در محل تعبیه می شود.

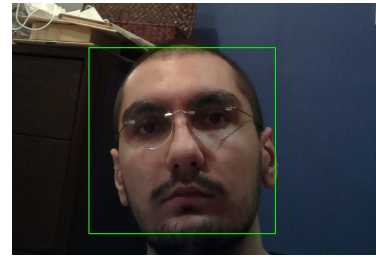
شکل ۱۰: خروجی الگوریتم های مختلف



(ج) مثال ۲ شبکه عصبی



(ب) مثال ۱ شبکه عصبی



Haar Cascade (آ)