دانشکده علوم و فنون نوین

گروه بین رشته ای فناوری (بخش علوم و فناوری شبکه)

گزارش پروژه

احراز هویت غیرحضوری متقاضیان خدمات الکترونیک انتظامی بر مبنای سنجه‌های بیومتریکی

توسط:

هادی ویسی

فرودین ۱۴۰۱

فهرست

فهرست شکل‌ها

فهرست جدول‌ها

خلاصه اجرایی

سامانه احراز هویت غیرحضوری موضوع قرارداد، روی سرورهای ناجا نصب و راه‌اندازی شد و سرویس‌های آن مورد ارزیابی قرار گرفت. این سامانه شامل سه لایه هوش مصنوعی، مدیریت و دمو است که درخواست‌ها توسط لایه دمو دریافت شده و پس از بررسی توسط لایه مدیریت (برای مدیریت درخواست‌ها و کنترل دسترسی)، به لایه هوش مصنوعی برای پردازش اصلی ارسال می‌شوند. لایه پردازش هوش مصنوعی با بهره‌گیری از مدل‌های یادگیری عمیق، فعالیت‌های پردازش تصویر و گفتار را انجام می‌دهد و نتیجه برگردانده می‌شود. برای در اختیار قرار دادن سرویس‌های پایه‌ی سامانه‌های احراز هویت شامل تطبیق چهره و تشخیص زنده بودن، بعد از برگزاری جلسات مشورتی با صاحب‌نظران در فرماندهی نیروی انتظامی و شرکت پژوهش و توسعه ناجی، از بین دو راه‌حل استفاده از API و Gateway که در گزارش مرحله‌ی قبل ارائه شده بود، روش مبتنی بر API انتخاب گردید. راهنمای فنی و بهره‌برداری از این سامانه و فعال‌سازی سرویس‌های آن در گزارش بیان شده است.

به منظور نصب و ارزیابی سامانه، جلسات متعددی برگزار گردید که به طور خلاصه می‌توان به فرایند نصب حضوری بر روی سرورها، تست سامانه، رفع محدودیت‌های موجود (عدم دسترسی به اینترنت)، تحویل کدها به مجموعه فرماندهی انتظامی و برطرف کردن مسائل مرتبط با بازخورهای دریافت شده و نصب نسخه‌های جدید در طی این جلسات اشاره کرد. همچنین جلسات آموزشی با موضوعات متعدد از جمله‌ی این جلسات بود که در این گزارش به آن‌ها اشاره شده است. در این گزارش، سرویس‌ها و نحوه‌ی دسترسی آن‌ها با جزییات ذکر می‌گردد. ماژول‌های هسته‌ی مرکزی و دسترسی به آن‌ها و پاسخ هر یک از آن‌ها در حالات مختلف بیان می‌گردد.

با توجه به عدم قطعیت‌های محیطی و ویژگی‌های منحصربه‌فرد پروژه‌ها، مدیریت مخاطرات یک ضرورت غیرقابل اجتناب بوده که در بخش پایانی گزارش، شرحی از مخاطرات پروژه و راه‌حل‌های پیشنهادی آورده شده است.

و در نهایت به تشریح پیکربندی، معماری، ساختار و نحوه نصب سامانه بر روی سرور پرداخته شده است.

فصل ۱ راهنمایی فنی و بهره‌برداری سامانه، جلسات آموزشی

۱-۱ راهنمای فنی و بهره‌برداری سامانه

ساختار پروژه به ترتیب دارای سه لایه هوش مصنوعی، مدیریت و دمو می باشد. در قسمت هوش مصنوعی مسائل مربوط به پردازش تصاویر و ویدئو ها است که به لایه مدیریت سرویس می دهند. در لایه مدیریت، مسائل مربوط به کنترل میزان مصرف، بررسی سطح دسترسی بررسی میشود. همانطور که از اسم لایه دمو برداشت می‌شود صرفا برای ارائه دمو و آزمون سیستم توسط کاربر نهایی می‌باشد و صرفا به دلیل در خواست کارفرما پیاده‌سازی شده است.

۱-۱-۱ دسترسی به ماژول‌های هسته‌ی مرکزی

به منظور استفاده از سرویس‌های پروژه لازم است مقدار ApiToken را در هدر درخواست قرار دهید. این توکن محرمانه در اختیار شما قرار می‌گیرد و شناسه‌ای برای تشخیص دادن شما از سایر مشتریان است. همچنین رابط کاربری swagger این سرویس‌ها نیز در آدرس https://[ServerAddress]/swagger/index.html قابل دسترسی می‌باشد که می‌توان به راحتی از آن استفاده کرد. در ادامه به تشریح هر کدام از سرویس‌ها پرداخته شده است.

۱-۱-۱-۱ احراز هویت

برای احراز هویت کاربران از ApiToken استفاده می‌شود. این شناسه یکتا و ثابت است که به ازای هر کاربر صادر می‌شود. به این ترتیب در هنگام فراخوانی سرویس‌ها از این شناسه برای تشخیص دسترسی کاربر استفاده می‌شود و با توجه به میزان دسترسی سرویس داده می‌شود. در حال حاضر تعدادی از آن صادر شده و در اختیار ناجا قرار گرفته است. برای صدور توکن‌های جدید Endpoint مشخصی قرار داده شده است که با استفاده از آن می‌توان توکن‌های جدید صادر نمود.

همزمان با استفاده از این توکن، سرویس IdentityServer هم در نظر گرفته شده است که غیر فعال است و کاربردی ندارد. در صورت نیاز به توسعه پرتال‌های خاص می‌توان از آن برای کنترل سطح دسترسی نیز استفاده نمود. در حال حاضر به جهت امنیت و دسترسی هیچ نیازی به این سرویس نمی‌باشد و صرفا در صورت بروز نیازمندی‌های جدید از آن استفاده می‌شود.ویژگی‌های مشترک سرویس‌ها

در این قسمت ویژگی‌های که در خروجی یا ورودی سرویس‌ها مشترک هستند معرفی می‌شوند.

۱-۱-۱-۱-۱ میزان حساسیت سامانه

همه‌ی سرویس‌های هوش مصنوعی مقداری ورودی SensitivityType را از ورودی دریافت می‌نمایند که نشان‌دهنده میزان حساسیت سامانه نسبت به داده ورودی می‌باشد. مقدار VeryHigh به مفهوم بیشترین سخت گیری می‌باشد که به طبع در تشخیص‌هایی با بیشینه حساسیت باید از آن استفاده شود. پیشنهاد می‌شود در کاربردهای عادی از میزان سخت گیری Normal استفاده شود. جدول جدول ۱–۱ مقادیر معتبر برای میزان سخت‌گیری را نشان می‌دهد.

جدول ۱–۱ جزئیات مربوط به Enum حساسیت

مقدار فیلد	توضیح
VeryLow	حساسیت پایین، سخت‌گیری کم
Low	حساسیت نسبتاً پایین، سخت‌گیری نسبتاً کم
Normal	حساسیت متوسط، سخت‌گیری متوسط
High	حساسیت نسبتاً بالا، سخت‌گیری زیاد
VeryHigh	حساسیت بالا، سخت‌گیری زیاد

۱-۱-۱-۱-۲ نتیجه احراز هویت موفقیت آمیز

در هنگام فراخوانی سرویس‌های احراز هویت در صورت اجرای بدون خطای برنامه، خروجی برنامه شامل عبارت Status است که نشان‌دهنده نتیجه بازشناسی می‌باشد. در صورتی که مقدار این عبارت، Approved باشد به این معنی است که احراز هویت با موفقیت انجام شده است و در اطلاعات ورودی همخوانی وجود دارد. در صورتی که مقدار این عبارت Rejected باشد به این معنی است که احزار هویت موفقیت آمیز نبوده و بین اطلاعات ورودی همخوانی وجود ندارد. در نهایت، در صورت صورتی که نتیجه سامانه به Approved نزدیک باشد ولی شرایط لازم را نداشته باشد مقدار آن برابر OperatorCheck است که نشان می‌دهد مقادیر ورودی دارای همخوانی نسبی هستند و نیاز است تا توسط انسان یا سامانه (با سطح سخت گیری متفاوت) دوباره بررسی شوند. جدول ۱–۲ مقادیر مجاز Status را نشان می دهد.

جدول ۱–۲ جزئیات مربوط به Enum وضعیت

مقدار فیلد	توضیح
Approved	احراز هویت موفق
OperatorCheck	نیاز به بررسی اپراتور
Rejected	رد شدن احراز هویت

۱- ۱- ۱- ۱- ۳ نتیجه احراز هویت در صورت وقوع خطا

در هنگام فراخوانی سرویس های احراز هویت در برخی موارد به دلیل مشکل در فایل‌های ورودی (ارسال ویدئو به جای تصویر) و یا در دسترسی نبودن سرویس‌های داخلی، خطایی رخ می‌دهد. سامانه در صورت وقوع هر گونه خطا، پیغامی با ساختار زیر با HTTP Status Code مناسب آن خطا بر می‌گرداند (جدول ۱–۳ و جدول ۱–۴).


```
{
 "__unauthorizedRequest": true,
 "__wrapped": true,
 "__traceId": "",
 "error": {
  "errorCode": "USER_NOT_FOUND",
  "message": "User not found.",
  "details": "",
  "source": ""
 }
}
```


جدول ۱–۳ خروجی ناموفق - با بدنه Form-Data

نام فیلد	نوع فیلد	توضیح
__unauthorizedRequest	Boolean	-
__wrapped	Boolean	-
__traceId	String	-
error	Error	-

جدول ۱–۴ شرح فیلدهای شی Error

نام فیلد	نوع فیلد	توضیح
errorCode	String	کد خطا
message	String	شرح خطا
details	String	جزئیات خطا
source	String	منبع خطا

۱ -۱ -۱ -۲ سامانه شاهکار

در سرویس‌های تطبیق چهره و یا سایر سرویس‌های بر پایه‌ی تطبیق با چهره، یکی از تصاویر ورودی به عنوان عکس چهره مرجع در نظر گرفته می‌شود. با توجه به اینکه سامانه شاهکار تصاویر ثبت شده همه ایرانیان را دارد، می‌توان این سامانه را با سامانه‌های احراز هویت ترکیب نمود و همواره از عکس چهره سامانه شاهکار به عنوان عکس مرجع استفاده کرد. با توجه به حساسیت موجود بر روی سامانه شاهکار، در حال حاضر به این سامانه دسترسی نداشته ولی می‌توان با نصب سامانه‌های احراز هویت بر روی سرورهای ناجا، دسترسی مشخصی به این سرویس بر روی سرورهای ناجا گرفت و با اتصال به سامانه شاهکار، سامانه احراز هویت یکپارچه‌ای داشت.

۱-۱-۱-۳ تطبیق چهره دو تصویر

در سرویس تطبیق چهره (Face Verification) دو تصویر (یکی مرجعی و دیگری آزمون) با یکدیگر مقایسه شده و یک نتیجه از بین حالات جدول ۱–۲ بازگردانده می‌شود.

برای راحتی پیاده‌سازی، این سرویس با دو endpoint با نوع بدنه Form-Data و JSON پیاده‌سازی شده است. جزئیات نحوه‌ی فراخوانی این دو API در جدول ۱–۵ و جدول ۱–۶ و جدول ۱–۷ آمده است.

جدول ۱–۵ جزئیات نحوه فراخوانی API نطبیق چهره – با بدنه Form-Data

عنوان	توضیح
نوع API	POST
قالب URL	https://api.sepidid.com/api/verification/face-by-image
هدر موردنیاز	ApiToken

جدول ۱–۶ ورودی های سرویس تطبیق چهره - با بدنه Form-Data

نام فیلد	نوع فیلد	توضیح
SensitivityType	String	درجه حساسیت (جزئیات مربوط به این فیلد در جدول ۱–۱شرح داده شده است)
FirstImage	Byte []	عکس اول
SecondImage	Byte []	عکس دوم

جدول ۱–۷ خروجی موفق سرویس تطبیق چهره

نام فیلد	نوع فیلد	توضیح
SensitivityType	String	درجه حساسیت (جزئیات مربوط به این فیلد در جدول ۱–۱شرح داده شده است)
status	String	وضعیت (جزئیات مربوط به این فیلد در جدول ۱–۲ شرح داده شده است)

جدول ۱–۸ جزئیات نحوه فراخوانی API نطبیق چهره – با بدنه JSON

عنوان	توضیح
نوع API	POST
قالب URL	https://api.sepidid.com/api/verification/face-by-image-byte-array
هدر موردنیاز	ApiToken

جدول ۱–۹ ورودی های سرویس تطبیق چهره - با بدنه JSON

نام فیلد	نوع فیلد	توضیح
SensitivityType	String	درجه حساسیت (جزئیات مربوط به این فیلد در جدول ۱–۱شرح داده شده است)
FirstImage	Base64	عکس اول
SecondImage	Base64	عکس دوم

جدول ۱–۱۰ خروجی موفق سرویس تطبیق چهره

نام فیلد	نوع فیلد	توضیح
SensitivityType	String	درجه حساسیت (جزئیات مربوط به این فیلد در جدول ۱–۱شرح داده شده است)
status	String	وضعیت (جزئیات مربوط به این فیلد در جدول ۱–۲ شرح داده شده است)

در صورت ناموفق بودن درخواست، خطایی مطابق با بخش «خطاهای عمومی» و «خطاهای تطبیق چهره» برگشت داده می‌شود. لیست این خطاها به‌ترتیب در جدول ۱–۳۶ و جدول ۱–۳۷ قابل ملاحظه هستند.

۱-۱-۱-۴ تشخیص زنده بودن (غیرتعاملی)

این سرویس با دریافت یک ویدئو، با بررسی passive نتیجه معتبر بودن ویدئو را به عنوان نتیجه باز می‌گرداند. برای راحتی پیاده‌سازی، این سرویس با دو endpoint با نوع بدنه Form-Data (جدول ۱–۱۱ جدول ۱–۱۲) و JSON (جدول ۱–۱۳ و جدول ۱–۱۴) پیاده‌سازی شده است.

جدول ۱–۱۱ جزئیات نحوه فراخوانی API تشخیص زنده بودن (غیرتعاملی)– با بدنهForm-Data

عنوان	توضیح
نوع API	POST
قالب URL	https://api.sepidid.com/api/verification/passive-liveness
هدر موردنیاز	ApiToken

جدول ۱–۱۲ ورودی های سرویس تشخیص زنده بودن (غیرتعاملی) - با بدنهForm-Data

نام فیلد	نوع فیلد	توضیح
SensitivityType	String	درجه حساسیت (جزئیات مربوط به این فیلد در جدول ۱–۱شرح داده شده است)
Video	Byte []	ویدئو ضبط‌شده از سمت کاربر

جدول ۱–۱۳ جزئیات نحوه فراخوانی API تشخیص زنده بودن (غیرتعاملی)– با بدنه json

عنوان	توضیح
نوع API	POST
قالب URL	https://api.sepidid.com/api/verification/passive-liveness-byte-array
هدر موردنیاز	ApiToken

جدول ۱–۱۴ ورودی های سرویس تشخیص زنده بودن (غیرتعاملی) - با بدنهJson

نام فیلد	نوع فیلد	توضیح
SensitivityType	String	درجه حساسیت (جزئیات مربوط به این فیلد در جدول ۱–۱شرح داده شده است)
Video	Base64	ویدئو ضبط‌شده از سمت کاربر

جدول ۱–۱۵ خروجی موفق سرویس تشخیص زنده بودن(غیرتعاملی)

نام فیلد	نوع فیلد	توضیح
SensitivityType	String	درجه حساسیت (جزئیات مربوط به این فیلد در جدول ۱–۱شرح داده شده است)
status	String	وضعیت (جزئیات مربوط به این فیلد در جدول ۱–۲ شرح داده شده است)

در صورت ناموفق بودن درخواست، خطایی مطابق با بخش «خطاهای عمومی» و « خطاهای تشخیص زنده بودن » برگشت داده میشود.لیست این خطاها در بخش پیوست به‌ترتیب در جدول ۱–۳۶ و جدول ۱–۳۷ قابل ملاحظه هستند.

۱ -۱ -۱ -۵ تشخیص زنده بودن (تعاملی: پلک زدن)

برای استفاده از ماژول تشخیص زنده پلک زدن، نیاز است ابتدا یک «الگو» دریافت کنیم (جدول ۱–۱۶ و جدول ۱–۱۷). این الگو شامل لیستی از اعداد صحیح (به میلی‌ثانیه) است که مشخص می‌کند فرد در چه لحظاتی لازم است پلک بزند. شما به‌عنوان توسعه‌دهنده لازم است با دریافت این لیست در رابطه کاربری خود در لحظات مناسب (به مدت یک ثانیه) از کاربر بخواهید پلک بزند. به عنوان مثال اگر الگوی دریافتی شامل اعداد ۴۵۰۰ و ۸۸۰۰ باشد لازم است در دو بازه زیر از کاربر بخواهید چشم‌های خود را ببندد:

- ۴۵۰۰ میلی‌ثانیه پس از شروع ضبط ویدئو تا ۶۵۰۰ میلی‌ثانیه پس از شروع ضبط
- ۸۸۰۰ میلی‌ثانیه پس از شروع ضبط ویدئو تا ۱۰۸۰۰ میلی‌ثانیه پس از شروع ضبط

سپس لازم است «کد الگوی دریافتی» را به همراه ویدئو ضبط‌شده به ماژول تشخیص زنده بودن ارسال کنید. این ماژول یک نتیجه از بین حالات جدول ۱–۲ را باز می‌گرداند.

۱– ۱– ۱– ۵– ۱ نحوه دریافت الگو

جدول ۱–۱۶ جزئیات فراخوانی سرویس دریافت الگو

عنوان	توضیح
نوع API	GET
قالب URL	https://api.sepidid.com/api/verification/active-liveness-pattern
هدر موردنیاز	ApiToken

جدول ۱–۱۷ خروجی موفق سرویس دریافت الگو

نام فیلد	نوع فیلد	توضیح
requestId	string	کد درخواست (مقدار این فیلد باید در مرحله بعدی به همراه ویدئو ارسال شود.)
type	string	نوع. دارای مقدار BlinkTimes
value	string	رشته‌ای است شامل اعداد صحیح که نمایانگر لحظه‌های موردنظر به میلی‌ثانیه است. به‌عنوان مثال اعداد ۴۵۰۰ و ۸۸۰۰ به صورت "4500,8800" بازگردانده می‌شود

۱– ۱– ۱– ۵– ۲ نحوه فراخوانی سرویس تشخیص زنده بودن

این سرویس با دریافت یک ویدئو، و یک «کد الگو» با بررسی زمان‌های پلک زدن، نتیجه معتبر بودن ویدئو را به عنوان نتیجه باز می‌گرداند. برای راحتی پیاده‌سازی، این سرویس با دو endpoint با نوع بدنه Form-Data (جدول ۱–۱۸ و جدول ۱–۱۹) و JSON پیاده سازی شده است.

جدول ۱–۱۸ جزئیات نحوه فراخوانی سرویس تشخیص زنده بودن – با بدنه Form-Data

عنوان	توضیح
نوع API	POST
قالب URL	https://api.sepidid.com/api/verification/active-liveness
هدر موردنیاز	ApiToken

جدول ۱–۱۹ورودی های سرویس تشخیص زنده بودن - با بدنه Form-Data

نام فیلد	نوع فیلد	توضیح
SensitivityType	String	درجه حساسیت (جزئیات مربوط به این فیلد در جدول ۱–۱شرح داده شده است)
RequestId	string	کد درخواست دریافتی از مرحله قبل
Video	Byte []	ویدیو

جدول ۱–۲۰ خروجی موفق سرویس تشخیص زنده بودن

نام فیلد	نوع فیلد	توضیح
SensitivityType	String	درجه حساسیت (جزئیات مربوط به این فیلد در جدول ۱–۱شرح داده شده است)
status	String	وضعیت (جزئیات مربوط به این فیلد در جدول ۱–۲ شرح داده شده است)

در صورت ناموفق بودن درخواست، خطایی مطابق با بخش «خطاهای عمومی» و « خطاهای تشخیص زنده بودن » برگشت داده میشود. لیست این خطاها در بخش پیوست به‌ترتیب در جدول ۱–۳۶ و جدول ۱–۳۷ قابل ملاحظه هستند.

۱ -۱ -۱ -۶ تشخیص زنده بودن (تعاملی: لب خوانی)

برای استفاده از ماژول تشخیص زنده لب خوانی، نیاز است ابتدا یک «الگو» دریافت کنیم (جدول ۱–۲۱ و جدول ۱–۲۲). این الگو شامل یک جمله کوتاه بوده که کاربر آن‌را روخوانی می‌نماید. شما به‌عنوان توسعه‌دهنده لازم است با دریافت این جمله در رابطه کاربری خود، از کاربر بخواهید جمله را روخوانی نماید و همزمانی با روخوانی کاربر، تصویر او را ضبط نمایید. سپس لازم است «کد الگوی دریافتی» را به همراه ویدئو ضبط‌شده به ماژول تشخیص زنده بودن با لب خوانی ارسال کنید.

به این ترتیب، ویدویی از کاربر تهیه می‌شود که در آن در حال بازخوانی یک متن است و سامانه با توجه به حرکت لب‌ها و همچنین متن ارسالی، زنده بودن شخص را تشخیص می‌دهد.

۱– ۱– ۱– ۶– ۱ نحوه دریافت الگو

جدول ۱–۲۱ جزئیات فراخوانی سرویس دریافت الگو

عنوان	توضیح
نوع API	GET
قالب URL	https://api.sepidid.com/api/verification/lip-reading-liveness-pattern
هدر موردنیاز	ApiToken

جدول ۱–۲۲ خروجی موفق سرویس دریافت الگو

نام فیلد	نوع فیلد	توضیح
requestId	string	کد درخواست (مقدار این فیلد باید در مرحله بعدی به همراه ویدئو ارسال شود.)
type	string	نوع. دارای مقدار LipreadingV1
value	string	مقدار این قسمت همان جمله‌ای است که کاربر روخوانی می‌کند و همزمان تصویر او ضبط می-شود .

۱- ۱- ۱- ۶- ۲ نحوه فراخوانی سرویس تشخیص زنده بودن (تعاملی: لب خوانی)

جدول ۱–۲۳ جزئیات نحوه فراخوانی سرویس تشخیص زنده بودن (تعاملی: لب خوانی)– با بدنهForm-Data

عنوان	توضیح
نوع API	POST
قالب URL	https://api.sepidid.com/api/verification/lip-reading-liveness
هدر موردنیاز	ApiToken

جدول ۱–۲۴ ورودی های سرویس تشخیص زنده بودن (تعاملی: لب خوانی)- با بدنهForm-Data

نام فیلد	نوع فیلد	توضیح
SensitivityType	String	درجه حساسیت (جزئیات مربوط به این فیلد در جدول ۱–۱شرح داده شده است)
RequestId	string	کد درخواست دریافتی از مرحله قبل
Video	Byte []	ویدیو

جدول ۱–۲۵ خروجی موفق سرویس تشخیص زنده بودن (تعاملی: لب خوانی)

نام فیلد	نوع فیلد	توضیح
SensitivityType	String	درجه حساسیت (جزئیات مربوط به این فیلد در جدول ۱–۱شرح داده شده است)
status	String	وضعیت (جزئیات مربوط به این فیلد در جدول ۱–۲ شرح داده شده است)

در صورت ناموفق بودن درخواست، خطایی مطابق با بخش «خطاهای عمومی» و « خطاهای تشخیص زنده بودن» برگشت داده میشود. لیست این خطاها در بخش پیوست به‌ترتیب در جدول ۱–۳۶ و جدول ۱–۳۸ قابل ملاحظه هستند.

۱- ۱- ۱- ۶- ۳ محدودیت ها و ملاحظات سرویس

- حداقل ابعاد ویدئو ۴۰۰ پیکسل (برای طول یا عرض) است.
- حداکثر ابعاد ویدئو ۲۰۰۰ پیکسل (برای طول یا عرض) است.
- حداقل طول ویدئو یک ثانیه است.
- حداکثر طول ویدئو ۱۵ ثانیه است.

۱-۱-۱-۷- سرویس تشخیص زنده بودن (تعاملی: تشخیص گفتار)

برای استفاده از ماژول تشخیص زنده بودن با تشخیص گفتار، نیاز است ابتدا یک «الگو» دریافت کنیم (جدول ۱–۲۶ و جدول ۱–۲۷). این الگو شامل یک جمله کوتاه بوده که کاربر آنرا روخوانی می‌نماید. شما به‌عنوان توسعه‌دهنده لازم است با دریافت این جمله در رابطه کاربری خود، از کاربر بخواهید جمله را روخوانی نماید و همزمانی با روخوانی کاربر، تصویر و صدای او را ضبط نمایید. سپس لازم است «کد الگوی دریافتی» را به همراه ویدئو ضبط‌شده به ماژول تشخیص زنده بودن با تشخیص گفتار ارسال کنید.

به این ترتیب، ویدویی از کاربر تهیه می‌شود که در آن در حال بازخوانی یک متن است و سامانه با توجه به نحوه گفتار و همچنین متن ارسالی، زنده بودن شخص را تشخیص می‌دهد.

۱-۱-۱-۷-۱ نحوه دریافت الگو

جدول ۱–۲۶ جزئیات فراخوانی سرویس دریافت الگو

عنوان	توضیح
نوع API	GET
قالب URL	https://api.sepidid.com/api/verification/speech-liveness-pattern
هدر موردنیاز	ApiToken

جدول ۱–۲۷ خروجی موفق سرویس دریافت الگو

نام فیلد	نوع فیلد	توضیح
requestId	string	کد درخواست (مقدار این فیلد باید در مرحله بعدی به همراه ویدئو ارسال شود.)
type	string	نوع. دارای مقدار SpeechRecognition
value	string	مقدار این قسمت همان جمله‌ای است که کاربر روخوانی می‌کند و همزمان تصویر او ضبط می‌شود .

۱-۱-۱-۷-۲ نحوه فراخوانی سرویس تشخیص زنده بودن (تعاملی: تشخیص گفتار)

جدول ۱–۲۸ جزئیات نحوه فراخوانی سرویس تشخیص زنده بودن (تعاملی: تشخیص گفتار)– با بدنه Form-Data

عنوان	توضیح
نوع API	POST
قالب URL	https://api.sepidid.com/api/verification/speech-liveness
هدر موردنیاز	ApiToken

جدول ۱–۲۹ ورودی های سرویس تشخیص زنده بودن (تعاملی: تشخیص گفتار)- با بدنهForm-Data

نام فیلد	نوع فیلد	توضیح
SensitivityType	String	درجه حساسیت (جزئیات مربوط به این فیلد در جدول ۱–۱شرح داده شده است)
RequestId	string	کد درخواست دریافتی از مرحله قبل
Video	Byte []	ویدیو

جدول ۱–۳۰ خروجی موفق سرویس تشخیص زنده بودن (تعاملی تشخیص گفتار)

نام فیلد	نوع فیلد	توضیح
SensitivityType	String	درجه حساسیت (جزئیات مربوط به این فیلد در جدول ۱–۱شرح داده شده است)
status	String	وضعیت (جزئیات مربوط به این فیلد در جدول ۱–۲ شرح داده شده است)

در صورت ناموفق بودن درخواست، خطایی مطابق با بخش «خطاهای عمومی» و « خطاهای تشخیص زنده بودن » برگشت داده میشود. لیست این خطاها در بخش پیوست به‌ترتیب در جدول ۱–۳۶ و جدول ۱–۳۸ قابل ملاحظه هستند.

محدودیت‌ها و ملاحظات سرویس عبارتند از:

- حداقل ابعاد ویدئو ۴۰۰ پیکسل (برای طول یا عرض) است.
- حداکثر ابعاد ویدئو ۲۰۰۰ پیکسل (برای طول یا عرض) است.
- حداقل طول ویدئو یک ثانیه است.
- حداکثر طول ویدئو ۱۵ ثانیه است.

۱ -۱ -۱ -۸ تطبیق چهره و تشخیص زنده بودن (غیرتعاملی)

این سند به معرفی API مربوط به تطبیق چهره (Face Verification) و تشخیص زنده بودن مبتنی بر تحلیل ویدئو (Liveness Detection) و نحوه استفاده از آن می‌پردازد. این دو ماژول (تطبیق چهره و تشخیص زنده بودن) در قالب یک API به‌صورت یکپارچه قابل دسترسی است (جدول ۱–۳۱، جدول ۱–۳۲، جدول ۱–۳۳، جدول ۱–۳۴ و جدول ۱–۳۵) و در راهکارهای احراز هویت الکترونیکی مانند شناسایی مشتریان از راه دور (e-KYC) مورد استفاده قرار می‌گیرند. چهره استخراج‌شده از تصویر ارسالی به عنوان مرجع با چهره موجود در ویدئو مقایسه می‌شود.

جدول ۱–۳۱ جزئیات نحوه فراخوانی سرویس تطبیق چهره و تشخیص زنده بودن (غیرتعاملی)– با بدنه Form-Data

عنوان	توضیح
نوع API	POST
قالب URL	https://api.sepidid.com/api/verification/video-by-image
هدر موردنیاز	ApiToken

جدول ۱–۳۲ ورودی های سرویس تطبیق چهره و تشخیص زنده بودن (غیرتعاملی)- با بدنه Form-Data

نام فیلد	نوع فیلد	توضیح
SensitivityType	String	درجه حساسیت (جزئیات مربوط به این فیلد در جدول ۱–۱شرح داده شده است)
Image	Byte []	عکس مرجع جهت تطبیق چهره
Video	Byte []	ویدیو چهت بررسی لایونس و تطبیق چهره

جدول ۱–۳۳ جزئیات نحوه فراخوانی سرویس تطبیق چهره و تشخیص زنده بودن (غیرتعاملی)– با بدنهJson

عنوان	توضیح
نوع API	POST
قالب URL	https://api.sepidid.com/api/verification/video-by-image-byte-array
هدر موردنیاز	ApiToken

جدول ۱–۳۴ ورودی های سرویس تطبیق چهره و تشخیص زنده بودن (غیرتعاملی)- با بدنهJson

نام فیلد	نوع فیلد	توضیح
SensitivityType	String	درجه حساسیت (جزئیات مربوط به این فیلد در جدول ۱–۱شرح داده شده است)
Image	Base64	عکس مرجع جهت تطبیق چهره
Video	Base64	ویدیو چهت بررسی لایونس و تطبیق چهره

جدول ۱–۳۵ خروجی موفق سرویس تطبیق چهره و تشخیص زنده بودن (غیرتعاملی)

نام فیلد	نوع فیلد	توضیح
SensitivityType	String	درجه حساسیت (جزئیات مربوط به این فیلد در جدول ۱–۱شرح داده شده است)
status	StatusEnum	وضعیت درخواست
livenessStatus	StatusEnum	وضعیت درخواست تشخیص زنده بودن
verificationStatus	StatusEnum	وضعیت درخواست تطبیق چهره

در صورت ناموفق بودن درخواست، خطایی مطابق با بخش «خطاهای عمومی» و « خطاهای تشخیص زنده بودن» برگشت داده می‌شود. لیست این خطاها در بخش پیوست به‌ترتیب در جدول ۱–۳۶ و جدول ۱–۳۷ قابل ملاحظه هستند.

.

۱ -۱ -۲ کدهای خطا

در صورت بروز خطا در سیستم خطای مناسب آن خطا برگردانده میشود. از جمله موارد بروز خطا می توان به ارسال فایل با فرمت اشتباه و یا سایر موارد لیست شده در بخش های بعدی اشاره نمود. در صورت وقوع خطا در سیستم، ساختار خطای زیر با کد وضعیت (HTTP Status Code) مناسب برگدانده میشود.

```
{
 "__unauthorizedRequest": true,
 "__wrapped": true,
 "__traceId": "",
 "error": {
  "errorCode": "UNSUPPORTED_IMAGE_FORMAT",
  "message": "Invalid image file, supported formats are JPG,JPEG,PNG,BMP,and TIF.",
  "details": "",
  "source": ""
 }
}
```

۱ -۱ -۲ -۱ خطاهای عمومی

این خطاها به واسطه اضافه کردن یک تصویر یا یک ویدئو ممکن است رخ دهد (جدول ۱–۳۶).

جدول ۱–۳۶ کد خطاهای عمومی

کد خطا (ErrorCode)	جزئیات (Message)	کد وضعیت	توضیح
INVALID_REQUEST_BODY	Invalid request body, missing: 'image'	400	درخواست نامعتبر است

کد خطا (ErrorCode)	جزئیات (Message)	کد وضعیت	توضیح
UNSUPPORTED_IMAGE_FORMAT	Invalid image file, supported formats are JPG, JPEG, PNG, BMP, and TIF.	400	فرمت تصویر پشتیبانی نمی‌شود
UNSUPPORTED_VIDEO_FORMAT	Invalid video file, supported video formats are WebM, MP4, MOV, and AVI.	400	فرمت ویدئو پشتیبانی نمی‌شود
TOO_SMALL_IMAGE_DIMENTIONS	The minimum image size is 100 pixels for both height and width.	400	ابعاد تصویر بسیار کوچک است
TOO_LARGE_IMAGE_DIMENTIONS	The maximum image size is 7000 pixels for both height and width.	400	ابعاد تصویر بسیار بزرگ است
TOO_SMALL_VIDEO_DIMENTIONS	The minimum video size is 300 pixels for both height and width.	400	ابعاد ویدئو بسیار کوچک است
TOO_LARGE_VIDEO_DIMENTIONS	The maximum video size is 2000 pixels for both height and width.	400	ابعاد ویدئو بسیار بزرگ است
TOO_SHORT_VIDEO_LENGTH	The minimum video length is 1 second(s).	400	ویدئو بسیار کوتاه است
TOO_LONG_VIDEO_LENGTH	The maximum video length is 30 seconds.	400	ویدئو بسیار طولانی است
SERVICE_UNAVAILABLE	The server is temporarily unable to service your request due to maintenance downtime or capacity problems. Please try again later.	503	سرویس موقتاً در دسترس نیست
INTERNAL_SERVER_ERROR	The server encountered an internal error and was unable to complete your request. Either the server is overloaded or there is an error in the application.	500	خطایی در سرور رخ داده است

۱-۱-۲-۲ خطاهای تطبیق چهره و تشخیص زنده بودن غیرتعاملی

در صورتی که یکی از وظایف سرویس انجام عملیات تطبیق چهره یا بررسی تشخیص زنده بودن غیرتعاملی باشد، ممکن است یکی از خطاهای جدول ۱–۳۷ رخ دهد.

جدول ۱–۳۷ کد خطاهای تطبیق چهره

کد خطا (ErrorCode)	جزئیات (Message)	کد وضعیت	توضیح
TOO_DARK_VIEW	Image is too dark, try again with proper lightning.	400	نور تصویر بسیار کم است
TOO_LIGHT_VIEW	Image is too light, try again with proper lightning.	400	نور تصویر بسیار زیاد است
LOW_QUALITY	Image quality is poor.	400	کیفیت تصویر نامطلوب است
UNACCEPTABLE_BACKLIGHT	Remove backlight and try again.	400	نور پس‌زمینه زیاد است
NO_FACE_DETECTED	Can not detect any face; make sure your face is clearly visible in the image.	400	چهره در تصویر شناسایی نشد
	Can not detect face; make sure your face is clearly visible in every single frame of the the video.	400	چهره در تمام طول ویدئو شناسایی نشد

کد خطا (ErrorCode)	جزئیات (Message)	کد وضعیت	توضیح
MULTIPLE_FACE_DETECTED	More than one face detected; record the video again with a plain background.	400	بیش از یک چهره در تصویر شناسایی شد
	More than one face detected; record the video again with a plain background.	400	بیش از یک چهره در ویدئو شناسایی شد
SMALL_FACE_SIZE	The minimum face size is 150 pixels.	400	اندازه چهره در تصویر کوچک است
SMALL_INTERPUPILLARY_DISTANCE	The minimum interpupillary distance is 80 pixels.	400	فاصله بین مرکز دو چشم در تصویر کم است
CROPPED_FACE	Face should appear in center of the frame.	400	چهره در مرکز تصویر قرار نگرفته است

۱-۱-۲-۳ خطاهای تشخیص زنده بودن تعاملی

در صورتی که یکی از وظایف سرویس، تشخیص زنده بودن تعاملی باشد، ممکن است یکی از خطاهای جدول ۱–۳۸ رخ دهد.

جدول ۱–۳۸ کد خطاهای تشخیص زنده بودن تعاملی

کد خطا (ErrorCode)	جزئیات (Message)	کد وضعیت	توضیح
INVALID_ACTIVE_LIVENESS_TOKEN	Pattern token is not valid.	400	کد درخواست؛ نامعتبر است
EXPIRED_ACTIVE_LIVENESS_TOKEN	Your pattern token is expired, you can use your pattern id for 3 minutes.	400	کد درخواست؛ منقضی شده است
USED_ACTIVE_LIVENESS_TOKEN	Your pattern token have used once, please try getting an other pattern id and perform active liveness again.	400	کد درخواست قبلا استفاده شده است
INVALID_ACTIVE_LIVENESS_OPERATION	The video length is not compatible with pattern token.	400	طول ویدئو مطابق با شرایط اجرای درخواست نیست
	You used this pattern token too early.	400	درخواست؛ سریعتر از حد انتظار ارسال شد

۱-۱-۳ ملاحظات فنی و جامعه بهره‌بردار

برای اطمینان از اینکه سیستم بهترین عملکرد خود را ارائه می‌دهد، مجموعه‌ای از محدودیت‌ها ارائه شده است که باید در حین استفاده از سامانه رعایت شوند. این محدودیت‌ها هم برای توسعه‌دهندگان و برنامه‌نویسان شرکت پژوهش و توسعه و هم برای کاربران عادی سامانه می‌باشد.

در این سند ملاحظاتی که کاربر نهایی[1] می‌بایست با در نظر گرفتن آن‌ها از سامانه احراز هویت غیرحضوری استفاده کند و همچنین محدودیت‌های فنی که تیم توسعه[2] استفاده‌کننده از سرویس‌های این سامانه باید مدنظر قرار دهند، مرور شده‌اند. سازمان‌ها و کسب و کارهایی که این سامانه را مورد استفاده قرار می‌دهد لازم است در رابط کاربری خود محدودیت‌های ذکرشده سطح کاربر

[1] End User
[2] Developer

را درنظر گرفته و در مورد ملاحظات به کاربر بهره‌بردار اطلاع دهد تا حد امکان از ارسال تصویر و ویدئوی نامناسب جلوگیری کند. همچنین تیم توسعه لازم است ملاحظات فنی را در یکپارچه‌سازی در نظر بگیرد.

۱ -۱ -۳ -۱ ملاحظات سطح کاربر

برای بهره‌گیری هرچه بهتر از سرویس‌های تطبیق چهره و تشخیص زنده بودن لازم است در تصاویر و ویدئوهای ورودی دریافت شده از کاربر، شرط‌های زیر برقرار باشد؛ لازم به ذکر است که رعایت نکردن این نکات ممکن است باعث شود که سامانه به اشتباه زنده بودن یک چهره را تایید یا رد کند و یا در تطبیق دو چهره دچار اشتباه شود.

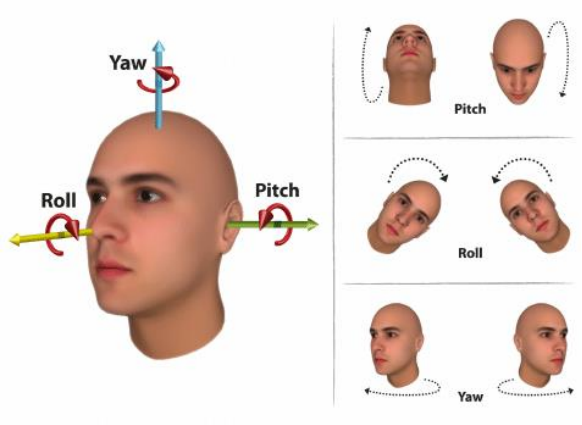
- در هریک از تصاویر یا ویدئوها (تمامی فریم‌ها)، دقیقاً یک چهره (چهره خودتان) وجود داشته باشد.
- محیطی که در آن سلفی می‌گیرید یا ویدئو ضبط می‌کنید، نور کافی داشته باشد.
- در زمان گرفتن تصویر/ویدئو نور روبروی چهره شما باشد و در پشت سر شما نور نباشد تا چهره شما (بخشی از صورت یا همه) تاریک نشود.
- در حین تصویربرداری (ویدئو یا عکس) دستتان زیاد تکان نخورد تا تصویر تار نباشد.
- کیفیت دوربین مورد استفاده مناسب باشد.
- چهره شما به صورت مستقیم و روبروی دوربین باشد و سر دارای چرخش نباشد.
- پیشنهاد می‌شود کاربر روبه‌روی دوربین نشسته یا ایستاده باشد و ارتفاع دوربین از سطح زمین با ارتفاع صورت وی حدوداً برابر باشد.
- برای دریافت نتیجه مطلوب اکیداً پیشنهاد می‌شود وضوح تصویر و فاصله‌ی چهره از دوربین به‌گونه‌ای باشد که ابعاد چهره در داخل کادر تعیین شده در نرم‌افزار قرار گیرد.
- برای استفاده از سرویس پلک زدن، فقط در زمان‌هایی که سامانه مشخص می‌کند پلک بزنید، در سایر موارد از پلک زدن بپرهیزید.
- برای استفاده از سرویس تشخیص گفتار، جمله نشان داده شده را با صدای رسا و شمرده بخوانید و جمله را در یک محیط ساکت (بدون سر و صدا) بخوانید.

۱ -۱ -۳ -۲ محدودیت‌های فنی

محدودیت‌های و ملاحظات فنی برای توسعه و یکپارچه‌سازی در فراخوانی سرویس‌ها به شرح زیر است:

- فرمت‌های قابل پشتیبانی برای تصویر عبارتند از: JPG، JPEG، PNG، BMP و TIF
- فرمت‌های قابل پشتیبانی برای ویدئو عبارتند از: WebM، MP4، MOV و AVI
- حداقل ابعاد قابل پذیرش برای تصویر چهره ۱۰۰ پیکسل و برای ویدئو ۳۰۰ پیکسل (برای طول یا عرض) است
- حداکثر ابعاد قابل پذیرش برای تصویر ۷۰۰۰ پیکسل و برای ویدئو ۲۰۰۰ پیکسل (برای طول یا عرض) است.
- حداکثر طول قابل پذیرش برای ویدئو ۳۰ ثانیه است.
- حداکثر حجم فایل ۲۰ مگابایت است.
- پیشنهاد می‌شود فایل‌ها تا حد امکان بدون فشرده‌سازی و یا با حداقل فشرده‌سازی ارسال شوند تا کیفیت آن‌ها کاهش پیدا نکند، به عنوان مثال در صورتی که از فرمت JPEG استفاده می‌کنید، شاخص کیفیت آن را کمتر از ۷۰ (از ۱۰۰) قرار ندهید.
- حداکثر چرخش برون صفحه (Out-of-plane rotation) (شکل ۱—۱) قابل قبول برای سر کاربر ۲۰ درجه (برای زوایای pitch و yaw) در هر جهت است.

- پیشنهاد می‌شود چرخش درون صفحه (In-plane rotation) سر کاربر کمتر از ۴۰ درجه (زاویه roll) در هر جهت باشد. گرچه این امکان در سامانه پیش‌بینی شده است که تصاویر و ویدئو با چرخیده و برعکس را نیز تشخیص دهد اما برای افزایش کارایی سامانه، پیشنهاد می‌شود رابط کاربری (خصوصا در تلفن همراه) به گونه‌ای طراحی شود که جهت چرخش دوربین را مدیریت کرده و تصاویر و ویدئوها را بدون چرخش ارسال کند.



شکل ۱–۱ جهت‌های چرخش سر

- برای دریافت نتیجه مطلوب پیشنهاد می‌شود وضوح تصویر و فاصله‌ی چهره از دوربین به‌گونه‌ای باشد که ابعاد چهره در هر تصویر حداقل ۱۵۰ پیکسل باشد و به عنوان مثال فیلم‌برداری با وضوح ۷۲۰ گزینه مناسبی است.
- با توجه به سطح کارایی روش‌های تشخیص زنده بودن غیرتعاملی، لازم است در تشخیص زنده بودن از این روش به تنهایی استفاده نشود و در کنار روش‌های تشخیص زنده بودن تعاملی مانند تشخیص گفتار و تشخیص پلک زدن به کار گرفته شود. بدین منظور در APIهای ارائه شده، API تشخیص زنده بودن به صورت ترکیب روش‌های تعاملی و غیرتعاملی ارائه شده است که لازم است از آن‌ها استفاده شود.

۱ -۱ -۳ -۳ جامعه بهره‌بردار سامانه

جامعه‌ی بهره‌بردار به تصمیم مجموعه‌ی فرمانده‌ی نیروی انتظامی انتخاب می‌گردد و ما نقشی در تعیین آن نداریم. سامانه برای هر نوع جامعه‌ای که آن‌ها تصمیم بگیرند قابل استفاده می‌باشد اما در این راستا، پیشنهاد می‌شود که در ابتدا یک جامعه‌ی محدود و در کنترل انتخاب گردد بتوان بازخوردها را بررسی و در صورت رخ دادن خطا بتوان آن‌ها را برطرف نمود.

۱- ۲ جلسات آموزشی

با مشخص شدن ساختار، جلسات متعددی برگزار شد (۱۲ جلسه). در طی این جلسات سیستم تحویل کارفرما گردید و بعضی جلسات به آموزش‌های لازم در جهت نصب و به کارگیری سیستم اختصاص پیدا کرد. در ادامه به زمان و محل این جلسات پرداخته می‌شود.

۱ -۲ -۱ اولین جلسه (۲۶ – ۴ – ۱۴۰۰): نصب اولیه هسته هوش مصنوعی

برای اولین بار سرور در اختیار مجری قرار گرفت تا برنامه‌های مورد نیاز همچون داکر نصب و ایمیج‌های مورد نیاز دانلود شوند. در ابتدای حضور، سرور از سمت فرماندهی انتظامی آماده نبود و بخشی از زمان صرف انتظار برای آماده‌سازی آن شد. در ادامه به

دلیل کندی زیاد در سرعت اینترنت، دانلود برنامه‌های مورد نیاز زمان زیادی را به خود گرفت و در نهایت نسخه اول با سرویس‌های اولیه تطبیق چهره و زنده بودن در اختیار همکاران فرماندهی انتظامی قرار گرفت. محل انجام کار ساختمان حکمت بود.

۱-۲-۲ دومین جلسه (۱۱ - ۷ - ۱۴۰۰): نصب نسخه جدید هسته هوش مصنوعی و دموی تحت وب

هدف از این جلسه نصب نسخه جدید و اضافه نمودن سایت دمو بوده است. در این جلسه به دلیل مشکل در اینترنت فرماندهی انتظامی نصب به صورت کامل انجام نشد و تصمیم بر این شد تا همه برنامه‌های مورد نیاز به صورت آفلاین و از قبل تنظیم شده آماده شوند. در طی این جلسه، اینترنت فرماندهی انتظامی برای دریافت برخی از پکیج‌های رایج جهت نصب کاملا قطع بود. (محل کار ساختمان حکمت).

۱-۲-۳ سومین جلسه (۲۶ – ۷ - ۱۴۰۰): نصب هسته و دمو به صورت آفلاین

به دلیل مشکلات پیش آمده در جلسات قبل و قطعی اینترنت، تمام نیازمندی‌ها به صورت آفلاین تهیه و بعد از تست توسط تیم فنی مجری در محل توسعه، برای نصب به محل فرماندهی انتظامی آورده شد و نصب به درستی انجام گردید. وب‌سایت دمو جهت نمایش سرویس‌های نصب شده در این جلسه به همکاران فرماندهی انتظامی تحویل داده شد (مکان: ساختمان حکمت).

۱-۲-۴ چهارمین جلسه (۲۸ – ۷ - ۱۴۰۰): رفع مشکل دسترسی

به دلیل محدودیت‌های شبکه و بسته بودن پورت ۱۲۰۰۰ سرور نصب، همکاران فرماندهی انتظامی از مرکز فرماندهی انتظامی واقع در میدان عطار به وب‌سایت دمو دسترسی نداشتند که با حضور در محل و ارتباط با تیم شبکه فرماندهی انتظامی مشکل برطرف گردید (مکان: فرماندهی انتظامی کل جمهوری اسلامی – میدان عطار).

۱-۲-۵ پنجمین جلسه (۲۴ – ۰۸ - ۱۴۰۰): نصب نسخه جدید دموی تحت وب با قابلیت‌های جدید

با توجه به نیازمندی‌های فرماندهی انتظامی، برخی سرویس‌ها و امکانات به وب‌سایت دمو افزوده گردید (مانند دریافت ویدئو) و نصب به صورت کامل انجام شد (مکان: ساختمان حکمت).

۱-۲-۶ ششمین جلسه (۲۲-۰۹-۱۴۰۰): هماهنگی مدیریتی برای یکپارچه‌سازی

دراین جلسه، باحضور کارشناسان شرکت پژوهش و توسعه ناجی و فرماندهی انتظامی، به ارائه نرم‌افزار پرداخته شد و راه‌های نصب و اتصال نرم افزار به سامانه‌های فرماندهی انتظامی مورد بررسی قرار گرفت (مکان: ساختمان فرماندهی انتظامی کل جمهوری اسلامی، میدان عطار).

۱-۲-۷ هفتمین جلسه (۲۲-۰۹-۱۴۰۰): هماهنگی فنی برای یکپارچه‌سازی

جلسه‌ای کاملا فنی با حضور دوستان شرکت پژوهش و توسعه و فرماندهی انتظامی برگزار شد و در آن نحوه اتصال نرم‌افزار با سامانه‌های فرماندهی انتظامی بررسی گردید. در این دیدار برای همگام‌سازی نرم‌افزار با سامانه‌های فرماندهی انتظامی سه راه حل زیر از سمت تیم مجری به شرح زیر ارائه گردید (مکان: ساختمان شرکت پژوهش و توسعه واقع در میدان هفتم تیر).

- **استفاده از مکانیزم Message Broker**: به دلیل نامتعارف بودن حجم درخواست‌ها در ساعات مختلف روز، پیشنهاد گردید از سیستم Event-Sourcing برای اتصال نرم‌افزار به سیستم‌های فرماندهی انتظامی استفاده شود. به این ترتیب، یک صف از درخواست‌ها ایجاد شده و با توجه به میزان منابع سیستم، به آن‌ها پاسخ داده می‌شود. در این رویکرد، هر دو سامانه مجری و شرکت پژوهش و توسعه به یک صف از درخواست و پاسخ وصل شده و از این طریق با یکدیگر تعامل می‌نمایند. دوستان شرکت پژوهش و توسعه ترجیح دادند که در این مرحله از این مکانیزم استفاده نشود.

- **استفاده از مکانیزم Web Hook**: همانند مورد قبل، به دلیل ناهمگون بودن حجم درخواست‌ها، پیشنهاد گردید تا صفی در سامانه مجری قرار داده شود و درخواست‌های سامانه شرکت پژوهش و توسعه در سمت سامانه مجری و در یک صف ذخیره شوند و بعد از پردازش درخواست‌ها، نتیجه آن از طریق فراخوانی یک اندپوینت RESTful در سمت سامانه‌های شرکت پژوهش و توسعه، برای دوستان شرکت پژوهش و توسعه ارسال شود. همانند مورد قبل، دوستان شرکت پژوهش و توسعه ترجیح دادند که در این مرحل از این مکانیزم استفاده نشود.

- **فراخوانی و انتظار**: این رویکرد همان فراخوانی یک اندپوینت RESTful می‌باشد. به این ترتیب، درخواست دریافت می‌شود و بی‌درنگ پردازش آن شروع و نتیجه بازگردانده می‌شود. دوستان شرکت پژوهش و توسعه ترجیح دادند در این مرحله از این رویکرد استفاده نمایند. این نوع پیاده‌سازی، از اولین نسخه نصب شده بر روی سرورهای فرماندهی انتظامی وجود داشته و داکیومنت‌های مورد نیاز در اختیار دوستان قرار گرفت.

۱-۲-۸ هشتمین جلسه (۱۴۰۰-۰۹-۲۴): تغییرات برای دسترسی

در این جلسه دو هدف دنبال گردید. اول آنکه، به دلیل انتقال سرور به مکان جدید، نیاز به انجام تنظیمات جدید بر روی سرور بود که به صورت کامل انجام گردید. هدف دوم، تعامل با همکاران فرماندهی انتظامی برای اعمال تغییرات برروی Gateway نرم‌افزاری سمت فرماندهی انتظامی بود تا امکان دیده شدن سرویس‌های سامانه مجری برای سامانه‌های شرکت پژوهش و توسعه را فراهم آوردند که این امر هم انجام شد (مکان: ساختمان فرماندهی انتظامی کل جمهوری اسلامی، میدان عطار).

۱-۲-۹ نهمین جلسه (۱۴۰۰-۱۰-۰۶): نصب نسخه جدید دموی تحت وب

در این جلسه، نسخه جدیدی از وب‌سایت دمو بر روی سرور بارگزاری شد. در این نسخه، بهبودهایی در رابط کاربری آن ایجاد شد (مکان: ساختمان فرماندهی انتظامی کل جمهوری اسلامی، میدان عطار).

۱-۲-۱۰ دهمین جلسه (۱۴۰۰-۱۱-۰۳): بررسی مشکل فرمت تصاویر

کارشناسان فنی شرکت پژوهش و توسعه در هنگام ارسال برخی تصاویر برای بازشناسی مشکل داشتند. با حضور در محل شرکت پژوهش و توسعه و بررسی تصاویر، مشخص گردید فرمت تصاویر ذخیره شده در پایگاه داده فرماندهی انتظامی مشکل داشته است. بعد از مشخص شدن مشکل تصاویر در پایگاه داده شرکت پژوهش و توسعه و رفع آن، مشکل دیگری مشاهده نگردید. (مکان: ساختمان فرماندهی انتظامی کل جمهوری اسلامی واقع در میدان هفتم تیر).

۱-۲-۱۱ یازدهمین جلسه (۱۴۰۰-۱۱-۱۳): نصب و تحویل سورس کد برای تست امنیتی

این جلسه به جهت نصب نسخه‌ای همراه با سورس کد از سامانه برای دوستان فرماندهی انتظامی بود تا تست‌های امنیتی بر روی سامانه انجام شود. بعد از نصب نسخه‌ای از سامانه، تمام جزئیات پیاده‌سازی و کدهای سامانه به دقت بررسی و تحویل گردید.

در ادامه به سوال‌های امنیتی کارشناسان فنی، نظیر نحوه احراز هویت درخواست‌ها در سامانه پاسخ داده شد و این قسمت از کد سامانه با حساسیت دوچندان مورد بررسی قرار گرفت (مکان: ساختمان حکمت).

۱ -۲ -۱۲ دوازدهمین جلسه (۱۶-۱۱-۱۴۰۰): نصب نسخه جدید هسته هوش مصنوعی

با توجه به نیازمندی جدید دوستان فرماندهی انتظامی، نسخه‌ای اختصاصی و بهبود یافته توسعه گردید. در این جلسه، نسخه سرور به روزرسانی شد و پس از تست، تحویل دوستان فرماندهی انتظامی گردید (مکان: ساختمان فرماندهی انتظامی کل جمهوری اسلامی، میدان عطار).

فصل ۲ راهنمای فعالسازی خدمات بر روی سامانه‌های مختلف

برای استفاده از سرویس‌های احراز هویت نیاز است تا سایر سامانه‌ها از طریق شبکه به سرور سامانه احراز هویت دسترسی داشته باشند و در ادامه با اختصاص یک توکن اختصاصی، می‌توانند از خدمات سامانه احراز هویت استفاده نمایند.

در هنگام فراخوانی سرویس‌ها، احراز هویت بر اساس یک توکن ایستا به نام ApiToken انجام می‌شود. این توکن در برنامه gateway به ازای هر کاربر فقط یکبار با طول عمر نامتنهی تعریف می‌شود. به این ترتیب به ازای هر سامانه‌ای در سمت کارفرمای بهره‌بردار که از این سرویس‌ها استفاده می‌نماید، یک توکن تعریف شده و در هنگام فراخوانی سرویس‌ها در هدر درخواست مقداردهی می‌شود. این توکن بر اساس استاندارد JWT ساخته می‌شود و برای تعریف توکن‌های جدید برای سامانه‌های مختلف، باید از اندپونت زیر در برنامه gateway که بر روی پورت ۱۱۰۰۰ قرار داده شده است، استفاده نمود. در این اندپوینت نیاز به یک توکن ادمین سیستم است که برنامه تشخیص دهد درخواست از سمت فرد قابل اطمینانی ارسال شده است و در جواب یک توکن جدید ایجاد کرده و به عنوان جواب برمی‌گرداند (برای اطلاعات بیشتر، سوگر آن در آدرس زیر چک شود:

“https://[ServerAddress:11000]/swagger/index.html”

POST api/account/new-token

فصل ۳ نصب، پیکربندی، اجرا، کاربری، راهبری و پشتیبانی

این فصل شامل توضیحاتی در مورد نحوه نصب و پیکربندی، فعال سازی، اجرا و کاربری و در نهایت راهبری و پشتیبانی می‌باشد.

۳-۱ نصب و پیکربندی

در ادامه این بخش به تشریح پیکربندی، معماری، ساختار و نحوه نصب سامانه بر روی سرور پرداخته شده است. لازم به ذکر است که فایل‌های مورد نیاز که در ادامه به آن‌ها اشاره شده است، در اختیار تیم بهره بردار قرار گرفته است.

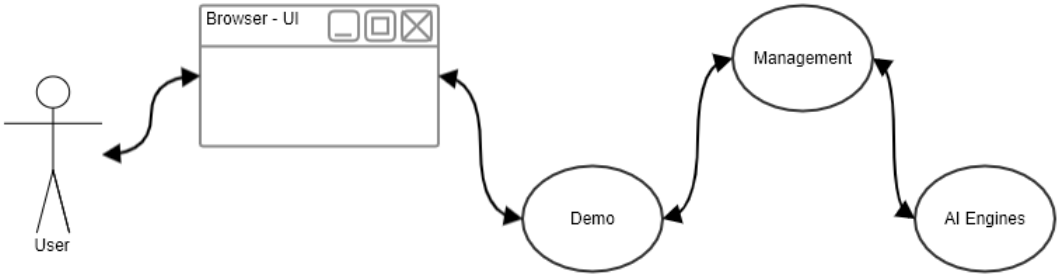
۳-۱-۱ سرور مورد نیاز

معماری سامانه برای نصب بر روی سرور بر اساس داکر است. به این ترتیب هر کدام از برنامه‌های مورد نیاز در قالب یه فایل ایمیج داکر خروجی گرفته شده و بر روی سرور اجرار می‌شوند. به دلیل آنکه از ایمیج‌های لینوکسی استفاده شده است، نیاز است تا سیستم عامل سرور بر اساس یکی از توزیع‌های لینوکس باشد و بر روی آن آخرین نسخه داکر نصب شود.

۳-۱-۲ معماری پروژه

ساختار پروژه به ترتیب دارای سه لایه هوش مصنوعی، مدیریت و دمو می‌باشد. در قسمت هوش مصنوعی مسائل مربوط به پردازش تصاویر و ویدئوها است که به لایه مدیریت سرویس می‌دهند. در لایه مدیریت، مسائل مربوط به کنترل میزان مصرف، بررسی سطح دسترسی بررسی می‌شود. همانطور که از اسم لایه دمو برداشت می‌شود، این لایه برای ارائه دمو و آزمون سیستم

توسط کاربر نهایی می‌باشد و برای تست توسط کارفرما پیاده‌سازی شده است. در بخش‌های بعدی به تشریح هر کدام از لایه‌ها پرداخته شده است.



شکل ۳–۱ لایه‌های یک سامانه احراز هویت غیرحضوری

۳ -۱ -۳ لایه هوش مصنوعی

این لایه شامل مدل‌های هوش مصنوعی و الگوریتم‌های مربوطه برای تشخیص هویت می‌باشد و از بیرون سیستم قابل دسترسی نبوده و صرفا با لایه مدیریت در تعامل می‌باشد. این لایه شامل مجموعه از فایل‌های داکر می‌باشد که به ترتیب با دستورات زیر اجرا می‌شوند. تمام فایل‌ها و ایمیج‌های مورد نیاز در قالب یک پوشه در سرور قرار داده شده‌اند و برای نصب بر روی سرورهای جدید نیاز است تا یک نسخه از آن بر روی سرور جدید بارگذاری شود. این پوشه شامل سه فایل زیر است:

- فایل load-images.sh
- فایل docker-compose.yml
- فایل‌هایی با پسوند sepidsystem-dide-services با حجم حداکثر ۵۱۲ مگابایت (ایمیج‌های ذخیره‌شده داکر)

۳ -۱ -۳ -۱ اجرای فایل load-images.sh و دریافت کلید

به دلیل اهمیت لایه هوش مصنوعی، روال فعال‌سازی برای آن در نظر گرفته شده است. این روال به ازای نصب بر روی هر سرور باید انجام و لایه هوش مصنوعی بر روی آن سرور فعال شود. به این منظور با استفاده از دستورات زیر یک کلید با فرمت xxx-xxx-xxx ایجاد می‌شود که باید در اختیار تیم پشتیبانی قرار گرفته تا با توجه به آن یک کد فعال‌سازی تولید شود. کد فعال‌سازی باید در فایل sepidsystem-dide.lic در روت سرور ذخیره شود.

```
chmod +x load-images.sh
/.load-images.sh
```

با انجام روال فوق، فعال‌سازی سیستم تکمیل می‌شود و فقط نیاز به اجرای برنامه می‌باشد که با دستور زیر برنامه اجرا می‌شود. پورت پیش فرض این لایه ۵۰۰۰ است و نیازی به باز کردن این پورت در سرور نمی‌باشد.

```
docker-compose up -d
```

۳ -۱ -۴ لایه دمو

این لایه برای ارائه دمو پیاده‌سازی شده است که شامل یک اپلیکیشن بک‌اند برای ارائه سرویس و یک اپلیکیشن فرانت‌اند برای نمایش نحوه کارکرد سیستم می‌باشد. از اپلیکیشن بک‌اند به این دلیل استفاده شده است که برای فراخوانی سرویس‌های هوش مصنوعی نیاز به شناسه یکتای ApiToken می‌باشد که عبارتی محرمانه است و نمی‌توان آن را در برنامه سمت کاربر ذخیره نمود. به این ترتیب، یک برنامه بک‌اند توسعه داده شده است که درخواست‌های سمت کاربر را بدون احراز هویت دریافت می‌نماید و در پس زمینه برنامه، شناسه یکتا را به درخواست اضافه کرده و برای سرویس‌های لایه مدیریت ارسال می‌نماید. همچنین برای راحتی تست، نیاز به آن بود تا برخی تصاویر به صورت پیش‌فرض در سیستم ذخیره شده باشند تا برای آزمودن سیستم از آن‌ها استفاده شود. لایه بک‌اند بر اساس C# ASP.NET Core و تکنولوژی‌های مرتبط توسعه داده شده است و پورت مورد نیاز APIهای این لایه ۱۲۰۰۰ است و در صورتی که نیاز به ارائه دمو باشد، این پورت نیز باید بر روی سرور باز شود.

برنامه سمت فرانت نیز برای ایجاد یک وب سایت که امکان تست سرویس‌های تطبیق چهره، تشخیص زنده بودن غیر تعاملی و تعاملی (با حالت‌های مختلف) را می‌هد، توسعه داده شده است. این برنامه با React توسعه داده شده است و در صورت تغییر آی‌پی و یا پورت سرویس‌های بک‌اند، باید آدرس‌های جدید در فایل زیر به روزرسانی شوند.

/front/src/config/baseUrl.json

برنامه‌های بک‌اند و فرانت‌اند فوق در هنگام اجرای برنامه‌های سمت لایه مدیریت به صورت خودکار اجرا می‌شوند.

۳ -۱ -۵ لایه مدیریت

این لایه برای ارائه سرویس‌های هوش مصنوعی به سرویس‌های خارج از سیستم می‌باشد. وظایفی همچون احراز هویت و سنجش میزان استفاده از سرویس‌ها در این لایه انجام شود. این لایه از ترکیب چند داکر ایمیج می‌باشد که کد تمام ایمیج‌ها در پوشه app قرارداده شده‌اند. ساختار پوشه app به صورت زیر است:

- پوشه accounting: در این پوشه کد مربوط به محاسبه میزان مصرف قرار دارد و تمام درخواست‌ها از این برنامه عبور کرده و برای سرویس‌های لایه هوش مصنوعی ارسال می‌شوند. در صورت تغییر آدرس سرور لایه هوش مصنوعی، نیاز است تا آدرس جدید در فایل زیر و در متغییر EkycCoreUrl جایگزین شود.

  Accounting/Sepid.EKYC.Api/appsettings.json

- پوشه gateway: در این پوشه کد مربوط به کنترل دسترسی قرار دارد و تمام درخواست‌ها توسط این برنامه دریافت شده و برای accounting ارسال می‌شوند. پورت ۱۱۰۰۰ برای دسترسی به این برنامه است که در صورت باز بودن می‌توان سرویس‌های مورد نظر را با آدرس زیر مشاهده نمود. این برنامه برای کنترل دسترسی از ApiToken ارسال شده به همراه درخواست استفاده می‌نماید که در بخش‌های بعدی به تشریح آن پرداخته شده است.

  https://[ServerAddress:11000]/swagger/index.html

- پوشه demo: در این پوشه کد مربوط به بک‌اند برنامه دمو قرار داشته که در بخش لایه دمو به تشریح آن پرداخته شده است. این برنامه از پورت ۱۲۰۰۰ استفاده می‌نماید و صرفا جهت دمو است و می‌توان از برنامه حذف نمود. برای مشاهده سرویس‌های این لایه می‌توان از آدرس زیر استفاده نمود.

https://[ServerAddress:12000]/swagger/index.html

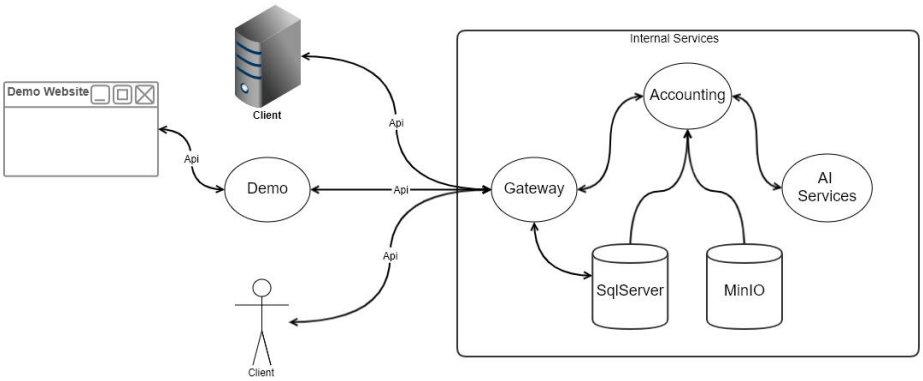
- پوشه front: در این پوشه کد مربوط به فرانت‌اند برنامه دمو قرار داشته که در بخش لایه دمو به تشریح ان پرداخته شده است. این برنامه صرفا جهت دمو است و می‌توان از برنامه حذف نمود.
- پوشه minio-data: برای ذخیره عکس و ویدئوهای مورد نیاز در لایه مدیریت از پایگاه داده MinIO استفاده شده است. در هنگام اجرای این برنامه، محل ذخیر عکس و ویدئوها در این پوشه است. برای تغییر محل ذخیره سازی باید آدرس آن در فایل docker-compose.yml تغییر نماید.
- پوشه sql-data: برای ذخیره داده‌های مورد نیاز سیستم از پایگاه داده Microsoft SQL Server استفاده شده است. در هنگام اجرای برنامه، محل ذخیره سازی داده‌ها این پوشه است. برای تغییر محل ذخیره سازی باید آدرس آن در فایل docker-compose.yml تغییر نماید.
- فایل docker-compose.yml: در این فایل همه تنظیمات مربوط به ساخت و اجرای برنامه قرار داده شده است. برای اتصال برنامه‌های ذکر شده در بالا، یک شبکه مجازی به نام naja-net در داخل داکر ایجاد شده است که باعث می‌شود برنامه‌های بالا یکدیگر را دیده ولی از بیرون از سرور و حتی بیرون این شبکه مجازی در داخل سرور مشاهده نشوند.

این لایه نیاز به فعال‌سازی نداشته و بر اساس C# ASP.NET Core و تکنولوژی‌های مرتبط توسعه داده شده. برای استفاده از سرویس‌های هوش مصنوعی باید از برنامه gateway استفاده شود که بر روی پورت ۱۱۰۰۰ قابل مشاهده هست. همچنین برای مشاهده لیست همه سرویس‌ها می‌توان از آدرس زیر استفاده نمود.

https://[ServerAddress:11000]/swagger/index.html

برای اجرای این لایه باید از دستورات زیر به ترتیب استفاده نمود.

```
docker-compose build
docker-compose up -d
```



شکل ۳–۲ معماری سامانه احراز هویت غیرحضوری

۳- ۲ اجرا و کاربری

همانطور که در فصل‌های گذشته نیز اشاره گردید، در هنگام فراخوانی سرویس‌ها، احراز هویت بر اساس یک توکن ایستا به نام ApiToken انجام می‌شود. این توکن در برنامه gateway به ازای هر کاربر فقط یکبار با طول عمر نامتنهی تعریف می‌شود. به این ترتیب به ازای هر سامانه‌ای در سمت کارفرمای بهره‌بردار که از این سرویس‌ها استفاده می‌نماید، یک توکن تعریف شده و در هنگام فراخوانی سرویس‌ها در هدر درخواست مقداردهی می‌شود. این توکن بر اساس استاندارد JWT ساخته می‌شود و برای تعریف توکن‌های جدید برای سامانه‌های مختلف، باید از اندپونت زیر در برنامه gateway که بر روی پورت ۱۱۰۰۰ قرار داده شده است، استفاده نمود. در این اندپوینت نیاز به یک توکن ادمین سیستم است که برنامه تشخیص دهد درخواست از سمت فرد قابل اطمینانی ارسال شده است و در جواب یک توکن جدید ایجاد کرده و به عنوان جواب برمی‌گرداند (برای اطلاعات بیشتر، سوگر آن در آدرس https://[ServerAddress:11000]/swagger/index.html چک شود).

POST api/account/new-token

در فصل اول به تشریح نحوه استفاده از سرویس‌های سامانه پرداخته شده است.

۳- ۳ راهبری و پشتیبانی

پیاده‌سازی سامانه به صورت رفع اشکال خودکار می‌باشد، به این ترتیب در صورت بروز مشکل و از دسترس خارج شدن هر کدام از بخش‌های سامانه، عملیات بازیابی انجام شده و نسخه‌ای جدید از آن بخش سامانه بارگزاری می‌شود. در صورتی این فرایند خودکار به درستی انجام نشود، به داخل دو پوشه هوش مصنوعی و اپ رفته و به ترتیب دستورات زیر اجرا شود (در بخش قبلی به تشریح محتوای این پوشه‌ها پرداخته شده است).

```
docker-compose down
docker-compose up -d
```

فصل ۴ تحلیل مخاطرات

با توجه به عدم قطعیت‌های محیطی و ویژگی‌های منحصربه‌فرد پروژه‌ها، مدیریت مخاطرات یک ضرورت غیرقابل اجتناب بوده و از درجه اهمیت بالایی برخوردار است. به منظور مدیریت مخاطرات پروژه، فعالیت‌های زیر در راستای اهداف پروژه انجام شده است.

- شناسایی مخاطرات: تعیین مخاطرات‌هایی که ممکن است بر پروژه اثر بگذارند.
- ارزیابی مخاطرات: الویت‌بندی مخاطرات‌ها بر اساس اهمیت و تاثیر آن‌ها بر پروژه.
- پاسخگویی به مخاطرات: تهیه‌ی رویه‌ها و تکنیک‌هایی جهت افزایش فرصت‌ها و کاهش تهدیدها.

در ادامه مخاطرات پروژه در دو دسته‌ی عملیاتی کردن سامانه و به کارگیری آن بررسی و راه‌حل‌های پیشنهادی ارائه می‌گردد.

۴-۱ مخاطرات مربوط به عملیاتی کردن سامانه

۴-۱-۱ خرابی در تجهیزات ارتباطی یا دیتاسنتر و از دسترس خارج شدن سامانه

با توجه به این که سامانه پردازش مرکزی به صورت ابری و در سروری جداگانه درحال اجراست، ممکن است به علت مشکلات پیش‌بینی نشده مانند خرابی قطعات، اختلالات اینترنت یا قطعی برق در دسترس نباشد.

راه‌حل پیشنهادی: این دست از مخاطرات می‌بایست با داشتن یک برنامه برای بازیابی از حادثه (DRP) به دقت توسط کارفرما مورد بررسی قرار بگیرد و استراتژی‌های بازیابی مخصوصی برای هریک از حالت‌های احتمالی در نظر گرفته شود. برخی از این

مشکلات را می‌توان با راهکارهایی مانند داشتن سرورهای پشتیبان به صورت آینه‌ای و ارسال درخواست‌ها به سرور پشتیبان به جای سرور اصلی در مواقع نیاز رفع کرد. کارفرما نیاز است با بررسی سناریوهای مختلف و قرار دادن سامانه‌های پشتیبان در دیتاسنترهای مختلف دسترس بودن سامانه را به حداکثر برساند.

۴ -۱ -۲ تعداد درخواست‌های ورودی سامانه بیش از توان پردازشی سخت‌افزار

در عمل هنگام استفاده از این سامانه با توجه به سخت‌افزاری که سامانه روی آن اجرا می‌شود، سقفی برای تعداد درخواست همزمان تعریف می‌شود. در صورتی که تعداد درخواست همزمان برای ساعات پیک بیش از توان پردازشی سرور باشد، زمان پاسخ‌دهی به درخواست‌ها افزایش خواهد یافت و تاثیر آن مستقیما برای کاربران نهایی قابل مشاهده است.

راه‌حل پیشنهادی: با تخمین هرچه دقیق‌تر تعداد کاربرانی که از این سامانه استفاده می‌کنند و در نظر گرفتن ضریب اطمینان، سخت‌افزاری تامین شود که توانایی پاسخ‌دهی به حداکثر درخواست‌های همزمان را داشته باشد. یا با معماری توزیع‌شده در صورت لزوم امکان اختصاص سخت‌افزار بیشتر تامین شود.

۴ -۱ -۳ ایجاد بار غیرواقعی برای سامانه توسط برنامه‌های مخرب

برنامه‌های مخرب ممکن است با ارسال درخواست‌های غیرواقعی به سمت سامانه، ترافیک غیرواقعی ایجاد کنند و باعث کند شدن و یا از کار افتادن سامانه شوند.

راه‌حل پیشنهادی: سامانه می‌بایست به گونه‌ای پیاده‌سازی شود که ابزارهایی برای کشف و خنثی‌سازی درخواست‌های غیرواقعی محیا شود. به عنوان مثال اپلیکیشنی که توسط آن درخواست‌های احراز هویت ارسال می‌شود برای ارسال درخواست نیازمند تایید کپچا باشد و یا در لایه‌های اولیه پردازش درخواست، IP هایی که از آن‌ها درخواست‌های متعددی ارسال شده است بلاک شوند و درخواست‌های آن‌ها پردازش نشود.

۴ -۱ -۴ نفوذ به سرور و دسترسی و یا دستکاری اطلاعات موجود در پایگاه‌داده

در صورت در نظر نگرفتن استانداردهای امنیتی در بستری که سامانه نصب می‌شود، ممکن است دسترسی غیرمجاز به ماشینی که سامانه روی آن درحال اجراست، صورت بگیرد. در این صورت با وجود این که اطلاعات در پایگاه داده به صورت ایمن و استاندارد ذخیره می‌شود اما امکان خرابکاری در سامانه یا دسترسی غیرمجاز به داده‌ها محتمل خواهد بود.

راه‌حل پیشنهادی: بستری که بر روی آن سامانه عملیاتی و اجرایی می‌شود لازم است که با قرار دادن لایه‌های امنیتی استاندارد مانند فایروال، امکان دسترسی و درز اطلاعات راجع به سروری مرکزی را تا حد امکان پایین ببرد.

۴- ۲ مخاطرات مربوط نحوه به‌کارگیری سامانه

۴ -۲ -۱ استفاده از سطح سخت‌گیری نامناسب با توجه به کاربرد

سرویس‌های ارائه شده توسط سامانه همگی یک ورودی به عنوان «سطح سخت‌گیری» نیز دارند (جزئیات چگونگی استفاده در سند راهنمای APIها آورده شده است). تنظیم درست سطح سخت‌گیری با توجه به حساسیت کاربرد مورد نظر به عهده سازمان به‌کارگیرنده سامانه می‌باشد. تنظیم سطح سخت‌گیری پایین‌تر یا بالاتر از چیزی که نیاز است، می‌تواند خطاهای ناخواسته برای سازمان به‌کارگیرنده به همراه داشته باشد.

راه‌حل پیشنهادی: با در نظر گرفتن دقت سامانه در هریک از سطوح سخت‌گیری، حساسیت سامانه پیش از عملیاتی شدن آن تنظیم شود و مورد آزمون قرار بگیرد و با پایش وضعیت خروجی‌های سامانه در صورت نیاز تغییر پیدا کند.

۴ -۲ -۲ استفاده از روش تشخیص زنده بودن نامناسب

همانطور که در مستندات سامانه ذکر شده، تشخیص زنده بودن به صورت غیرتعاملی به تنهایی یا تشخیص زنده بون تعاملی به تنهایی معمولا دقت مناسبی به همراه ندارد. به همین علت استفاده از روش‌های ترکیبی که در آن هم بررسی غیرتعاملی و هم تعاملی انجام می‌شود، توصیه شده است.

به عنوان مثال در یکی از سرویس‌های سامانه دو تصویر چهره از فرد با یکدیگر تطبیق داده می‌شوند و بررسی زنده بودن غیرتعاملی بر روی یکی از آن دو (که از دوربین دریافت شده است) صورت می‌گیرد. توجه داشته باشید که این بررسی زنده بودن غیرتعاملی نمی‌تواند دقت بررسی زنده بودن سرویس ترکیبی غیرتعاملی و تعاملی (تشخیص گفتار / پلک زدن) را داشته باشد.

راه‌حل پیشنهادی: در صورتی که کاربرد سازمان به کارگیرنده از حساسیت بالایی برخوردار است، به هیچ عنوان از تشخیص زنده بون تعاملی یا غیرتعاملی به تنهایی استفاده نشود. سرویس‌های ترکیبی که هردو مورد در آن‌ها بررسی می‌شوند با دقت بالاتر در اختیار قرار گرفته است.