



موسسه آموزش عالی صدرالمتالهین (صدر)
دانشکده کامپیوتر

پایان نامه کارشناسی ارشد
رشته مهندسی کامپیوتر گرایش نرم افزار

عنوان
ارائه یک روش کارآمد در نرم افزار دستگاه پزشکی **Viewer** جهت مانیتورینگ و
ذخیره سازی علائم حیاتی بیمار

نگارش
سپهر کریمی صدیق

استاد راهنما
سرکار خانم دکتر زهرا عسکری نژاد امیری

۱۴۰۱ مهر

الله اکبر

بسمه تعالی

اظهارنامه دانشجو

اینجانب سپهر کریمی صدیق دانشجوی کارشناسی ارشد رشته مهندسی کامپیوتر گرایش نرم افزار

دانشکده کامپیوتر - موسسه آموزش عالی صدرالمتألهین (صدر) گواهی می نمایم که تحقیقات

ارائه شده در پایان نامه با عنوان " ارائه یک روش کارآمد در نرم افزار دستگاه پزشکی Viewer جهت

مانیتورینگ و ذخیره سازی علائم حیاتی بیمار " با راهنمای استاد محترم سرکار خانم دکتر زهرا

عسکری نژاد امیری توسط شخص اینجانب انجام شده و صحت و اصالت مطالب نگارش شده در

این پایان نامه مورد تایید می باشد و در مورد استفاده از کار دیگر محققان به مراجع مورد استفاده

ارجاع گردیده است. به علاوه گواهی می نمایم که مطالب مندرج در پایان نامه تا کنون برای دریافت

هیچ نوع مدرک یا امتیازی توسط اینجانب یا فرد دیگری در هیچجا ارائه نشده است و در تدوین متن

پایان نامه چهار چوب (فرمت) مصوب دانشگاه را به طور کامل رعایت کرده ام.

تاریخ و امضای دانشجو: ---/---/----

بسمه تعالی

حق طبع و نشر و مالکیت نتایج

حق چاپ و تکثیر این پایان‌نامه متعلق به نویسنده آن و موسسه آموزش عالی صدرالمتألهین

(صدر) می‌باشد. هرگونه کپی برداری بصورت کل پایان‌نامه یا بخشی از آن تنها با موافقت

نویسنده یا کتابخانه موسسه آموزش عالی صدرالمتألهین (صدر) مجاز می‌باشد. ضمناً متن این

صفحه نیز باید در نسخه تکثیر شده وجود داشته باشد.

کلیه حقوق معنوی این اثر متعلق به موسسه آموزش عالی صدرالمتألهین می‌باشد و بدون

اجازه کتبی موسسه به شخص ثالث قابل واگذاری نیست.

استفاده از اطلاعات و نتایج موجود در پایان نامه بدون ذکر مراجع مجاز نمی‌باشد.



به نام خدا

تعهدنامه اصالت اثر

تاریخ:

اینجانب سپهر کریمی صدیق متعهد می‌شوم که مطالب مندرج در این پایان نامه حاصل کار پژوهشی اینجانب تحت نظرارت و راهنمایی استادی موسسه آموزش عالی صدرالمتألهین 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 مأخذ ذکر گردیده است. این پایان نامه قبلاً برای احراز هیچ مدرک هم‌سطح یا بالاتر ارائه نگردیده است.

در صورت اثبات تخلف در هر زمان، مدرک تحصیلی صادر شده توسط موسسه از درجه اعتبار ساقط بوده و موسسه حق پیگیری قانونی خواهد داشت.

کلیه نتایج و حقوق حاصل از این پایان نامه متعلق به موسسه آموزش عالی صدرالمتألهین می‌باشد. هرگونه استفاده از نتایج علمی و عملی، واگذاری اطلاعات به دیگران یا چاپ و تکثیر، نسخه‌برداری، ترجمه و اقتباس از این پایان نامه بدون موافقت کتبی موسسه آموزش عالی صدرالمتألهین ممنوع است. نقل مطالب با ذکر مأخذ بلامانع است.

سپهر کریمی صدیق

امضا

تقدیم

ماحصل آموخته‌هایم را تقدیم می‌کنم به آنان که مهر آسمانی‌شان آرام بخش آلام زمینی‌ام است...

به سبزترین نگاه زندگیم، چشمان پرمه‌ر مادرم...

به استوارترین تکیه گاهم، دستان پرمه‌ر پدرم...

که هرچه آموختم در مکتب عشق شما آموختم و هرچه بکوشم قطره‌ای از دریای بی‌کران مهربانی‌تان را سپاس نتوانم بگویم.

امروز هستی‌ام به امید شماست و فردا کلید باغ بهشتیم رضای شما.

آوردی گران سنگ‌تر از این ارزان نداشتیم تا به خاک پایتان نثار کنم، باشد که حاصل تلاشم نسیم گونه غبار خستگی‌تان را بزداید.

بوسه بر دستان پرمه‌ر تان

تقدیر و تشکر

به نام آفریدگار پاک که انسان را از خاک آفرید و به واسطه عقل بر تمام موجودات رجحان داد و آنگاه
دانش را وسیله تکامل عقل قرار داد. بزرگ پروردگار را سپاس که به ما استعداد آموختن عطا کرد.

از استاد فاضل و اندیشمند سرکار خانم دکتر زهرا عسکری‌نژاد امیری به عنوان استاد راهنمای با مشغله
فراوان قبول زحمت کردند و همواره اینجانب را مورد لطف و محبت خود قرار داده‌اند، کمال تشکر و
قدرتانی را دارم.

چکیده

امروزه، در دنیای کنونی و در عصر تکنولوژی، فناوری در اکثر صنایع نقش مهمی را ایفا میکند. یکی از این صنایع که فناوری در آن، بسیار مورد استفاده قرار گرفته، صنعت پزشکی است. در صنعت پزشکی انواع و اقسام دستگاهها با تکنولوژی‌های گوناگونی توسعه یافته است که هر کدام از این دستگاهها کاربرد مختص به خودش را دارد؛ مثلاً یک دستگاه وظیفه احیای بیمار از طریق کنترل نحوه تنفس را دارد و این کار را با دادن اکسیژن و کنترل دم و بازدم بیمار انجام می‌دهد، یا در مثالی دیگر، یک دستگاه وظیفه احیای بیمار را با دادن شوک به او دارد. خیلی از این دستگاهها باید باهم تعامل داشته باشند و به یکدیگر داده ارسال کنند. ارتباط این دستگاهها باهم امری پیچیده و دشوار است. همچنین به دلیل این که با جان انسان سروکار داریم، ارسال، دریافت و مدیریت داده‌های انتقالی مبحث بسیار مهمی است که در این پژوهش به آن پرداخته شده است تا در راستای احیای هرچه بهتر بیماران با کمک فناوری، گام مهمی را در صنعت پزشکی برداشته باشیم. در این پژوهش روش تحقیق شامل دو فاز تحقیقات و پیاده‌سازی بوده که یافته‌های علمی در فاز تحقیقات را با استفاده از زبان برنامه‌نویسی سی‌شارپ و تکنولوژی WPF، در فاز پیاده‌سازی به اجرا درآورده و آن را مورد ارزیابی قرار داده‌ایم. در این راستا نرمافزار دستگاه پزشکی Viewer جهت مانیتورینگ و ذخیره‌سازی علائم حیاتی بیمار توسعه یافته است که امکان ارتباط و تعامل با چهار دستگاه پزشکی دیگر را دارد و کاربر (پزشک، پرستار و ...) می‌تواند در قسمت‌های مختلف برنامه علائم حیاتی بیمار را مشاهده و بررسی کند.

واژه‌های کلیدی:

معماری نرمافزار، انتقال داده، دستگاه‌های پزشکی، برنامه‌نویسی، شبکه، نظارت بر مراقبت‌های بهداشتی، نظریه احیای بیمار، نرم‌افزار

صفحه

فهرست عناوین

۱	۱ مقدمه	۱
۷	۱.۱ بیان مسئله	
۸	۲.۱ ضرورت انجام تحقیق	
۹	۳.۱ سوالات تحقیق	
۱۰	۴.۱ اهداف تحقیق	
۱۰	۵.۱ روش تحقیق	
۱۱	۶.۱ معرفی فصول پایان نامه	
۱۱	۷.۱ تعاریف اصلی مورد استفاده	
۱۳	۸.۱ خلاصه فصل	
۱۴	۲ پیشینه تحقیق	۲
۱۵	۱.۲ مروری بر پژوهش های پیشین	
۲۵	۲.۲ جمع بندی و نتیجه گیری	
۲۶	۳.۲ خلاصه فصل	
۲۷	۳ روش تحقیق و داده ها	
۲۸	۱.۳ شرح دستگاه Viewer	
۳۰	۱.۱.۳ پروتکل ارتباطی با دستگاه آریا	
۳۱	۲.۱.۳ پروتکل ارتباطی با دستگاه ونتیلاتور	
۳۳	۳.۱.۳ پروتکل ارتباطی با مازول O3	
۳۴	۴.۱.۳ پروتکل ارتباطی با مازول سانترال	
۳۵	۲.۳ نیازمندی ها	
۳۷	Display Orientation ۱.۲.۳	
۴۰	Monitoring Aria's Modules ۲.۲.۳	
۴۲	Send And Receive Aria's Settings ۳.۲.۳	
۴۴	Customize Page ۴.۲.۳	
۴۷	Communication With Central ۵.۲.۳	
۴۹	Changing Network Setting By Password ۶.۲.۳	
۵۱	Storing Aria's Data ۷.۲.۳	
۵۳	Aria's Signals History ۸.۲.۳	
۵۶	Aria's Parameters History ۹.۲.۳	
۵۷	Aria's Parameters History ۱۰.۲.۳	
۶۰	Aria's Nibps History ۱۱.۲.۳	
۶۲	Aria's Arrhythmia History ۱۲.۳.۲	

۶۴	Taking Screenshot	۱۳.۲.۳
۶۶	Connection With Ventilator	۱۴.۲.۳
۶۶	Connection With O3	۱۵.۲.۳
۶۷	۳.۳ ابزارهای مورد استفاده	
۶۹	۴.۳ معماری سیستم	
۷۴	۵.۳ معماری لایه UI	
۷۴	۶.۳ پیاده‌سازی و ارزیابی	
۷۶	۴ تجزیه و تحلیل داده‌ها	
۷۷	۱.۴ پیاده‌سازی	
۷۸	۱.۱.۴ پیاده‌سازی پروژه Ventilator	
۷۹	۲.۱.۴ پیاده‌سازی پروژه O3	
۸۱	۳.۱.۴ پیاده‌سازی پروژه HeartlandAriaTransfer	
۸۷	۲.۴ پیاده‌سازی لایه UI پروژه HeartlandAriaTransfer	
۹۴	۳.۴ تجزیه و تحلیل داده‌ها (ارزیابی و کارآیی)	
۹۷	۵ بحث و نتیجه‌گیری همراه با مطالعات آتی	
۹۸	۱.۵ یافته‌های پژوهش	
۹۹	۲.۵ محدودیت‌ها و مشکلات پژوهش	
۹۹	۳.۵ پیشنهادهایی در خصوص پژوهش‌های آتی	
۱۰۱	منابع و مراجع	
۱۰۴	پیوست‌ها	
۱۲۱	چکیده انگلیسی	

صفحه

فهرست اشکال

۲ شکل ۱.۱ Viewer و دستگاه‌ها جانبی
۵ شکل ۲.۱ کاتالوگ دستگاه مانیتور عالیم حیاتی شرکت Mindray
۶ شکل ۳.۱ کاتالوگ دستگاه Viewer
۱۶ شکل ۱.۲ اجزای معماری MVC و MVVM
۱۷ شکل ۲.۲ سایز داده‌ها انتقالی قبل و بعد از پردازش اولیه
۱۸ شکل ۳.۲ معماری طبقه‌بندی شده تفسیر داده
۱۹ شکل ۴.۲ مانیتورینگ در لحظه داده‌ها
۲۰ شکل ۵.۲ روش ارتباط با دستگاه جانبی
۲۱ شکل ۶.۲ مانیتورهای سری N شرکت Mindray
۲۲ شکل ۷.۲ سیگنال‌های نویزی
۲۴ شکل ۸.۲ گراف ساختاری کدهای نرم‌افزار
۲۹ شکل ۱.۳ نحوه تعامل Viewer با دستگاه‌ها جانبی
۳۷ شکل ۲.۳ R01-Display Orientation Use Case Diagram
۳۸ شکل ۳.۳ R01-Display Orientation State Machine Diagram
۳۸ شکل ۴.۳ R01-Display Orientation Sequence Diagram
۳۹ شکل ۵.۳ R01-Display Orientation Class Diagram
۳۹ شکل ۶.۳ R01-Display Orientation Activity Diagram
۴۰ شکل ۷.۳ R02-Monitoring Aria's Modules Use Case Diagram
۴۰ شکل ۸.۳ R02-Monitoring Aria's Modules Sequence Diagram
۴۱ شکل ۹.۳ R02-Monitoring Aria's Modules Deployment Diagram
۴۱ شکل ۱۰.۳ R02-Monitoring Aria's Modules Class Diagram
۴۲ شکل ۱۱.۳ R02-Monitoring Aria's Modules Activity Diagram
۴۲ شکل ۱۲.۳ R03-Send And Receive Aria's Settings Use Case Diagram
۴۳ شکل ۱۳.۳ R03-Send And Receive Aria's Settings Sequence Diagram
۴۳ شکل ۱۴.۳ R03-Send And Receive Aria's Settings Deployment Diagram
۴۴ شکل ۱۵.۳ R03-Send And Receive Aria's Settings Class Diagram
۴۴ شکل ۱۶.۳ R03-Send And Receive Aria's Settings Activity Diagram
۴۵ شکل ۱۷.۳ R04-Customize Page Use Case Diagram
۴۵ شکل ۱۸.۳ R04-Customize Page Sequence Diagram
۴۶ شکل ۱۹.۳ R04-Customize Page Class Diagram
۴۶ شکل ۲۰.۳ R04-Customize Page Activity Diagram
۴۷ شکل ۲۱.۳ R05-Communicate With Central Use Case Diagram

٤٨	R05-Communicate With Central Sequence Diagram	٢٢.٣
٤٨	شكل R05-Communicate With Central Class Diagram	٢٢.٣
٤٩	R05-Communicate With Central Activity Diagram	٢٤.٣
٤٩	شكل R06-Changing Network Setting By Password Use Case Diagram	٢٥.٣
٥٠	R06-Changing Network Setting By Password Sequence Diagram	٢٦.٣
٥٠	شكل R06-Changing Network Setting By Password Class Diagram	٢٧.٣
٥١	R06-Changing Network Setting By Password Activity Diagram	٢٨.٣
٥١	شكل R07-Storing Aria's Datas Use Case Diagram	٢٩.٣
٥٢	R07-Storing Aria's Datas Sequence Diagram	٣٠.٣
٥٢	شكل R07-Storing Aria's Datas Class Diagram	٣١.٣
٥٣	R07-Storing Aria's Datas Activity Diagram	٣٢.٣
٥٣	شكل R08-Aria's Signals History Use Case Diagram	٣٣.٣
٥٤	R08-Aria's Signals History Sequence Diagram	٣٤.٣
٥٤	شكل R08-Aria's Signals History Class Diagram	٣٥.٣
٥٥	R08-Aria's Signals History Activity Diagram	٣٦.٣
٥٦	R09-Aria's Parameters History Use Case Diagram	٣٧.٣
٥٦	شكل R09-Aria's Parameters History Sequence Diagram	٣٨.٣
٥٧	R09-Aria's Parameters History Class Diagram	٣٩.٣
٥٧	شكل R09-Aria's Parameters History Activity Diagram	٤٠.٣
٥٨	R10-Aria's Alarms History Use Case Diagram	٤١.٣
٥٨	شكل R10-Aria's Alarms History Sequence Diagram	٤٢.٣
٥٩	R10-Aria's Alarms History Class Diagram	٤٣.٣
٥٩	شكل R10-Aria's Alarms History Activity Diagram	٤٤.٣
٦٠	R11-Aria's Nibps History Use Case Diagram	٤٥.٣
٦٠	شكل R11-Aria's Nibps History Sequence Diagram	٤٦.٣
٦١	R11-Aria's Nibps History Class Diagram	٤٧.٣
٦١	شكل R11-Aria's Nibps History Activity Diagram	٤٨.٣
٦٢	R12-Aria's Arrhythmia History Use Case Diagram	٤٩.٣
٦٢	شكل R12-Aria's Arrhythmia History Sequence Diagram	٥٠.٣
٦٣	R12-Aria's Arrhythmia History Class Diagram	٥١.٣
٦٣	شكل R12-Aria's Arrhythmia History Activity Diagram	٥٢.٣
٦٤	R13-Taking Screenshot Use Case Diagram	٥٣.٣
٦٤	شكل R13-Taking Screenshot Sequence Diagram	٥٤.٣

۶۵.....	R13-Taking Screenshot Class Diagram	۵۵.۳
۶۵.....	R13-Taking Screenshot Activity Diagram	۵۶.۳
۶۶.....	R14-Connection with Ventilator State Machine Diagram	۵۷.۳
۶۷.....	R15-Connection with O3 State Machine Diagram	۵۸.۳
۷۰.....	شکل ۵۹.۳ معماری نرمافزاری	
۷۸.....	شکل ۱.۴ معماری نرمافزاری پروژه Ventilator	
۷۹.....	شکل ۲.۴ معماری نرمافزاری لایه UI پروژه Ventilator	
۸۰.....	شکل ۳.۴ معماری نرمافزاری پروژه O3	
۸۰.....	شکل ۴.۴ معماری نرمافزاری لایه UI پروژه O3	
۸۱.....	شکل ۵.۴ معماری نرمافزاری پروژه HeartlandAriaTransfer	
۸۲.....	شکل ۶.۴ کد مربوط به باز کردن پورت نرمافزار	
۸۳.....	شکل ۷.۴ کد مربوط به خواندن و نوشتن دادهها	
۸۳.....	شکل ۸.۴ کد مربوط به تفسیر داده	
۸۴.....	شکل ۹.۴ کد مربوط به مدیریت دادهها	
۸۵.....	شکل ۱۰.۴ کد مربوط به هسته مرکزی معماری نرمافزار	
۸۶.....	شکل ۱۱.۴ لایه بندی بخش‌های مربوط به سانترال	
۸۶.....	شکل ۱۲.۴ کد مربوط به سانترال	
۸۷.....	شکل ۱۳.۴ نحوه ارجاع به نرمافزارها جانبی	
۸۸.....	شکل ۱۴.۴ لیست وابستگی‌ها	
۸۹.....	شکل ۱۵.۴ ساختار صفحه نرمافزار	
۹۰.....	شکل ۱۶.۴ صفحه اول نرمافزار	
۹۰.....	شکل ۱۷.۴ صفحه دوم نرمافزار	
۹۱.....	شکل ۱۸.۴ صفحه سوم نرمافزار	
۹۱.....	شکل ۱۹.۴ صفحه چهارم نرمافزار	
۹۲.....	شکل ۲۰.۴ صفحه پنجم نرمافزار	
۹۲.....	شکل ۲۱.۴ صفحه ششم نرمافزار	
۹۳.....	شکل ۲۲.۴ صفحه هفتم نرمافزار	
۹۳.....	شکل ۲۳.۴ صفحه هشتم نرمافزار	
۹۴.....	شکل ۲۴.۴ صفحه نهم نرمافزار	

صفحه

فهرست جداول

۳۱.....	جدول ۱.۳ ساختار بسته آریا
۳۱.....	جدول ۲.۳ ساختار بسته ونتیلاتور
۳۲.....	جدول ۳.۳ ساختار بخش داده بسته ونتیلاتور
۳۲.....	جدول ۴.۳ کدهای بسته ونتیلاتور
۳۳.....	جدول ۵.۳ ساختار بسته O3
۳۴.....	جدول ۶.۳ ساختار بخش داده بسته O3
۳۴.....	جدول ۷.۳ ساختار بسته سانترال
۳۵.....	جدول ۸.۳ سند URS نرم افزار

۱

فصل اول

مقدمه

مقدمه

دستگاه‌های حوزه پزشکی از طریق حسگرهایی^۱ که به بیمار متصل می‌شوند، داده‌هایی را که مشخص کننده وضعیت بیمار است را دریافت می‌کنند و بعضی آنها را نمایش می‌دهند؛ سپس از طریق کابل، به دستگاه Viewer متصل^۲ می‌شوند و این داده‌ها را نیز به Viewer ارسال می‌کنند. انواع دستگاه‌های Error! No text of Viewer را نیز به دارند، عبارتنند از (شکل ۱) specified style in document.

:۱)

- آریا^۳

- ونتیلاتور^۴

- O3^۵

- سانترال^۶ یا سیستم^۷ مانیتورینگ مرکزی



^۱ Pads

^۲ اتصالات تمامی دستگاه‌های پزشکی جانبی از طریق کابل است.

^۳ Aria Bedside

^۴ Ventilator

^۵ نام یکی از دستگاه پزشکی، ساخت شرکت آمریکایی Masimo

^۶ Central

^۷ System

شکل اول: Error! No text of specified style in document. و دستگاهها جانبی.

برای پزشکان نظارت و کنترل علائم حیاتی بیمار از طریق دستگاه Viewer راحت‌تر است چون که به پزشکان این امکان داده می‌شود تا علائم حیاتی که حاصل از کار چندین دستگاه است را به صورت همزمان، فقط در یک دستگاه و در صفحه نمایش بزرگ‌تر نظارت کنند. علاوه‌براین، دستگاه Viewer می‌تواند با فاصله از تخت بیمار قرار بگیرد و برای موقوعی که امکان نزدیک شدن به بیمار میسر نیست، مثل اپیدمی کرونا^۱ و ... ضروری است که نظارت با فاصله از بیمار انجام شود.

در واقع سیستم Viewer یک نوع مانیتورینگ^۲ پیشرفته^۳ است که وظیفه نمایش علائم حیاتی یک بیمار به کاربر^۴ (پزشک، پرستار و ...) را دارد. ارتباط دستگاه Viewer با چهار دستگاه جانبی دیگر به شرح زیر است:

۱. آریا: ارتباط دوطرفه است؛ یعنی علاوه‌براین که سیستم Viewer از آریا داده دریافت می‌کند، Viewer هم می‌تواند به آریا داده ارسال کند و تنظیمات دستگاه جانبی آریا را تغییر دهد.

۲. ونتیلاتور: ارتباط یک‌طرفه است؛ یعنی فقط ونتیلاتور به سیستم Viewer داده ارسال می‌کند.

۳. ماژول O3: ارتباط دوطرفه است.

۴. سانترال: ارتباط دوطرفه است.

^۱ ویروس جدید کووید (Coronavirus 2019) یا CoV۲۰۱۹ بیماری است که باعث بیماری تنفسی در افراد می‌شود و می‌تواند از فردی به فرد دیگر گسترش یابد. این ویروس برای اولین بار طی تحقیقات مربوط به شیوع بیماری در ووهان (Wuhan) چین شناسایی شد. در ایالات متحده مشخص شد این ویروس از فردی به فرد دیگر در تماس‌های نزدیک منتشر می‌شود.

^۲ Monitoring: نظارت بر علائم حیاتی بیمار.

^۳ Advanced Monitoring

^۴ کاربر یا User همان فرد استفاده‌کننده از سیستم می‌باشد که می‌تواند پزشک، پرستار یا هر فرد دیگری باشد.

پزشک، پرستار یا هر کاربر دیگری می‌تواند با متصل کردن هر کدام از دستگاه‌های ذکر شده از طریق کابل، اطلاعات مربوط به بیمار را در صفحه نمایش بزرگتر با قابلیت‌های بیشتر نظارت کند و علاوه براین، می‌تواند تاریخچه اطلاعات بیماران را تا ۹۶ ساعت گذشته به صورت دقیق نظارت کند.

جهت برقراری ارتباط با هر دستگاه، از پروتکل ارتباطی^۱ مربوط به آن استفاده می‌شود. پروتکل ارتباطی هر دستگاه توسط سازندگان آن دستگاه فراهم می‌شود و در اختیار دیگر توسعه دهنده‌گان^۲ قرار می‌گیرد. بنابراین در این پژوهش برای برقراری ارتباط با هر کدام از چهار دستگاه جانبی، از پروتکل مربوط به هر کدام استفاده می‌شود.

نرم‌افزار دستگاه Viewer با توجه به صفحات مختلفی که در اختیار کاربران قرار می‌دهد، این امکان را می‌دهد که هر کاربر با توجه به نیاز خود، انتخاب کند که چه پارامترها و چه سیگنال‌هایی را می‌خواهد نظارت کند (پارامترها و سیگنال‌های مربوط به علائم حیاتی بیمار).

در صفحه‌ی بعد، کاتالوگ^۳ یکی از دستگاه‌های مشابه Viewer (شکل Error! No text of Viewer) را مشاهده می‌کنید که در نمایشگاه‌های بین‌المللی خارج از ایران جهت معرفی محصول به مراجعین از این کاتالوگ استفاده می‌شود. همان‌طور که مشاهده می‌کنید، در این کاتالوگ‌ها اطلاعاتی درمورد ویژگی‌های محصول از قبیل اطلاعات صفحه نمایش، نحوه تعامل با دستگاه، نحوه کم و زیاد کردن پارامترها و سیگنال‌های مربوط به مازول‌ها، ذخیره‌سازی اطلاعات، جست‌وجو پیشرفته، قابلیت انتقال اطلاعات به^۴ USB، منوها و دکمه‌های کاربردی و ... ذکر شده است.

^۱ پروتکل ارتباطی مجموعه قوانینی هستند که تعیین می‌کند چگونه دو دستگاه با هم ارتباط برقرار کنند.

^۲ Developers

^۳ کاتالوگ (Catalog): مجموعه اطلاعات گردآوری شده از محصول.

^۴ شرکت Mindray، فعال در حوزه تولید تجهیزات پزشکی.

^۵ Universal Serial Bus

در ادامه‌ی این فصل، به مسائل و چالش‌های موجود در این مطالعه، مفاهیم اصلی، ضرورت و اهداف این پژوهش خواهیم پرداخت.

BeneVision N19/N22

PATIENT MONITORS

Features and Benefits

- 19" and 22" high definition capacitive touchscreen display
- Ultra-thin rotating screen for portrait or landscape viewing
- Platform-wide, intuitive user interface with multi-window viewing capability
- Features N1 Monitor/Module as a multi-parameter module and wireless transport solution
- Standard features are 3, 5, 6-lead ECG, Masimo RD SET® SpO₂, 2 IBP, NIBP, respiration, temperature and arrhythmia analysis with QT/QTc interval monitoring and atrial fibrillation (Afib) detection
- Optional ST-segment monitoring
- Mindray CAAs include SepsisSight™, HemoSight™, ST Graphic, Early Warning Score, Glasgow Coma Scale, and 24-Hour ECG Report
- 12-lead ECG interpretation with report storage, providing immediate access for up to twenty reports at POC
- Extensive data storage including trends, alarms, events, and 48 hours of full disclosure
- Optional 2.4/5 GHz wireless capability
- Connects to BeneVision™ DMS
- Compatible with Mindray eGateway for HL7 communication to EMR
- Optional BeneLink™ Module for ventilator, anesthesia machine and IV pump data integration
- iView module enables point-of-care access to hospital information systems (HIS)
- Optional Multi-gas Module, NMT, INVOS rSO₂ and EEG for the high-acuity monitoring



A new spin on high acuity patient monitoring

The BeneVision N19 and N22 are part of Mindray's premier patient monitoring solution designed to enhance patient care and support timely and accurate clinical assessment across Emergency, Critical Care, and Operating Room environments. With an expansive set of multi-parameter measurements and specialized Clinical Assistance Applications (CAAs), the N19 and N22 provide a fully modular and configurable solution connecting clinicians to precise, relevant clinical data in the most demanding care settings.

Partnered with the N19 and N22, the N1 Monitor/Module can be used as both a module and a transport monitor, ensuring a gap-free patient record from bedside to transport—and beyond. Using the BeneLink Communication Module, data from external devices such as ventilators, anesthesia machines and IV pumps may be viewed, stored and exported at point-of-care, further supporting a comprehensive health record that accompanies the patient throughout the entire care path.

mindray

شکل ۲. Error! No text of specified style in document. کاتالوگ دستگاه مانیتور علایم حیاتی شرکت

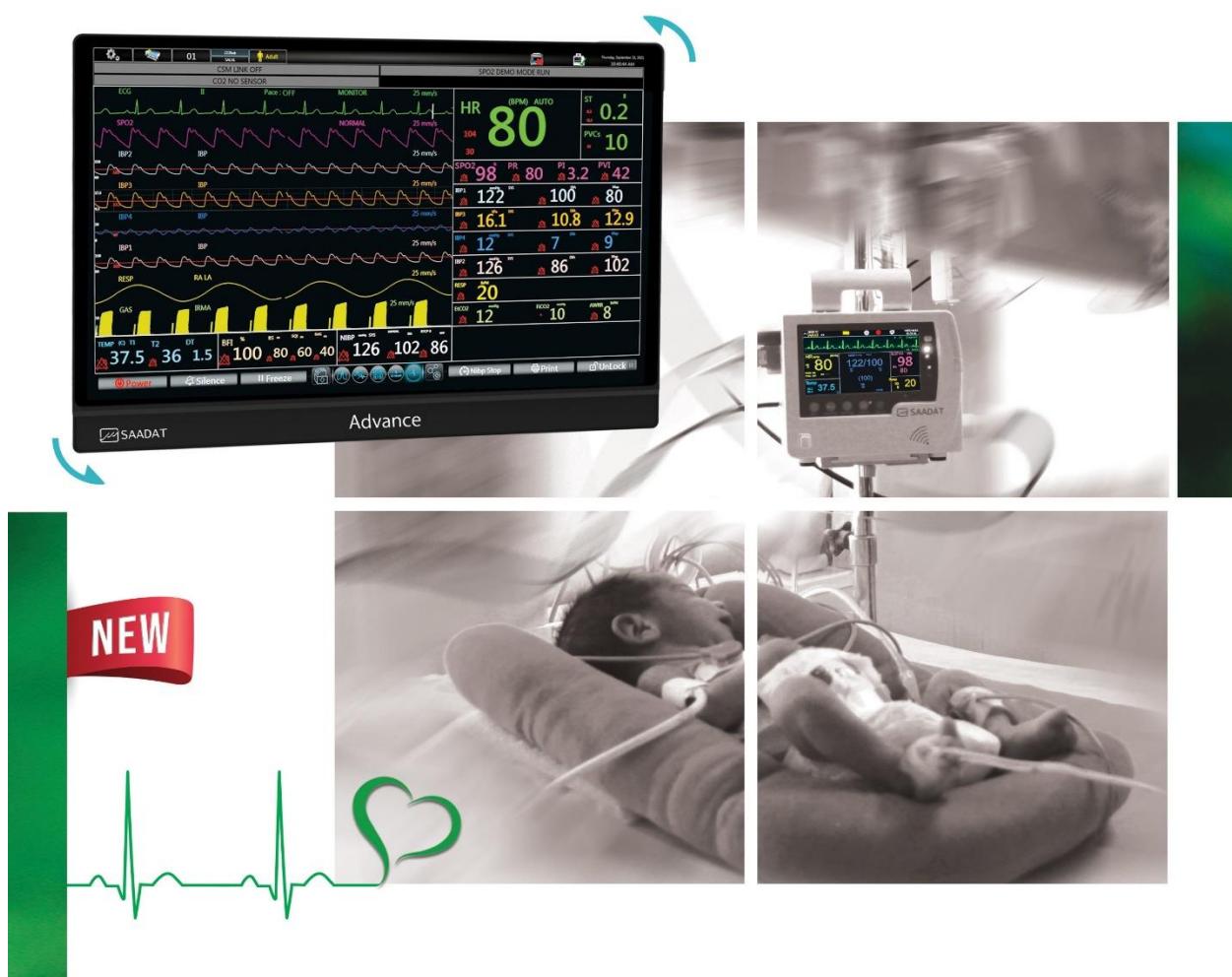
Mindray

Advance Monitor Aria + Viewer 5"+24"

24" color display with Capacitive Touch screen. Viewing angle: 170 degrees | Ability to connect to portable patient monitor (Aria) | Landscape or Portrait display | Dynamic pages with various layouts and zooms | Repositioning parameters and signals with drag & drop | Adjustable screen brightness | Screenshot from the display in JPEG format | Remote control of system | Save patient vital signs including waveforms (Disclosure) and Tabular trends up to 96 hours | Save all alarms that occur during 96 hours and display waveforms when the alarm occurs | Advanced search (Filtering) in patient information history | Export patient medical information to the flash memory in JPEG format | Dual Screen with HDMI port | Simple menu hierarchy gives fast access to all system features | Quick software access keys | Provide electrical power to other connected devices

Options :

Ability to connect to ventilator, central monitoring system, regional oximetry module (Masimo O3), HIS and PACS systems



شکل ۳. Error! No text of specified style in document. دستگاه کاتالوگ ۳.

۱.۱ بیان مسئله

برای اینکه دستگاه Viewer از لحاظ اقتصادی به صرفه باشد تا درمانگاهها قادر باشند این سیستم را برای مشاهده وضعیت بیماران تهیه کنند می‌بایست از سیستم‌هایی که سختافزار محدودی دارند استفاده کرد؛ یعنی نمی‌توان از سختافزارهای روز دنیا^۱ که سرعت و قدرت پردازش بالایی دارند استفاده کرد.

در این پژوهش به دنبال آن هستیم که بتوانیم نرمافزاری را توسعه دهیم تا بر بستر سختافزاری گیادا^۲، بتواند به صورت کاملاً صحیح اجرا شود. گیادا نام یک نوع کامپیوتر است که ویژگی‌های سختافزاری آن به شرح زیر است:

Processor: Intel(R) Celeron CPU^۳ N3160 @ 1.60GHz 1.60 GHz

Installed memory (RAM^۴): 2.00 GB (1.89 GB usable)

حساسیت در انتقال صحیح داده‌ها در صنعت پزشکی بسیار بالا است، زیرا پزشکان بر اساس این داده‌ها و اطلاعات جهت احیای بیمار تصمیم‌گیری‌های اساسی‌ای انجام می‌دهند. برای مثال نرخ داده‌های دریافتی از دستگاه آریا که یکی از دستگاه‌های جانبی است و می‌بایست با این سختافزار محدود مورد پردازش قرار بگیرد حداقل ۴۰ میلی‌ثانیه است، یعنی هر ۴۰ میلی‌ثانیه یک بسته^۵ دریافت می‌شود. هر یک ثانیه معادل ۱۰۰۰ میلی‌ثانیه است، پس در هر ثانیه ۲۵ بسته دریافت می‌کنیم:

$$1000 \div 40 = 25$$

^۱ این پژوهش در سال ۲۰۲۲ میلادی انجام شده است.

^۲ گیادا (Giada) نام یک نوع کامپیوتر.

^۳ CPU: Central Processing Unit

^۴ RAM: Random Access Memory

^۵ در شبکه‌های کامپیوتری برای تبادل اطلاعات، داده‌های بزرگ به قسمت‌های کوچکتر تقسیم می‌شوند و هر یک از این قسمت‌ها را یک بسته (Packet) می‌نامند.

این فقط نرخ داده‌های دریافتی از دستگاه جانبی آریا است، سیستم Viewer باید این قابلیت را داشته باشد تا هم‌زمان با هر چهار دستگاه جانبی ارتباط داشته باشد و به تعامل با هر کدام بپردازد؛ علاوه بر این سیستم Viewer باید بتواند در طولانی مدت کار کند و در شرایط حساس (مثل جراحی بیمار) کارایی خود را حفظ بکند و به هیچ دلیلی از کار نیافتد. بنابراین نیازمند نرم‌افزاری هستیم تا بتوانیم مبتنی بر آن و بر بستر سخت افزاری گیادا، از چندین دستگاه، داده‌هایی را در لحظه^۱ و بدون خطا دریافت، نظارت، ذخیره و مدیریت کنیم و سپس تحت شبکه محلی^۲، آنها را به سانترال ارسال کنیم که در این پژوهش به آن خواهیم پرداخت.

۲.۱ ضرورت انجام تحقیق

در درمانگاه‌ها خیلی اوقات امکان نزدیک شدن به بیمار میسر نیست و لازم است که نظارت وضعیت بیمار با فاصله از خود بیمار انجام شود (مثل اپیدمی کرونا)؛ در نتیجه لازم است داده‌هایی دستگاه‌هایی که بالاسر بیمار هستند به سیستم دیگری که با فاصله از تخت بیمار قرار دارد منتقل شوند. در حوزه پزشکی با جان انسان سروکار داریم و بنابر ضرورت این داده‌ها که مشخص کننده وضعیت بیمار هستند باید نرم افزاری تهیه شود تا داده‌ها را کاملاً صحیح دریافت و علاوه بر مانیتورینگ، عملیات ذخیره‌سازی و ارسال به سانترال را انجام دهد.

دستگاه‌هایی که مشابه با دستگاه Viewer هستند فقط امکان برقراری ارتباط با دستگاه‌های آریا و سانترال را دارند مانند دستگاه‌های البرز بی^۹ (Alborz B9)^۳، الوند اچ ۱۸ (Alvand H18)^۴، الوند

^۱ Realtime

^۲ LAN (Local Area Network)

^۳ Alborz B9 نام یکی از محصولات شرکت پویندگان راه سعادت است؛ جزئیات مربوط به محصول از طریق آدرس اینترنتی زیر قابل رویت است:

https://saadatco.com/?page_id=4023&lang=en

^۴ Alvand H18 نام یکی از محصولات شرکت پویندگان راه سعادت است؛ جزئیات مربوط به محصول از طریق این آدرس اینترنتی، قابل رویت است: https://saadatco.com/?page_id=4070&lang=en

اچ ۱۲ (Alvand H12)^۱، و زاگرس اس (Zagros S)^۲ از شرکت ایرانی پویندگان راه سعادت^۳ یا دستگاه کار迪وست آراد (Cardioset ARAD P10)^۴ از شرکت ایرانی صایران^۵. ما در این پژوهش به دنبال آن هستیم که امکان برقراری ارتباط با دستگاه ونتیلاتور و مازول O3 هم فراهم کنیم و اتصال با این دو دستگاه جانبی جزو نوآوری‌های این پژوهش است.

یکی دیگر از موارد ضروری که بسیار حائز اهمیت است، این است که نرمافزار سیستم Viewer باید با ویژگی‌های سخت افزاری گیادا که قبل تر اشاره شد، بتواند با هر چهار دستگاه جانبی آریا، ونتیلاتور، مازول O3 و سانترال، به صورت همزمان، ارتباط برقرار کند و در طولانی مدت و شرایط حساس مثل جراحی بیمار کارایی خود را حفظ کند.

۳.۱ سؤالات تحقیق

ما در این پژوهش به دنبال ارائه نرمافزاری برای دستگاه پزشکی Viewer هستیم؛ پرسش اصلی این پژوهش عبارت از:

۱ نام یکی از محصولات شرکت پویندگان راه سعادت است؛ جزئیات مربوط به محصول از طریق آدرس اینترنتی زیر قابل رویت است:

https://saadatco.com/?page_id=4086&lang=en

۲ نام یکی از محصولات شرکت پویندگان راه سعادت است؛ جزئیات مربوط به محصول از طریق آدرس اینترنتی زیر قابل رویت است:

https://saadatco.com/?page_id=4105&lang=en

۳ شرکتی فعال در حوزه تولید تجهیزات پزشکی.

۴ نام یکی از محصولات شرکت صایران است؛ جزئیات مربوط به محصول از طریق آدرس اینترنتی زیر قابل رویت است:

<https://www.sairanmed.ir/arhive/products/cardioset-arad-p10-2>

۵ شرکتی فعال در حوزه تولید تجهیزات پزشکی.

- چگونه می‌توان برای دستگاه Viewer با ویژگی‌های سخت‌افزاری ذکر شده (کامپیوتر گیادا)، نرم‌افزاری تهیه کرد که در لحظه، با چندین دستگاه جانبی ارتباط برقرار کند و به ارسال و دریافت داده پردازد؟

۴.۱ اهداف تحقیق

هدف از انجام این پژوهش، طراحی نرم‌افزار دستگاه پزشکی Viewer جهت مانیتورینگ و ذخیره سازی علائم حیاتی بیمار است. علاوه بر این نرم‌افزار باید قابلیت اجرا روی سخت‌افزار گیادا را داشته باشد. با توجه به حساسیت داده‌ها می‌بایست از دریافت صحیح داده‌ها در نرم‌افزار مطمئن بود و نرم‌افزار بتواند به صورت همزمان عملیات مانیتورینگ از چند دستگاه جانبی را انجام دهد.

۵.۱ روش تحقیق

تمامی مراحل تحقیق و توسعه نرم‌افزار در واحد تحقیقات^۱ شرکت دانش‌بنیان و بین‌المللی پویندگان راه سعادت^۲ انجام شده است. توسعه دهنده‌گان قبلی نرم‌افزار شرکت پویندگان راه سعادت، در سال ۲۰۱۷ میلادی این طرح را پیاده سازی کرده بودند؛ اما نرم‌افزار قابل قبول نبود چون‌که میزان استفاده از منابع سیستم سعودی بود، یعنی با گذر زمان میزان استفاده از منابع سیستم افزایش پیدا می‌کرد و فقط امکان مانیتورینگ دستگاه آریا و ارسال اطلاعات به سانترال وجود داشت[۲۱] و دیگر امکان توسعه نرم‌افزار جهت اتصال به دستگاه ونتیلاتور و مازول O3 میسر نبود. تجربه توسعه دهنده‌گان قبلی در جهت تحقیق و توسعه نرم‌افزار فعلی دستگاه Viewer کمک شایانی کرد. علاوه بر این، مطالعه مقالات ثبت شده در

^۱ R&D: Research and development

^۲ پویندگان راه سعادت (SAADATCO): این شرکت به مدیریت آقای عبدالرضا یعقوب زاده، یکی از پیشروان در عرصه تولید تجهیزات پزشکی کشور است.

سایت های معتبر از قبیل IEEE^۱, Sience Direct^۲ و ACM^۳ ... در حوزه های توسعه نرم افزار، معماری های نرم افزاری و بعضا پژوهشکی جهت تحقیق و توسعه لازم بود و انجام گردید. به تفصیل در ادامه این پژوهش به نحوه پیاده سازی و ارزیابی نرم افزار دستگاه Viewer خواهیم پرداخت.

۶.۱ معرفی فصول پایان نامه

- فصل اول، مقدمه: در این فصل خواننده با کلیات پژوهش آشنا می شود.
- فصل دوم، پیشینه تحقیق: به پیشینه پژوهش می پردازیم و بر مطالعات انجام شده مروری خواهیم داشت.
- فصل سوم، روش تحقیق و داده ها: در مورد تحقیقات، داده ها و الگوی بهینه اطلاعات کسب می کنید و سپس با روش پیشنهادی این پژوهش آشنا خواهد شد.
- فصل چهارم، آزمایش: به پیاده سازی طرح می پردازیم و با انجام آزمون های متعدد، تجزیه و تحلیلی انجام می دهیم و کارایی روش پیشنهاد شده را بررسی خواهیم کرد.
- فصل پنجم (فصل پایانی): به یافته های کار تحقیقاتی و مطالعات آتی می پردازیم.

۷.۱ تعاریف اصلی مورد استفاده

مانیتورینگ: نظارت بر علائم حیاتی بیمار.

^۱ یکی از بزرگترین پایگاه های اطلاعاتی در زمینه مهندسی برق، تکنولوژی اطلاعات، الکترونیک، کامپیوتر، فیزیک، کاربردی و علوم وابسته است. عنوان اصلی این پایگاه "انجمن مهندسین برق و الکترونیک آمریکا" نام دارد که یکصد سال از تاسیس آن می گذرد.

^۲ ساینس دایرکت یکی از معتبرترین مراجع برای جستجوی مقالات و کتب علمی و پژوهشکی است.

^۳ نام یک انجمن علوم کامپیوتری.

مجموعه داده(Dataset^۱): مجموعه داده به مجموعه‌ای گفته می‌شود که آزمایش مورد نظر روی آن‌ها انجام می‌شود.

معماری کامپیوتر: معماری کامپیوتر به دو گرایش زیر تقسیم می‌شود:

۱. معماری سخت‌افزار: در این گرایش با اجزای داخلی کامپیوتر که مراحل انجام یک دستور را بر عهده دارند و چگونگی کار آن‌ها مواجه هستیم. در این گرایش واحد کنترل مرکزی و حافظه به عنوان دو بخش اصلی کامپیوتر معرفی می‌شوند که باید میزان مصرف آن‌ها را کنترل و مدیریت کنیم.
۲. معماری نرم‌افزار: به طور مختصر نحوه گروه‌بندی اجزاء در نرم‌افزار است.

آریا: نام یکی از دستگاه‌های جانبی است که وظیفه جمع‌آوری داده‌های مربوط به علائم حیاتی بیمار را دارد و این کار را از طریق پدهایی^۲ که به بدن بیمار متصل می‌شوند انجام می‌دهد. دارای یک صفحه نمایش پنج اینچی است که برای نظارت در حمل و نقل (استفاده در آنبولانس) کاربرد دارد، اما برای نظارت در بیمارستان کاربردی نیست، چون کوچک است.

ونتیلاتور: دستگاه ونتیلاتور (تهویه مکانیکی یا تهویه مصنوعی) را می‌توان دستگاه تنفسی نامید که به بیمارانی که مشکلات تنفسی موقت یا دائمی دارند و نمی‌توانند به تنها‌یی نفس بکشند، تنفس می‌دهد.

ماژول O3: سیستم O3 یک اکسی‌متر غیرتهاجمی متصل به بیمار است که برای نظارت بر اشیاع اکسیژن منطقه‌ای هموگلوبین، از جمله بافت (rSO2) مغزی طراحی شده است. می‌توان از آن در هر محیط مراقبت بهداشتی استفاده کرد که در آن اندازه‌گیری rSO2 ممکن است.

پروتکل: جهت ارتباط دو نرم افزار مختلف در دو دستگاه، بگونه‌ای که بتوانند به راحتی با یکدیگر ارتباط برقرار کنند و تبادل اطلاعات داشته باشند.

کارآمد: منظور از کارآمدی این است که نرم‌افزار نباید از منابع سیستم از قبیل پردازنده و رم استفاده بیهوده بکند؛ بنابراین کارآیی شامل پاسخگویی، زمان پردازش، استفاده از حافظه و ... است [۲۲]. در این

^۱ Dataset

^۲ Pad

پژوهش منظور از کارآمدی این است که نرمافزار دستگاه Viewer، باید بتواند بر بستر سختافزاری کامپیوتر گیادا، با هر چهار دستگاه جانبی آریا، ونتیلاتور، ماژول O3 و سانترال، به صورت همزمان ارتباط برقرار کند و بسته‌های انتقالی را به صورت صحیح و بدون خطا دریافت کند.

۸.۱ خلاصه فصل

سیستم مانیتورینگ Viewer یکی از دستگاه‌های صنعت پزشکی در درمانگاه‌ها است که وظیفه مانیتورینگ و ذخیره‌سازی علائم حیاتی یک بیمار را دارد. انواع دستگاه‌های جانبی که با دستگاه Viewer ارتباط برقرار می‌کنند:

- آریا
- ونتیلاتور
- O3
- سانترال

نرم افزار دستگاه Viewer باید نرمافزاری باشد تا بر بستر سخت افزاری گیادا به صورت کاملاً صحیح اجرا شود و بتواند عملیات مورد نظر (مانیتورینگ به صورت همزمان، ذخیره سازی، ارسال داده‌ها به سانترال) را انجام دهد.

۲

فصل دوم

پیشینه تحقیق

پیشینه تحقیق

سیستم Viewer با چهار دستگاه جانبی (آریا، ونتیلاتور، مازول O3 و سانترال) می‌بایست ارتباط برقرار کند و به تعامل باهم بپردازند. وضعیت سلامتی بیمار، تاثیری در روند ارتباطات و تعاملات بین سیستم Viewer با دستگاه‌های جانبی نمی‌گذارد؛ یعنی این‌که بیمار از لحاظ سلامتی، چه در شرایط بحرانی باشد و چه نباشد داده‌های انتقالی اولویت یکسانی دارند و سیستم Viewer باید در هر شرایطی که بیمار است، به صورت همزمان علائم حیاتی بیمار را نظارت کند.

در حوزه‌ی نرمافزار وقتی که موضوع انتقال داده با یک یا چند گره (نود^۱) مطرح می‌شود، با مسائلی همچون سنجش، ارسال، پردازش و ذخیره‌سازی داده‌ها مواجه می‌شویم؛ درنتیجه نیاز به یک معماری نرمافزاری است که این نوع مسائل را مرتفع کند. در ادامه‌ی این فصل جهت طراحی نرمافزار دستگاه Viewer، مروری بر پژوهش‌هایی که صورت گرفته خواهیم داشت.

۱.۲ مروری بر پژوهش‌های پیشین

جهت طراحی نرمافزار دستگاه Viewer، لازم است که یک الگوی معماری نرمافزاری مناسبی را برای نرمافزار دستگاه طراحی کنیم. الگوی معماری، مجموعه‌ای از زیرسیستم‌های از پیش تعریف شده است که وظایف آن‌ها، قوانین و جهت‌دهی برای سازماندهی رابطه میان آن‌ها را فراهم می‌کند. یکی از الگوهایی که در نرمافزارهای کاربردی مورد استفاده قرار می‌گیرد، الگوی معماری^۲ MVC است که در فارسی به معنای «مدل-نما-کنترل‌گر» شناخته می‌شود. مدل، نما و کنترل‌گر اجزاء یا لایه‌های الگوی معماری MVC هستند. [۲۲]

وظایف هر کدام از لایه‌های معماری MVC به شرح زیر است:

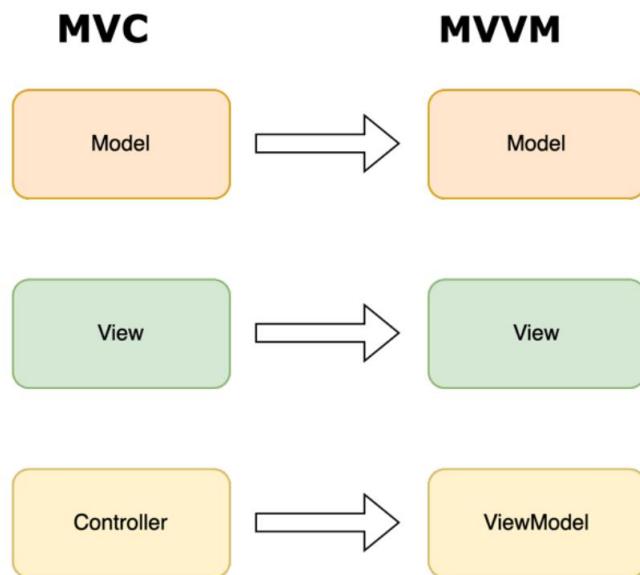
نما: این معماری سه لایه نرمافزار، شامل یک لایه رابط کاربری می‌باشد که این بخش همان نما می‌باشد. یعنی صفحاتی که به کاربر نمایش داده می‌شود بخش نما هستند.

^۱ گره یا نود (Node) به هر دستگاه فیزیکی که قادر به ارسال و دریافت اطلاعات است می‌گویند.

^۲ Model View Controller

مدل: در معماری MVC بخش مدل مربوط به پایگاه داده و ارتباط با پایگاهداده^۱ میباشد.

کنترل گر: این بخش از معماری سه لایه نرم افزار، ارتباط بین مدل و نما را برقرار میکند. به این صورت که درخواست کاربر به کنترل گر ارسال میشود، حال در کنترل گر دادهها بررسی میشوند و پس از تایید شدن، در صورت مجاز بودن اطلاعات را به مدل ارسال میکند و در مدل آنها در پایگاهداده ذخیره میشوند. یکی دیگر از الگوها، الگوی معماری MVVM^۲ است که الهام گرفته شده از الگوی MVC است (شکل ۱.۲)، با این تفاوت که نقطه ورودی به این الگو به جای کنترلرگر، نما است. ما در این پژوهش، در قسمتی از نرم افزاری که ارائه میدهیم از الگوی MVVM استفاده خواهیم کرد. این الگو، یک الگوی معماری مرسوم برای توسعه و ساخت برنامه های کاربردی است؛ اما کاری که در این پژوهش به دنبال آن هستیم تا انجام دهیم فراتر از این است؛ یعنی علاوه بر رابط کاربری، با مبحث انتقال دادهها مواجه هستیم و فقط در بخش مربوط به رابط کاربری از این الگو استفاده میکنیم [۲۲].



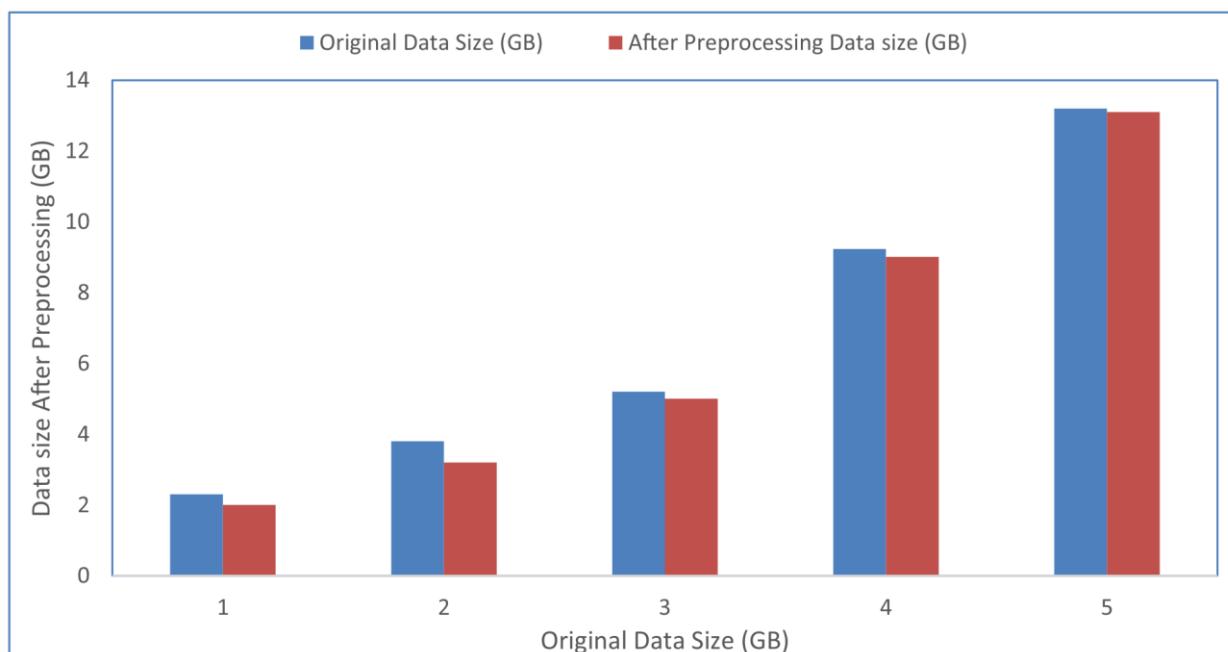
شکل ۱.۲ اجزای معماری MVC و MVVM

علاوه بر پژوهش های صورت گرفته شده مربوط به الگوهای توسعه نرم افزاری، پژوهش هایی هم در رابطه با انتقال دادهها صورت گرفته که به آنها اشاره میکنیم. در مقاله ای که آقای K.R. Aravind

^۱ محل ذخیره سازی دادهها.

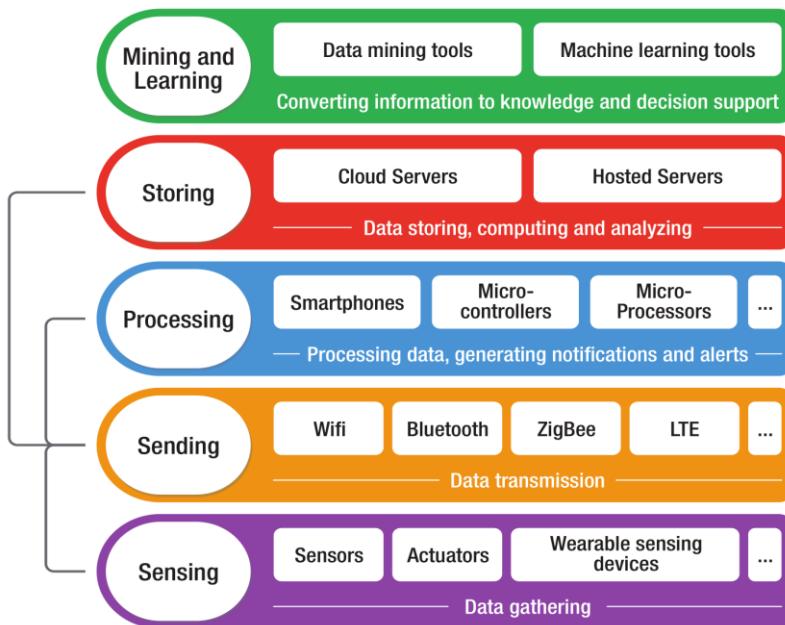
^۲ Model View ViewModel

Britto و دیگر همکارانشان در سال ۲۰۱۹ میلادی ارائه داده اند [۶]، به این مسئله، یعنی چالش مدیریت داده‌ها پرداخته‌اند و تلاش کرده‌اند تا در صنعت مراقبت‌های بهداشتی، یک فناوری شناختی جهت نظارت و انتقال داده‌های پزشکی ارائه بدهند، اما همانطور که در شکل مشاهده می‌کنید (شکل ۲.۲)، در روشی که برای انجام این کار، یعنی نظارت، ثبت و انتقال داده‌های مربوط به بیمار ارائه داده‌اند، اولویت در انتقال داده‌ها، زمانی اختصاص داده می‌شود که وضعیت سلامتی بیمار بحرانی است و اگر بیمار در حالت بحرانی نباشد بخشی از داده‌ها را قبل از انتقال حذف می‌کنند، اما در روشی که در این پژوهش به دنبال آن هستیم تا ارائه دهیم، فرقی نمی‌کند که بیمار در وضعیت سلامت بحرانی یا غیر بحرانی قرار دارد، بلکه تمامی داده‌های انتقالی در هر وضعیتی که بیمار است، دارای اولویت یکسانی هستند و همگی باید در لحظه انتقال پیدا کنند و مانیتور شوند.



شکل ۲.۲ سایز داده‌ها انتقالی قبل و بعد از پردازش اولیه.

در مقاله دیگری هم که توسط خانم h.nguyen و همکارانشون در سال ۲۰۱۷ میلادی ارائه شده است [۱] به مسائل سنجش، ارسال، پردازش و ذخیره‌سازی داده‌ها پرداخته‌اند و همانطور که در شکل مشاهده می‌کنید (شکل ۳.۲)، به این نتیجه رسیدند که یک معماری طبقه‌بندی شده جهت تفسیر داده‌ها ضروری است.



شکل ۳.۲ معماری طبقه‌بندی شده تفسیر داده.

زمانیکه از دیدگاه مهندس نرم‌افزار به این مسائل نگاه می‌کنیم، ما نیازمند به معماری نرم‌افزاری طبقه‌بندی شده‌ای هستیم که بتوانیم با آن، داده‌های دریافتی را مدیریت کنیم. معماری‌های نرم‌افزاری با همین منظور وجود دارند مثل معماری که آقای Y. Nait Malek که در سال ۲۰۱۷ میلادی پیشنهاد داده‌اند [۲]. معماری پیشنهاد شده برای پردازش جریان داده‌ها از سه لایه تشکیل شده است:

Data acquisition^۱ •

Data processing^۲ •

Data storage and visualization^۳ •

در این معماری نرم‌افزاری، وظیفه لایه Data acquisition ادغام داده‌های جمع‌آوری شده از حسگرها است، وظیفه لایه Data processing پردازش داده‌ها است و لایه Data storage and visualization هم برای نظارت و ذخیره‌سازی داده‌ها هستند. این معماری نرم‌افزاری برای کاری که در

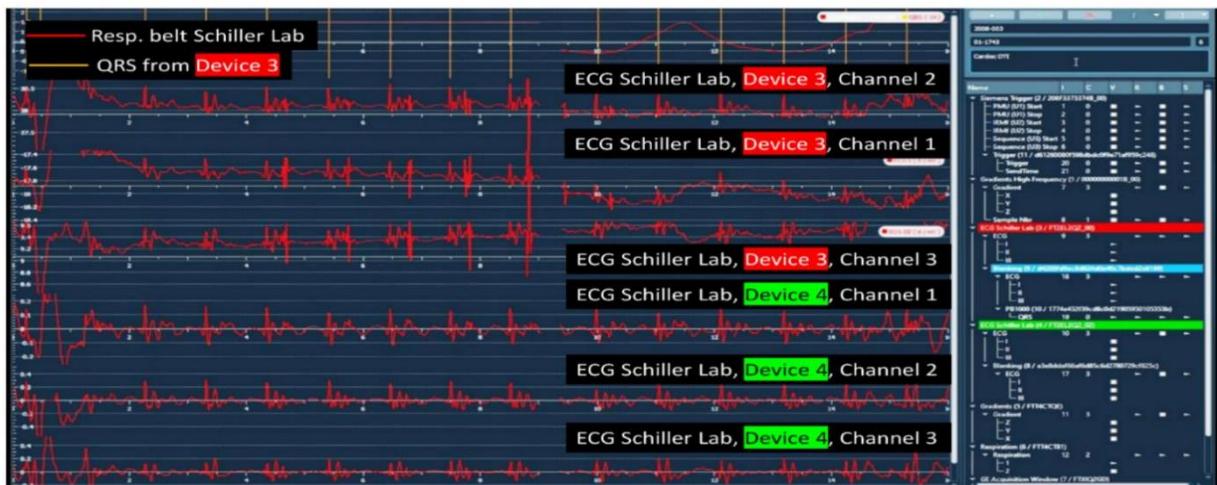
^۱ اکتساب داده‌ها.

^۲ پردازش داده‌ها.

^۳ ذخیره‌سازی و بصری‌سازی داده‌ها.

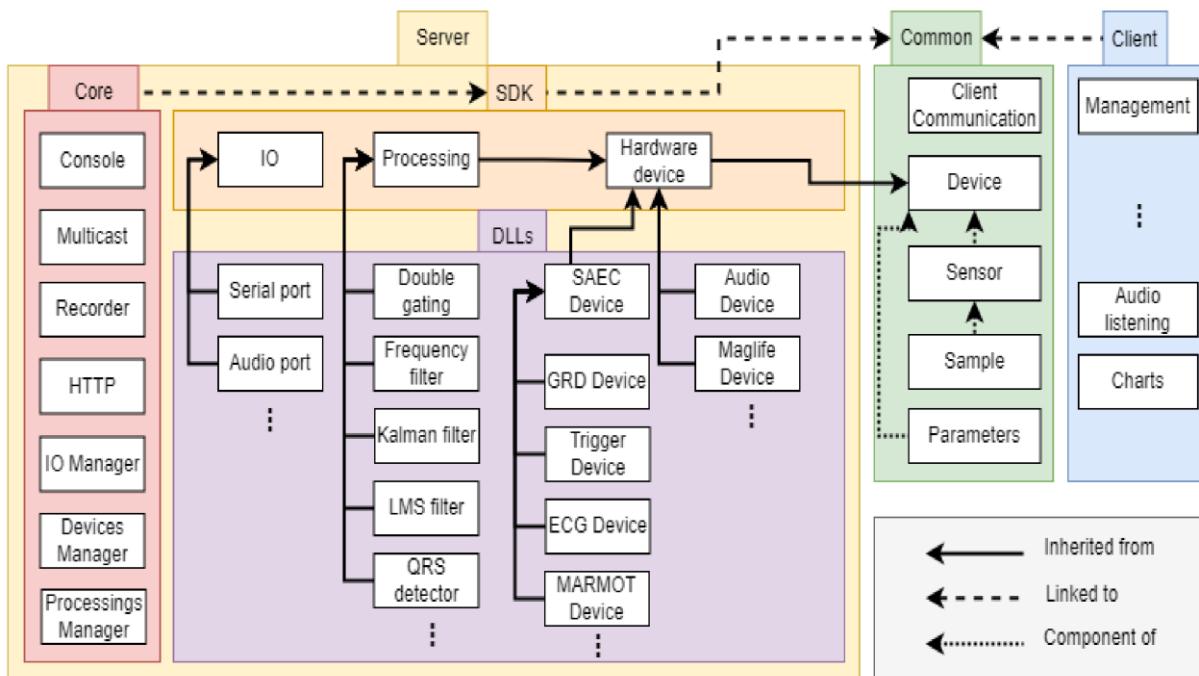
این پژوهش به دنبال آن هستیم تا انجام دهیم مناسب نیست، زیرا با توجه به محاسباتی که در فصل پیشین انجام شد پی بردم که در هر ثانیه ۲۵ بسته داده‌ای دریافت می‌کنیم، در نتیجه این نوع مرزبندی بین لایه‌ها مناسب این نرخ دریافتی نیست؛ مرزبندی بین لایه‌ها و تقسیم کارها می‌باست خیلی بهتر از این انجام شود و در این پژوهش قصد داریم به درستی این مرزبندی‌ها را انجام دهیم. به عنوان مثال لایه‌ی data storage و visualization باید جدا از هم باشند زیرا هر کدام وظیفه جدایهای دارند و به همیگر مرتبط نیستند، یعنی بعضی موقع باید به لایه data storage داده‌های خام داده شود تا در پایگاه‌داده ذخیره شوند، اما به لایه visualization باید داده‌های پردازش شده داده شود تا به پزشک نمایش داده شود، پس نمی‌توان این دو لایه را با هم ادغام کرد و داده‌های مشترک تزریقشان کرد.

در مقاله‌ای که توسط آقای Karyna Isaieva و دیگر همکارانشون در سال ۲۰۲۲ میلادی ارائه شده است [۱۹]، مانند کاری که ما در این پژوهش به دنبال آن هستیم تا ارائه دهیم، به مانیتورینگ سیگنال‌های ECG که نمایانگر وضعیت قلب بیمار است پرداخته‌اند و یک روش برای خواندن داده‌ها از دستگاه جانبی پیشنهاد شده داده‌اند (شکل ۴.۲).



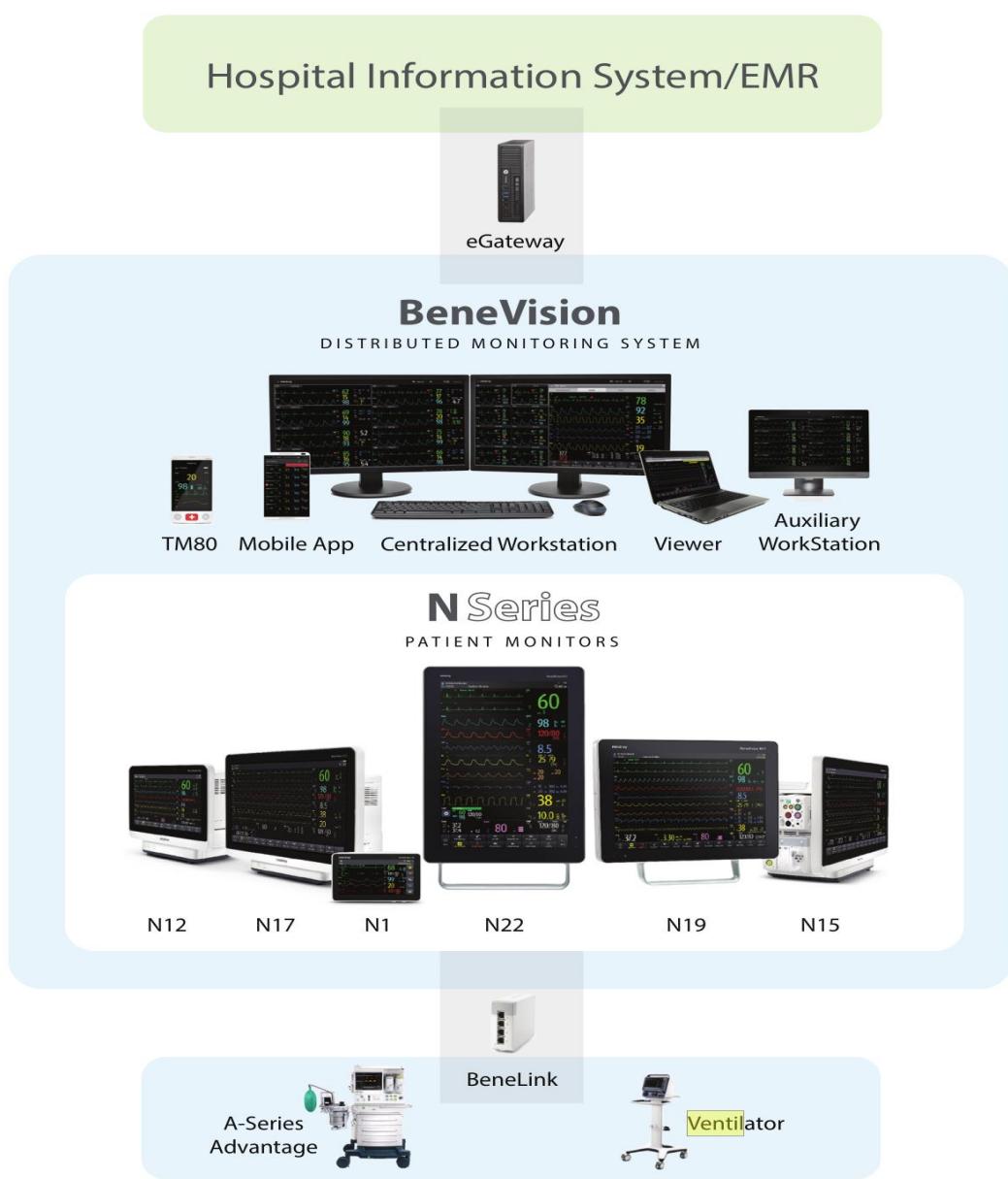
شکل ۴.۲ مانیتورینگ در لحظه داده‌ها.

همانطور که در شکل مشاهده می‌کنید (شکل ۵.۲)، در این معماری، داده‌ها به صورت همزمان از دستگاه جانبی دریافت شده و با موفقیت عملیات مانیتورینگ صورت گرفته؛ اما این روش فقط در صورتی پاسخگو است که ارتباط با دستگاه جانبی فقط به صورت یک‌طرفه باشد.



شکل ۵.۲ روش ارتباط با دستگاه جانبی.

در روشنی که ما در این پژوهش ارائه می‌دهیم، می‌خواهیم داده‌ها را به صورت همزمان از دستگاه‌های جانبی دریافت کنیم و عملیات مانیتورینگ را روی آن‌ها انجام دهیم اما علاوه بر این، به دنبال آن هستیم تا مانند دستگاه‌های مشابه خارجی، مثل مانیتورهای سری N شرکت Mindray [۲۱]، با دستگاه‌های جانبی به صورت همزمان ارتباط دو طرفه برقرار شود و هر ارتباط توانایی ارسال و دریافت داده به یکدیگر را داشته باشد (شکل ۶.۲).

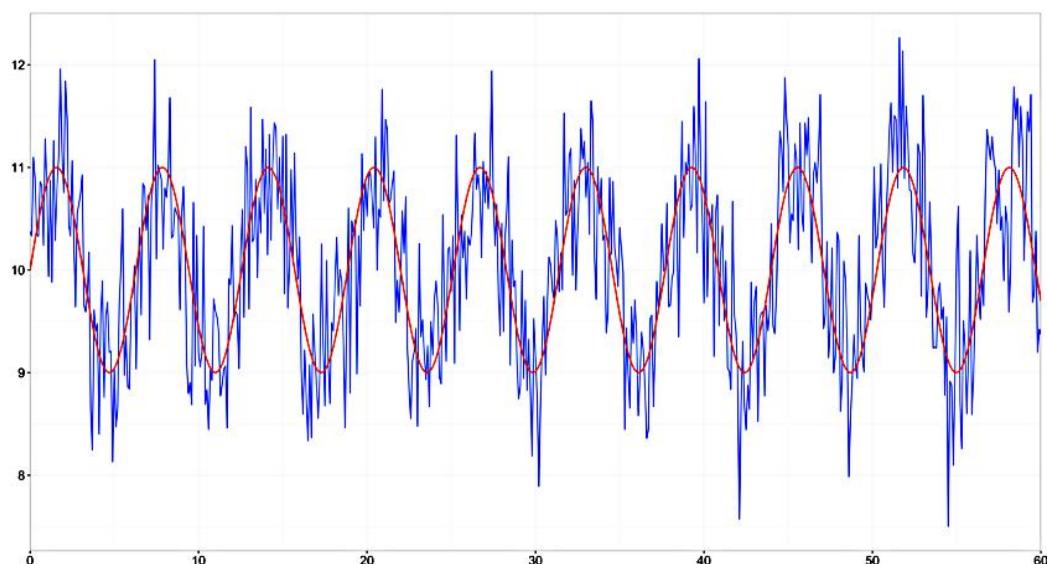


شکل ۶.۲ مانیتورهای سری N شرکت Mindray

یکی دیگر از مسائل رفع نویز^۱ سیگنال می‌باشد، یعنی در صورتی که سیگنال‌ها دارای نویز باشند نیاز به رفع نویز است و می‌بایست کاری مشابه با کاری که آقای Rajalakshmi Krishnamurthi و

^۱ نویز به سیگنالی ناخواسته گفته می‌شود که بر سیگنال‌های دیگر اثر نامطلوب می‌گذارد، مثلاً با اضافه شدن به آن‌ها شکل آن‌ها را تغییر می‌دهد، و در نتیجه باعث اختلال می‌شود.

همکارانشون در مقاله‌ای، در سال ۲۰۲۰ میلادی ارائه دادند [۴] انجام شود، یعنی از روشی برای تشخیص داده‌های پرت، داده‌های گمشده و تجمیع داده‌ها استفاده کرد (شکل ۷.۲). در این پژوهش تمامی دستگاه‌های جانبی که به دستگاه Viewer متصل می‌شوند، خودشان عملیات حذف نویز داده‌های حسگر را انجام می‌دهند و بعد از این که داده‌های پرت را حذف می‌کنند، داده‌های تجمیع شده‌ی درست (بدون نویز) را به Viewer ارسال می‌کنند. در صورتی که در آینده، مانیتور دستگاه جانبی دیگری به سیستم اضافه شود باید به این نکته حذف نویز سیگنال‌ها توجه داشت.



شکل ۷.۲ سیگنال‌های نویزی.

برای ارسال داده‌ها به مرکز از طریق شبکه، از پروتکل^۱ TCP و UDP^۲ استفاده می‌کنیم و طبق مقاله‌ای که آقای S.M. Dilip Kumar در سال ۲۰۱۸ میلادی ارائه داده اند [۵]، مسیریابی، حفظ انرژی، تراکم، مقیاس پذیری، قابلیت اطمینان، امنیت و ... در شبکه را می‌بایست مورد سنجش قرار داد تا شبکه از لحاظ بهینگی در وضع مناسبی برای استفاده قرار بگیرد.

^۱ Transmission Control Protocol

^۲ User Datagram Protocol

علاوه بر پژوهش‌های صورت گرفته، پروژه Viewer که قبلاً توسط دیگر برنامه‌نویسان در شرکت پویندگان راه سعادت توسعه یافته بود^[۲۱] و اصلاً به مرحله پروداکشن^۱ نرسیده بوده‌م مورد بررسی قرار گرفت و نقاط ضعف شناسایی شد و در پروژه جدید رفع شدند. برای مثال در پروژه قبلی Viewer از کامپوننت‌های^۲ آمده برای ترسیم سیگنال‌ها استفاده می‌شد، به همین دلیل میزان استفاده از منابع سیستم بیش از حد بالا می‌رفت؛ اما در این پژوهش کامپوننت شخصی‌سازی شده و سفارشی خودمان را درست کردیم تا در استفاده از منابع سیستم، صرفه جویی کرده باشیم تا به کارآمدی نرم‌افزار بیفزاییم.

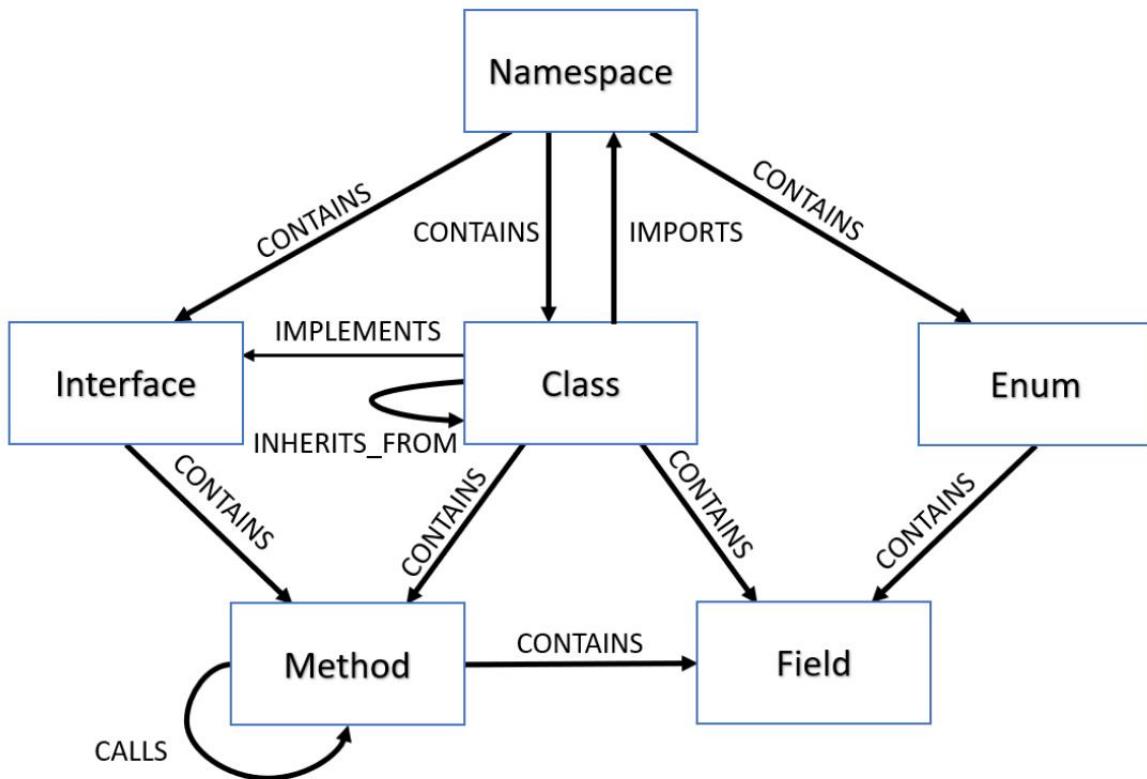
جهت پیاده‌سازی روشی که در این پژوهش ارائه می‌دهیم لازم است که اصول توسعه نرم‌افزار از قبیل SOLID^۳، الگوهای طراحی^۴ و ... را در نرم‌افزار رعایت کرد و مانند روشی که آقای David Heidrich همکارانشون در سال ۲۰۲۲ میلادی ارائه داده‌اند [۲۰]، از ساختار نظم یافته‌ای جهت توسعه کدهای نرم‌افزار استفاده کرد (شکل ۸.۲).

^۱ پروداکشن یا production: مرحله ارائه محصول.

^۲ کامپوننت یا Component در علم برنامه‌نویسی یک بلوک قابل تفکیک از برنامه است. این بلوک‌ها معمولاً خاصیت استفاده مجدد دارند و می‌توانند با سایر کامپوننت‌ها ترکیب شده و یک اپلیکشن جدید را ایجاد نمایند.

^۳ سولید (SOLID) یک کلمه مخفف برای پنج اصل اولیه طراحی شی‌گرا است که رابرт سیسیل مارتین، آن را مطرح کرد. این اصول زمانی که دست به دست هم میدهند، کار توسعه یا اضافه کردن قابلیت‌های جدید به برنامه و نگهداری یا همون دیباگ یک برنامه را برای برنامه نویس‌ها آسان می‌کنند.

^۴ الگوهای طراحی (Design patterns) راه حل‌های آزموده و بهینه‌ای در مسائل مربوط به توسعه نرم‌افزار هستند. دیزاین پترن الگویی است که باید در موقعیت مناسب خود مورد استفاده قرار بگیرد تا بهترین نتیجه را در توسعه نرم‌افزار در اختیار توسعه دهنده قرار دهد.



شکل ۸.۲ گراف^۱ ساختاری کدهای نرم افزار.

در این ساختار، نرم افزار می‌تواند شامل چندین فضای نام^۲ باشد که هر فضای نام می‌تواند شامل چندین اینترفیس^۳، کلاس^۴ و اینام^۵ باشد. هر کلاس می‌تواند از کلاس‌های دیگر و اینترفیس‌ها استفاده کند. هر

^۱ Graph

^۲ در علم رایانش، یک فضای نام به انگلیسی (Namespace) یک مجموعه نماد می‌باشد که از آن برای سازمان دادن به انواع متفاوت شیء استفاده می‌شود، هدف آن است که این اشیا بتوانند توسط نام ارجاع شوند.

^۳ اینترفیس مجموعه‌ای از قراردادها (contract) است که متدهای مختلف بدون تعریف بدنه (Body) و عملکرد در آن تعریف شده‌اند.

^۴ کلاس‌ها نقشه‌های اولیه‌ای هستند که رفتار و اطلاعات اشیا را تعریف می‌کنند. آن‌ها به توسعه‌دهنده اجازه می‌دهد نمونه‌های جدید را تولید و نمونه‌سازی کند.

^۵ کلمه اینام یا enum از واژه Enumeration به معنای «شمارش» گرفته شده است که در زبان‌های برنامه‌نویسی به منظور تعریف شرایطی با بیش از دو حالت مورد استفاده قرار می‌گیرد.

کلاس می‌تواند شامل چندین متدها^۱ و فیلد^۲ باشد اما هر اینترفیس فقط می‌تواند شامل چندین متدهای باشد. هر متدهای می‌توانند شامل چندین فیلد باشد که به آنها متغیرهای محلی گویند، همچنین هر متدهای می‌توانند از خودش و متدهای دیگر استفاده کند. هر فیلد می‌تواند از جنس اینامهای توسعه داده شده در فضای نام‌ها باشد.

همانطور که اشاره کردیم، ما باید نرم‌افزاری که درست می‌کنیم، به نحو صحیحی به بخش‌ها و لایه‌های مختلف تقسیم بشود؛ آقای Eric Evans که شروع کنندهٔ جریانی از نوعی طراحی معماری با عنوان طراحی محور دامنه است، در کتاب منتشر شده‌شان با عنوان "[۲] "Domain Driven Design" در مورد مفاهیمی مانند یکپارچگی، محدوده‌های مفاهیم و لایه‌هایی برای جلوگیری از خرابی و انحراف برنامه صحبت می‌کند. در این پژوهش با بررسی پژوهش‌های قبلی، مقالات، رویکردها، روش‌ها، معماری‌ها و ... بر آن هستیم که مبتنی بر رویکرد طراحی نرم‌افزار DDD، برنامه را به محدوده‌های مفاهیمی مناسبی تقسیم‌بندی کنیم و روشی را برای مدیریت (دریافت، ارسال، خواندن، ذخیره سازی و نمایش) داده‌ها ارائه دهیم.

۲.۲ جمع‌بندی و نتیجه‌گیری

با توجه به نیازهای سیستم Viewer و مدل ارتباطی آن با دستگاه‌های جانبی، نتیجه می‌گیریم که جهت مانیتورینگ و ذخیره‌سازی علائم حیاتی بیمار، دستیابی به یک معماری نرم‌افزار مناسب، امری ضروری است.

با توجه به مطالعات انجام شده که در این فصل مرور کردیم، به دنبال آن هستیم که یک نرم‌افزاری، جهت مدیریت داده‌هایی که حاصل از تعامل با بیش از یک دستگاه هستند ارائه دهیم. این روش مبتنی

^۱ متدهای مجموعه‌ای از دستورات در برنامه‌نویسی هستند که دارای نام خاصی می‌باشند.

^۲ فیلد‌ها معمولاً با سطح دسترسی خصوصی و برای نگهداری از داده‌هایی که مورد نیاز بیش از یک متدهای دستگاه هستند، درون کلاس است و آن داده‌ها باید پس از خاتمه کار یک متدهای همچنان باقی بمانند استفاده می‌شود. بدیهی است در غیر این صورت به جای تعریف فیلد می‌توان از متغیرهای محلی (متغیری که درون خود متدهای تعریف می‌شود) استفاده نمود.

^۳ Domain Driven Design

بر رویکرد DDD است [۳]، زیرا می‌توانیم بین لایه‌ها مرزبندی درستی را انجام دهیم و داده‌های دریافتی را مدیریت کنیم.

۳.۲ خلاصه فصل

در این پژوهش انتقال داده با چهار گره صورت می‌گیرد، همچنین سخت افزار گیادا که نرم‌افزار باید روی آن اجرا شود محدود است، در نتیجه نیاز به معماری نرم‌افزاری مناسب جهت مدیریت داده‌ها بیشتر حس می‌شود. با توجه به مطالعات صورت گرفته نتیجه گرفتیم که:

- معماری نرم‌افزار باید مرزبندی خوبی داشته باشد.
- بهینگی در شبکه جهت ارسال داده‌ها به سانترال، بسیار حائز اهمیت است.
- معماری نرم‌افزار باید عملیات مانیتورینگ را چه در شرایط بحرانی و چه در شرایط غیربحرانی به صورت همزمان انجام دهد.
- به جای استفاده از کامپوننت‌های آماده، کامپوننت‌های شخصی‌سازی شده و مختص به نیاز سیستم را توسعه دهیم.
- در صورت نیاز، می‌بایست نویزهای مربوط به سیگنال (علائم حیاتی) بیمار رفع گردد.

۳

فصل سوم

روش تحقیق و داده‌ها

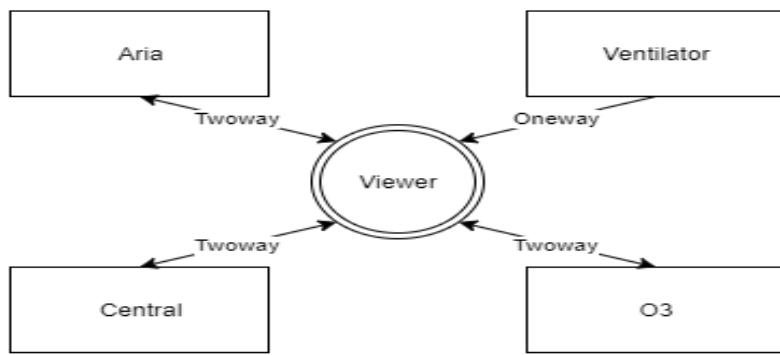
روش تحقیق و داده‌ها

تا به اینجا با مفاهیم اصلی، دستگاه Viewer و سخت‌افزار آن آشنا شدیم و به اهمیت وجود این دستگاه در درمانگاه‌ها پی بردیم. در ارتباطات بین دستگاه Viewer با دستگاه‌های جانبی، با چالش انتقال داده‌ها مواجه هستیم. طبق بررسی‌هایی که صورت گرفت، نتیجه گرفتیم که بنابر ضرورت این داده‌ها که مشخص‌کننده وضعیت بیمار است باید نرم‌افزاری تهیه شود تا داده‌ها را کاملاً صحیح دریافت و علاوه بر مانیتورینگ، عملیات ذخیره‌سازی و ارسال به واحد مرکزی یا سانترال را انجام دهد.

نیازمندی‌های دستگاه Viewer می‌بایست به طور دقیق مورد بررسی قرار بگیرد و همچنین معماری نرم‌افزاری دستگاه‌هم باید از لحاظ مهندسی نرم‌افزار به درستی مرزبندی شود. روش انجام این پژوهش بدین صورت است که در ابتدا نیازمندی‌های سیستم از دیدگاه کاربر فراهم می‌شود تا مسائل و چالش‌هایی که در حین توسعه نرم‌افزار با آن‌ها مواجه می‌شویم، مشخص شوند. طبق مطالعاتی که در فصل قبل موروی بر آن‌ها داشتیم، برای حل این مسائل روش‌هایی وجود داشت، در این فصل به دنبال تفصیل روش طراحی نرم‌افزار دستگاه پژوهشی Viewer هستیم که از طریق آن؛ عملیات مانیتورینگ، ذخیره‌سازی علائم حیاتی بیمار و ارسال آنها به سانترال امکان‌پذیر می‌باشد. در این پژوهش، معماری نرم‌افزاری مبتنی بر رویکرد DDD طراحی و توسعه خواهیم داد و در ادامه، روش پیشنهاد شده‌ی این پژوهش را پیاده‌سازی خواهیم کرد و مورد ارزیابی قرار خواهیم داد. بدین منظور در ادامه این فصل به شرح دستگاه Viewer، نیازمندی‌های نرم‌افزار، ابزارهای استفاده شده و معماری سیستم می‌پردازیم.

۱.۳ شرح دستگاه Viewer

دستگاه Viewer می‌بایست با چهار دستگاه جانبی آریا، ونیلاتور، مازول O3 و سانترال ارتباط برقرار کند و به تعامل با آن‌ها بپردازد؛ همانطور که در شکل مشاهده می‌کنید (شکل ۱.۳)، فقط ارتباط با دستگاه ونیلاتور یک‌طرفه است و مابقی ارتباطات با دستگاه‌های جانبی دیگر دوطرفه است.



شکل ۱.۳ نحوه تعامل Viewer با دستگاه‌ها جانبی.

نحوه برقراری ارتباط دستگاه Viewer با دستگاه‌های جانبی به شرح زیر است:

۱. جهت برقراری ارتباط با دستگاه جانبی آریا از پروتکل ارتباطی مربوط به آن استفاده می‌شود که

توسط جناب آقای رضا شالباف و دیگر همکارانشان در سال ۲۰۲۱ میلادی تهیه شده است. [۷]

۲. برای برقراری ارتباط با دستگاه جانبی Ventilator از پروتکل ارتباطی مربوط به آن استفاده می

شود که توسط جناب آقای حمید عزیززاده و همکارانشان در سال ۲۰۲۱ میلادی تهیه شده

است. [۸]

۳. برای برقراری ارتباط با دستگاه جانبی O3 از پروتکل ارتباطی مربوط به آن استفاده می‌شود که

توسط شرکت آمریکایی Masimo در سال ۲۰۱۸ میلادی تهیه شده است. [۹]

۴. جهت برقراری ارتباط با دستگاه جانبی سانترال (سیستم مانیتورینگ مرکزی) از پروتکل ارتباطی

مربوط به آن استفاده می‌شود که توسط سرکار خانم ناهید عسگری و همکارانشان در سال

۲۰۲۱ میلادی تهیه شده است. [۱۰]

در پروتکلهای ارتباطی توضیحات کاملی درمورد چگونگی نحوه برقراری ارتباطات و تعاملات با

دستگاه‌های جانبی داده شده است و در ادامه این بخش، توضیحاتی درمورد پروتکلهای ارتباطی با هر

کدام از دستگاه‌ها جانبی خواهیم داد.

۱.۱.۳ پروتکل ارتباطی با دستگاه آریا

داده‌های انتقالی دارای حجم زیادی هستند و به همین دلیل داده‌ها برای انتقال به بخش‌ها و بسته‌های کوچکتری تقسیم می‌شوند و سپس بین دستگاه‌ها منتقل می‌شوند. برخی از این داده‌ها در لحظه ارسال می‌شوند مثل داده‌های مربوط به سیگنال‌ها و پارامترهای زیر:

- سیگنال‌ها و پارامتر ECG
- پارامتر ST
- پارامتر PVCs
- سیگنال و پارامترهای SPO2
- سیگنال و پارامترهای IBP1
- سیگنال و پارامترهای IBP2
- سیگنال و پارامترهای IBP3
- سیگنال و پارامترهای IBP4
- سیگنال و پارامتر RESP
- سیگنال و پارامترهای EtCO2
- پارامترهای NIBP
- پارامترهای BFI
- پارامترهای TEMP

برخی دیگر از این داده‌ها فقط به هنگام تغییر ارسال می‌شوند، مثل داده‌های مربوط به تنظیمات دستگاه، تنظیمات سیگنال‌ها و پارامترها. داده‌هایی که فقط در هنگام تغییر ارسال می‌شوند، در زمان برقراری ارتباط به مدت پنج ثانیه به صورت مستمر ارسال می‌شوند. ساختار تمامی بسته‌های انتقالی در جدول (۱) نشان داده شده است و توضیح

هر بخش به شرح زیر است:

جدول ۱.Error! No text of specified style in document. ساختار بسته آریا.

Header	Len	Code1	Code2	Data	Checksum
0xAA,0x55	2 Byte(High,Low)	1 Byte	1 Byte	N Byte	1 Byte

- دو بایت به Header بسته اختصاص داده شده است که نمایانگر شروع بسته است، این دو بایت حاوی مقادیر 0xAA و 0x55 هگزادسیمال^۱ یا ۱۷۰ و ۸۵ دسیمال^۲ است.
- دو بایت دیگر به Len اختصاص داده شده است که نمایانگر طول داده‌های موجود در بسته است.
- به کد ۱ یک و کد ۲، هر کدام یک بایت اختصاص داده شده است که نمایانگر نوع بسته است.
- Data بسته حاوی مقادیر اصلی مربوط به بسته است که با توجه به کد ۱ و کد ۲، طول بسته متغیر است.
- یک بایت هم به Checksum اختصاص داده شده است که نمایانگر مجموع داده‌های بسته است.

۲.۱.۳ پروتکل ارتباطی با دستگاه ونتیلاتور

ارتباط با دستگاه جانبی ونتیلاتور یک طرفه است، یعنی داده‌ها فقط از سمت ونتیلاتور به Viewer منتقل می‌شوند و فرمت^۳ کلی بسته‌های ارسالی در جدول (document) نشان داده شده است و توضیح هر بخش به شرح زیر است:

جدول ۲.Error! No text of specified style in document. ساختار بسته ونتیلاتور.

^۱ hexadecimal

^۲ decimal

^۳ ساختار

Packet :	Header 0xAA (1 byte)	LEN N (1 byte)	Data XX (N bytes)	CRC XX (1 byte)
----------	----------------------------	----------------------	-------------------------	-----------------------

- Header: به صورت ثابت و برابر با 0xAA می‌باشد.
- LEN: تعداد بایت‌های موجود در قسمت Data که نمایانگر طول بسته است را مشخص می‌کند.
- Data: این قسمت حاوی بافر اصلی مورد نظر جهت ارسال می‌باشد.
- CRC: عنوان "کد آشکارساز خطأ" به انتهای بسته اضافه می‌شود و حاصل عملیات ریاضی بر روی تمامی داده‌های درون بسته (جز خودش) است. برای محاسبه CRC از روش معمول حاصل جمع اعداد با سرریز یک بایتی استفاده می‌شود یعنی فرمول محاسبه CRC به نحو زیر است:

$$\text{CRC} = \text{LEN} + \text{DATA}$$

Error! No text of specified Data بسته، شامل دو قسمت است که در جدول (۳) مشخص شده است:

جدول ۳. Error! No text of specified style in document. ساختار بخش داده بسته ونتیلاتور.

Code	Buffer
Data:	1 byte N-1 bytes

این که هر کد نمایانگر چه نوع بسته‌ای و شامل چه داده‌هایی است در جدول زیر (۴) قابل مشاهده است.

جدول ۴. Error! No text of specified style in document. کدهای بسته ونتیلاتور.

Code	Buffer (# of bytes)	N	interval
0x05	Introduction Text	101	10s
0x06	Signals	26	10 ms
0x07	Parameters	60	500 ms
0x09	Alarms	11	500 ms

0x0A	Hardware Errors	5	10s
0x0B	Algorithm Settings S1	25	2s
0x0C	Algorithm Settings S2	17	2s
0x0D	Alarm Setting	21	2s
0x0E	Apnea Setting	6	2s
0x0F	Reserved	5	2s

۳.۱.۳ پروتکل ارتباطی با مازول O3

ارتباط با مازول O3 دوطرفه است، یعنی داده‌ها هم از سمت Viewer به O3 ارسال می‌شود و هم از سمت O3 به سمت Viewer. فرمت کلی پکتهای انتقالی در جدول (۵) نشان داده شده است و توضیح هر بخش به شرح زیر است:

.O3 ۵.Error! No text of specified style in document. جدول ۵ ساختار بسته .O3

SOM	LEN1	LEN2	d1, d2, d3...dN	CRC1	CRC2	EOM
0xA1	N _{MSB}	N _{LSB}	Message bytes	R _{MSB}	R _{LSB}	0xAF

- بایت SOM و EOM که حاوی مقادیر 0xA1 و 0xAF هگزادسیمال یا ۱۶۱ و ۱۷۵ دسیمال هستند نمایانگر شروع و پایان بسته هستند.
- دو بایت به LEN اختصاص داده شده که نمایانگر طول^۱ Message است یعنی تعداد بایت‌هایی که بین LEN و CRC هستند.
- دو بایت به CRC^۲ اختصاص داده شده است که استاندارد CRC-16^۳ است یعنی جهت تشخیص خطاهای انتقال، یک جمع کنترلی تولید می‌کند.

^۱ Message یا پیام قابل انتقال

^۲ Cyclic Redundancy Check

^۳ یک نوع استاندارد و روش تشخیص خطا در بسته‌های شبکه.

• d1، d2، d3 تا dN مسیج یا داده‌هایی است که داخل بسته قرار گرفته است.

Error! No text of specified style in document. ساختار مسیج در بسته‌ها انتقالی، در جدول (۶) نشان داده شده است و به شرح زیر است:

جدول ۰۳. ۶ ساختار بخش داده بسته Error! No text of specified style in document.

SOM	LEN1	LEN2	d1	d2	d3...dN	CRC1	CRC2	EOM
0xA1	N _{MSB}	N _{LSB}	CMD _{MSB}	CMD _{LSB}	Message bytes	R _{MSB}	R _{LSB}	0xAF

SOM	LEN1	LEN2	d1	d2	d3...dN	CRC1	CRC2	EOM
0xA1	N _{MSB}	N _{LSB}	RSP _{MSB}	RSP _{LSB}	Message bytes	R _{MSB}	R _{LSB}	0xAF

دو بایت به CMD و RSP اختصاص داده شده است که نمایانگر شناسه‌ی بسته انتقالی فرمان^۱ و پاسخ^۲ هستند و مابقی بایت‌ها، داده‌های دیگر مسیج هستند. نرخ انتقال داده‌ها بین ۱۶ میلی ثانیه تا یک دقیقه است.

۴.۱.۳ پروتکل ارتباطی با سانترال

Error! No text of specified style in document. ارتباط با سانترال نیز دوطرفه است. ساختار بسته‌های انتقالی در جدول (۷) نشان داده شده است:

جدول ۷. ۷ ساختار بسته سانترال Error! No text of specified style in document.

^۱ Command

^۲ Response

Header byte #1(0x55)	Header byte #2(0xAA)	Header byte #3(0x00)	Header byte #4(0x99)
Code 1 (High byte)	Code 1(Low byte)	Length 1(High byte)	Length 1(Low byte)
Data			
Code 2 (High byte)	Code 2(Low byte)	Length 2 (High byte)	Length 2(Low byte)
Data			
0xFF	0xFF		

- چهار بایت به Header اختصاص داده شده است که نمایانگر شروع بسته است.
- هر بسته شامل چند المان است و هر المان شامل بخش‌های Code و Length است.
- Code نمایانگر نوع المان است مثلا: ECG, SPO2 و ...
- Length نمایانگر طول داده‌های المان است.
- هم داده‌های اصلی المان است.
- دو بایت هم در انتهای قرار دارد که نمایانگر پایان هر بسته است.
- نرخ انتقال داده‌ها بین چهل میلی ثانیه تا یک ثانیه است.

۲.۳ نیازمندی‌ها

در کنار هر نرمافزاری اسنادی وجود دارد که نیازمندی‌های سیستم را نشان می‌دهد؛ مثل اسناد SRS¹[20], SDD²[19], SAD³[18]، URS⁴[17] و دیاگرام‌ها که در این بخش می‌خواهیم این نیازمندی‌ها را بررسی کنیم. در سند URS، از دیدگاه کاربر سیستم مشخص می‌شود که کاربر چه

1 user requirement(s) specification

2 Software Architecture Document

3 software design description

4 software requirements specification

(**\Error! No text of specified style in document.**) همان‌طور که در جدول (**\Error! No text of specified style in document.**) مشاهده می‌کنید، نیازها در این سند به صورت کاملاً مختصر و تک خطی مشخص می‌شود.

جدول **\Error! No text of specified style in document.** سند **URS** نرم‌افزار.

R#	Requirement Name	Function Description	Verification Reference
R1	Display Orientation	User can work with system vertically and horizontally	SIT ⁵
R2	Monitoring Aria's Modules	Receive and monitor parameters, signals and alarms related to Aria's modules	SIT
R3	Send And Receive Aria's Settings	Send and receive modules setting packets, bed setting packets and patient setting packets	SIT
R4	Customize Page	User can add/remove module(parameter & signal) on screen	SIT
R5	Communicate With Central	Communicate with the central and send some modules data to that	SIT
R6	Changing Network Setting By Password	User enters password to open network setting menu for changing network settings like UDP port, TCP port, Central IP & Base IP.	SIT
R7	Storing Aria's Data	Save Aria's signals, parameters and alarm in SqlServer database	SIT
R8	Aria's Signals History	User can monitor signals of last 96 hours	SIT
R9	Aria's Parameters History	User can monitor parameters of last 96 hours	SIT

⁵ system integration testing

R#	Requirement Name	Function Description	Verification Reference
R10	Aria's Alarms History	User can monitor alarms of last 96 hours	SIT
R11	Aria's Nibps History	User can monitor nibps of last 96 hours	SIT
R12	Aria's Arrhythmia History	User can monitor arrhythmias of last 96 hours	SIT
R13	Taking Screenshot	User can take a pictures from the screen and see them later	SIT
R14	Connection with Ventilator	Receive and monitor parameters, signals and alarms related to Ventilator	SIT
R15	Connection with O3	Receive and monitor parameters, signals and alarms related to O3	SIT

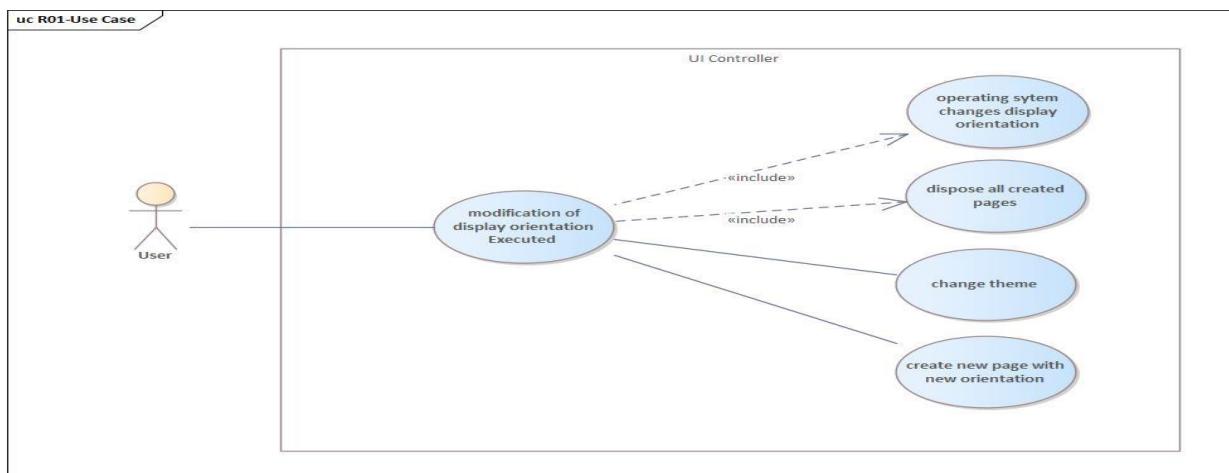
هر کدام از ستون‌های سند URS که در جدول (۸) است، به شرح زیر است:

- در ستون نخست شماره نیازمندی مشخص شده است.
- در ستون دوم عنوان نیازمندی درج شده است.
- در ستون سوم توضیح مختصری راجب به نیازمندی داده شده است.
- در ستون چهارم واحد تایید کننده نیازمندی‌ها مشخص می‌شود.

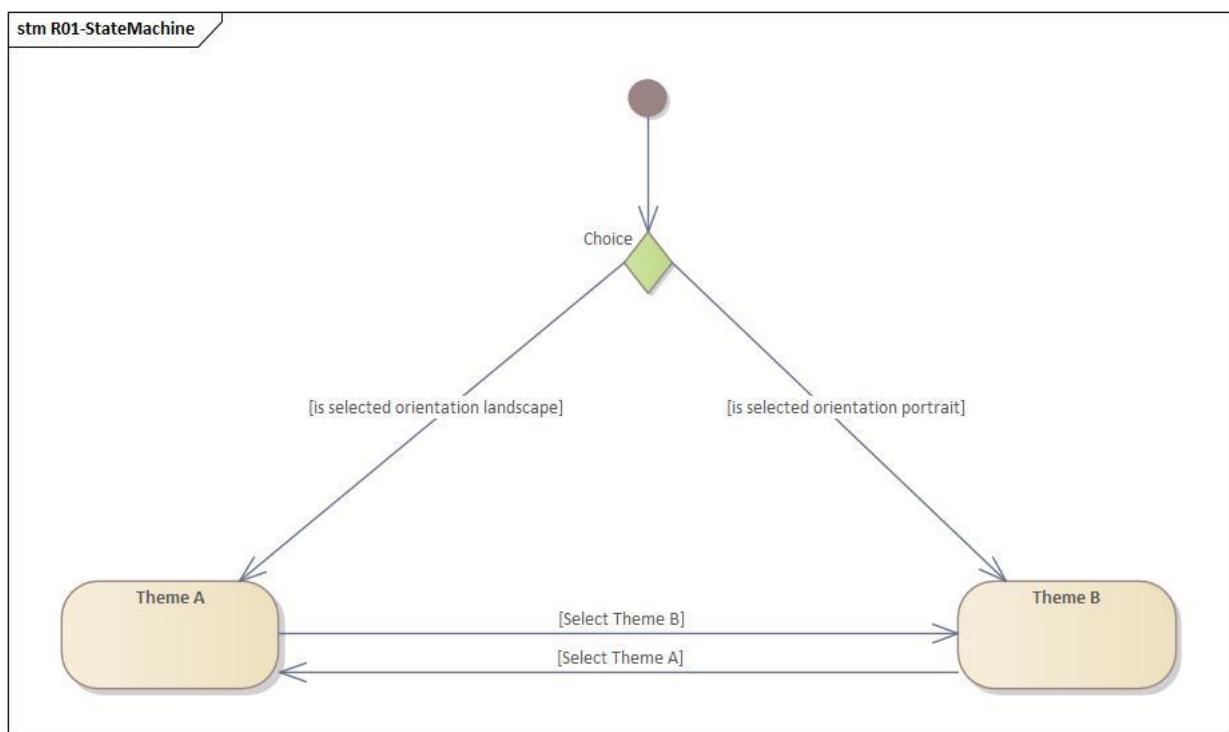
در ادامه این بخش به هر کدام از نیازمندی‌های سیستم می‌پردازیم.

Display Orientation ۱.۲.۳

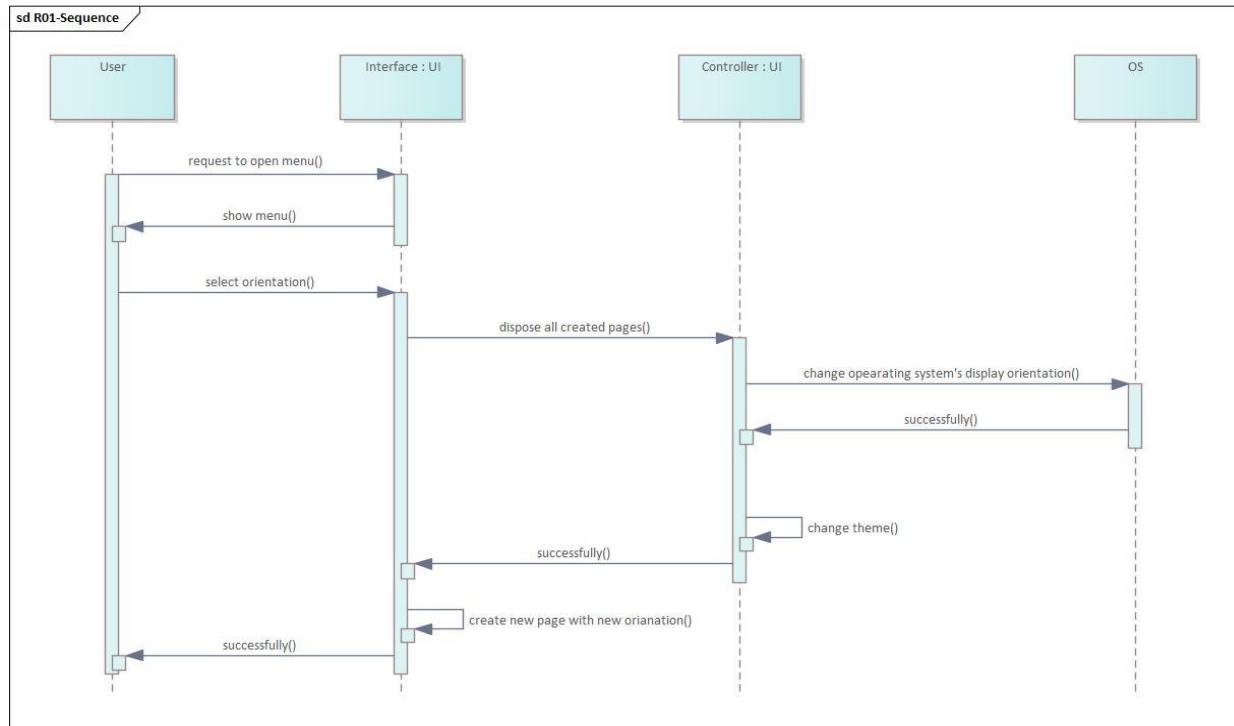
در این نیازمندی کاربر قصد دارد که با صفحه نمایش، هم به صورت افقی و هم به صورت عمودی قادر به کار باشد. دیاگرام‌های مربوط به این نیازمندی را مشاهده می‌کنید:



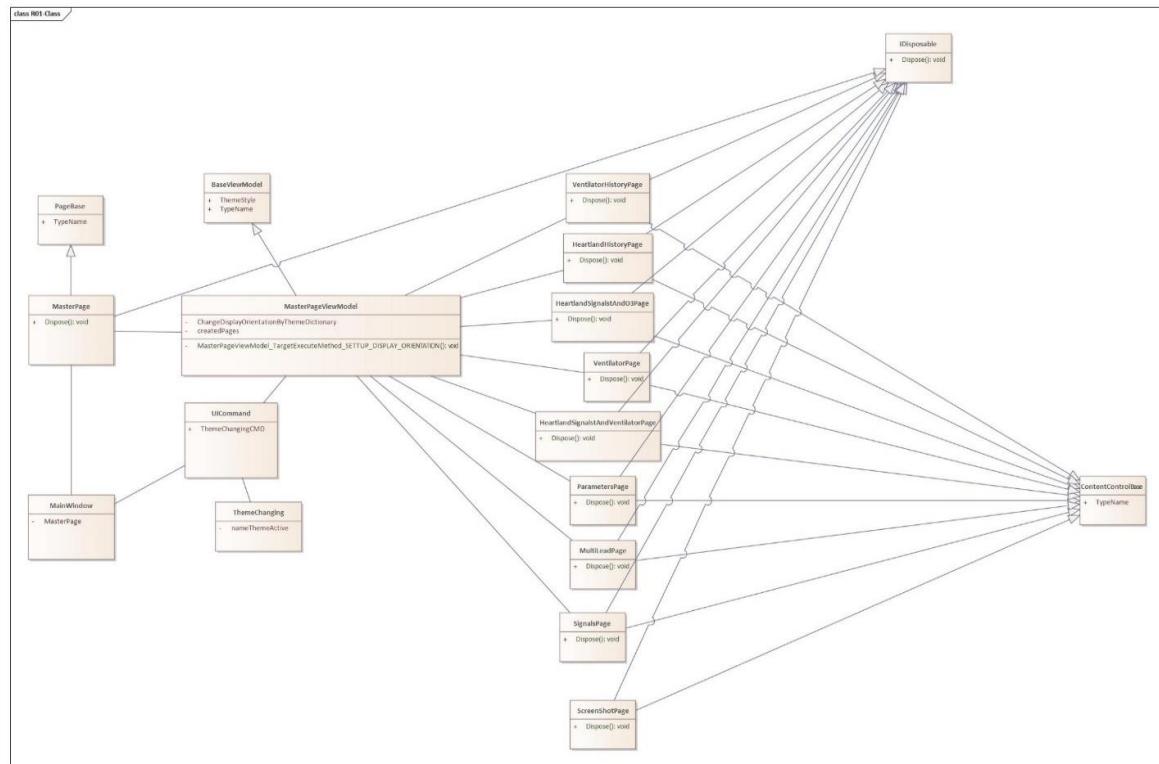
R01-Display Orientation Use Case Diagram ۲.۳



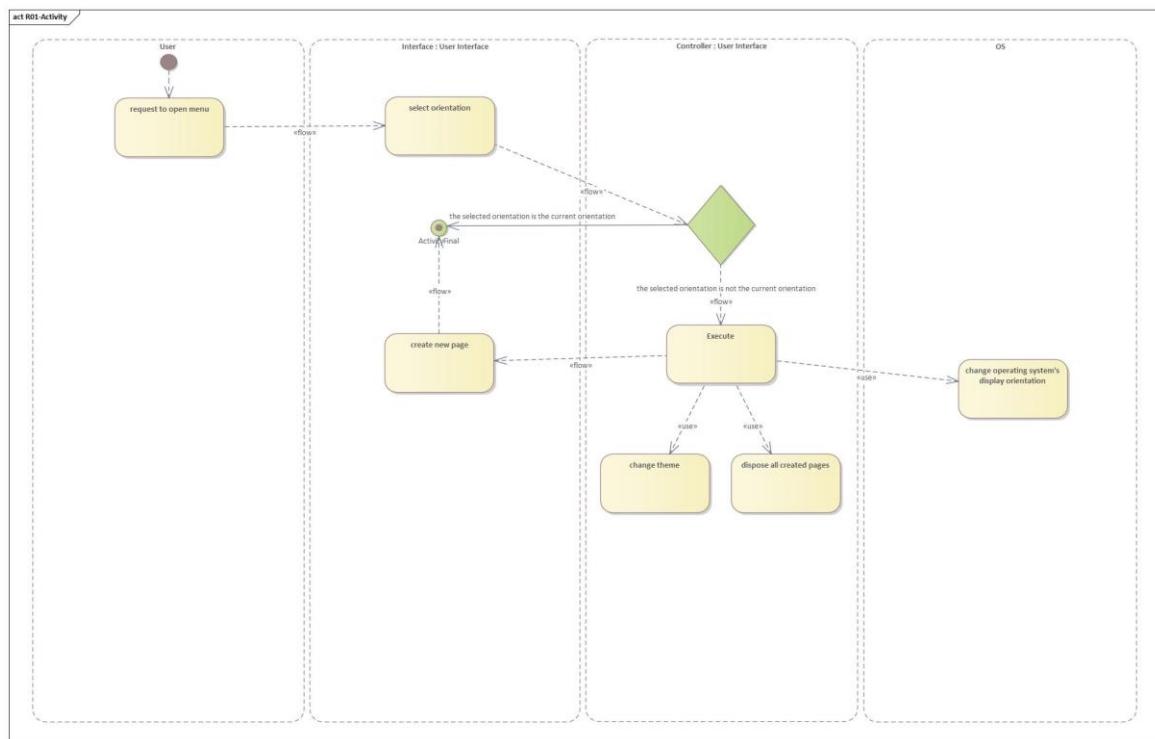
R01-Display Orientation State Machine Diagram ۳.۳



شکل ۴.۳ R01-Display Orientation Sequence Diagram



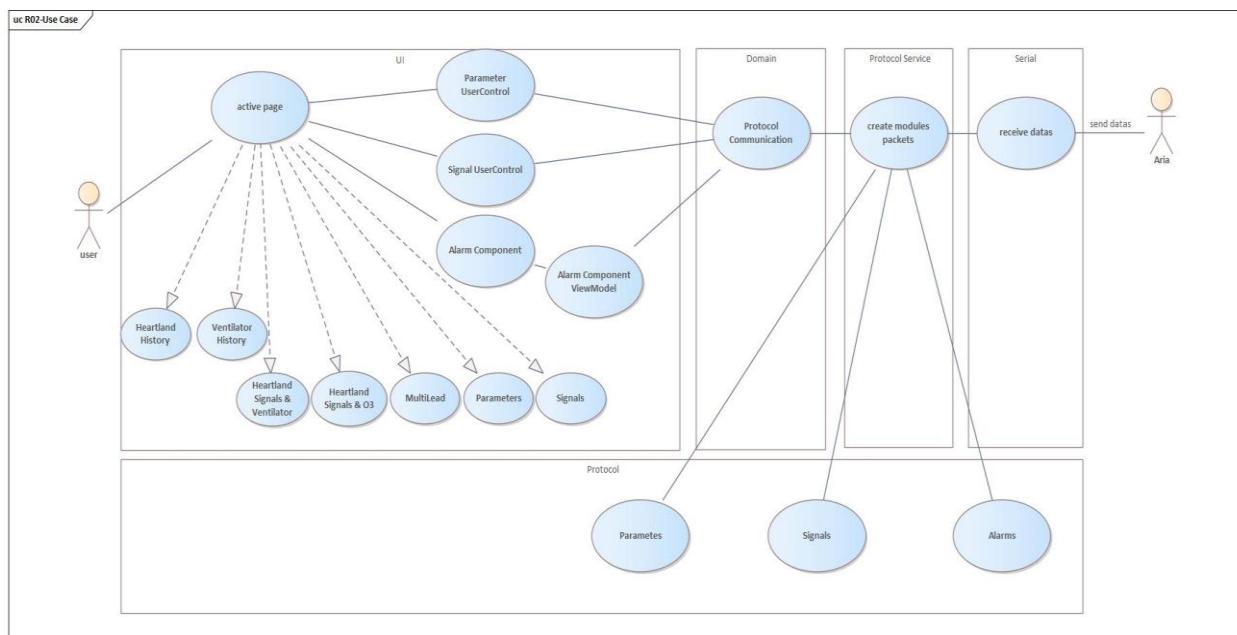
R01-Display Orientation Class Diagram ٥.٣ شکل



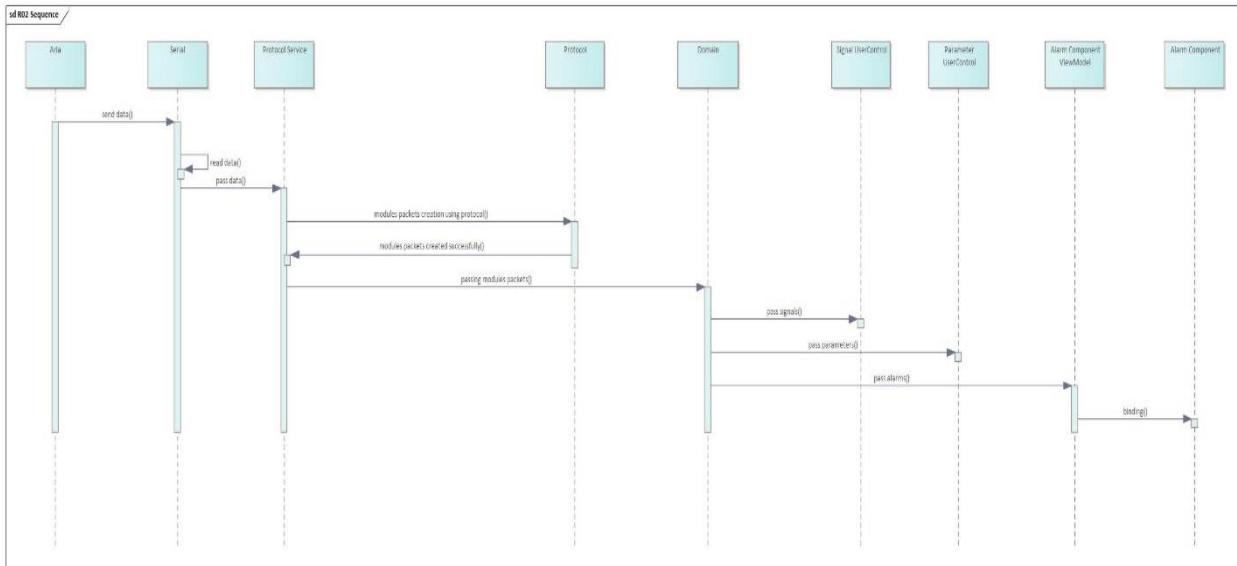
شكل ٦.٣ R01-Display Orientation Activity Diagram

Monitoring Aria's Modules ۲.۲.۳

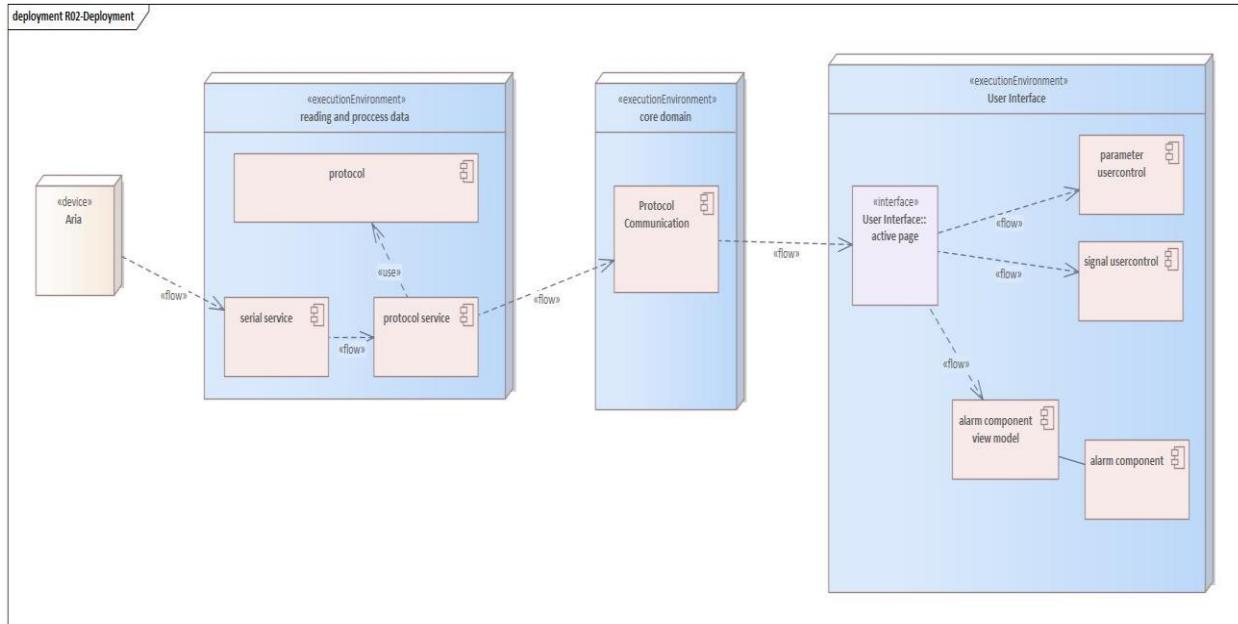
در این نیازمندی کاربر تمایل دارد که بتواند تمام اطلاعات مربوط به مازولهای آریا را در Viewer نظارت کند. دیاگرامهای مربوط به این نیازمندی را مشاهده میکنید:



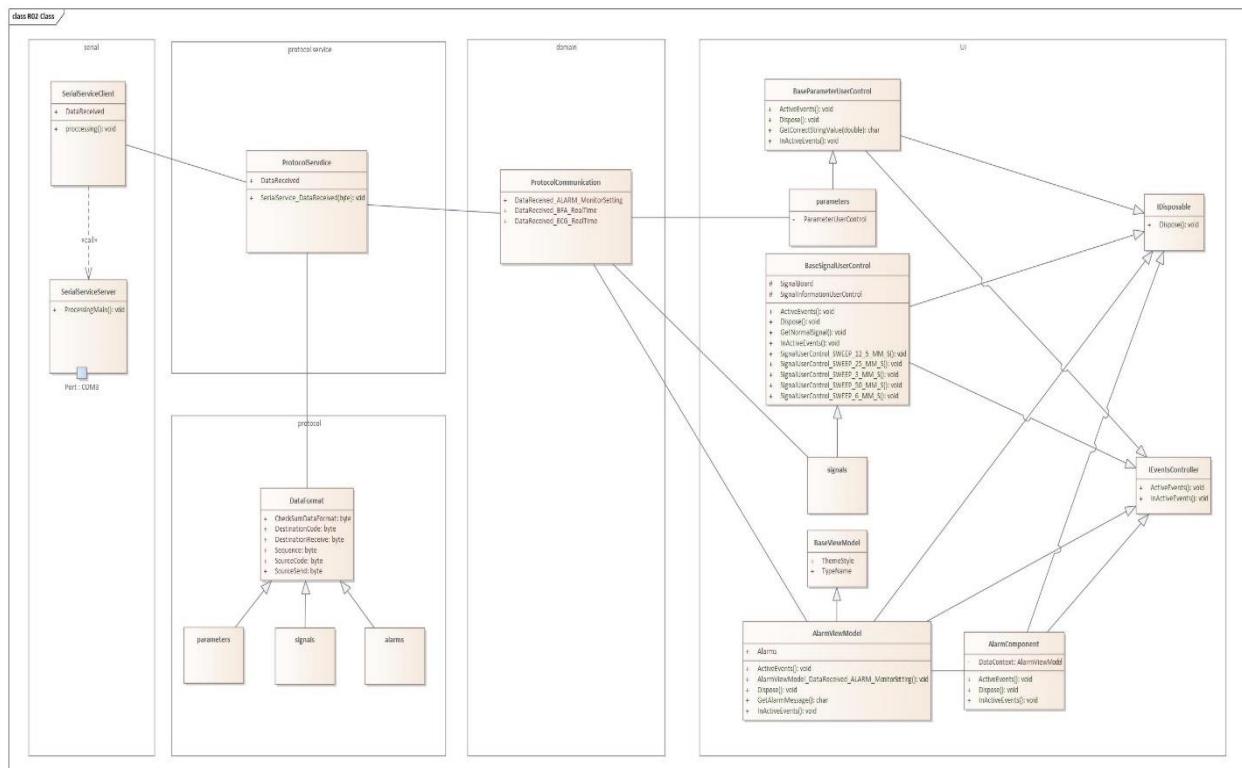
شكل ۷.۳ R02-Monitoring Aria’s Modules Use Case Diagram



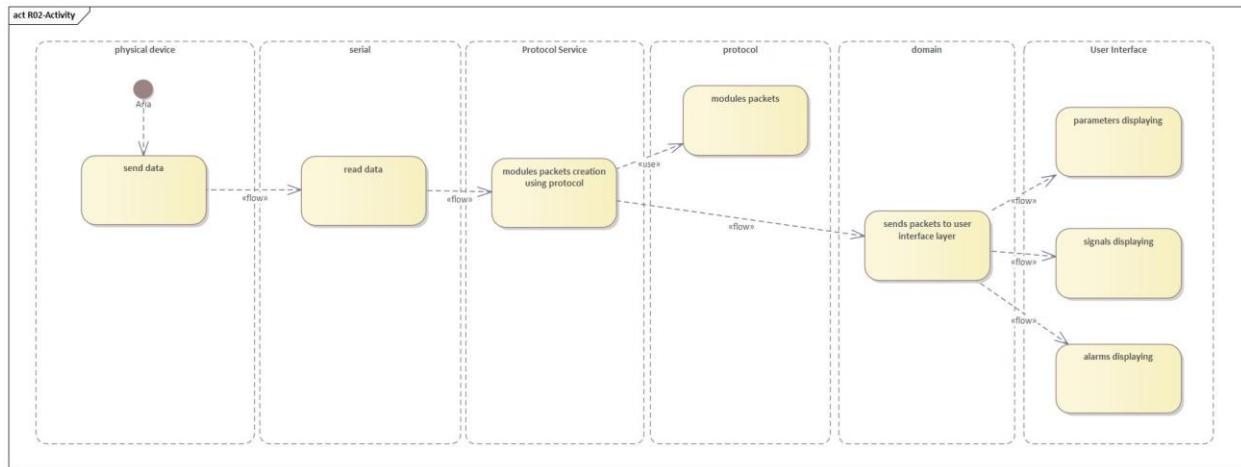
شكل ۸.۳ R02-Monitoring Aria’s Modules Sequence Diagram



شكل ٩.٣ R02-Monitoring Aria's Modules Deployment Diagram ٩.٣



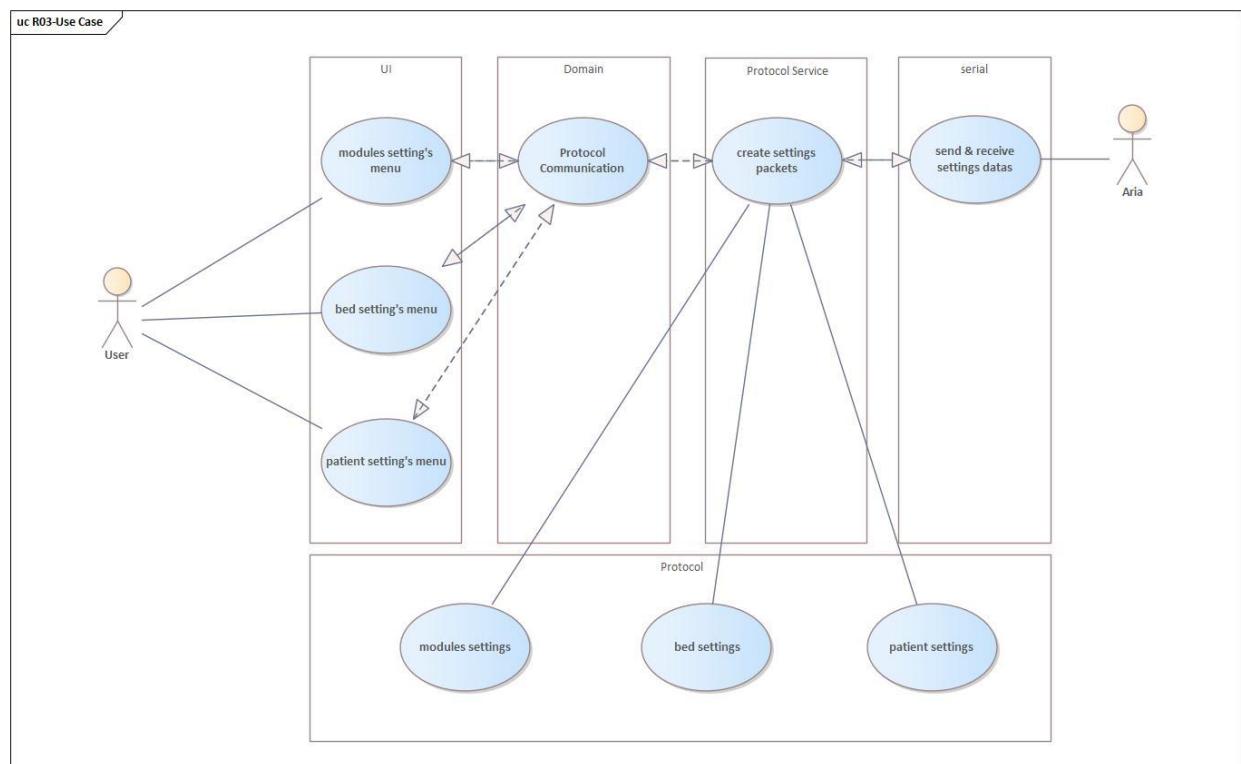
شكل ١٠.٣ R02-Monitoring Aria's Modules Class Diagram ١٠.٣



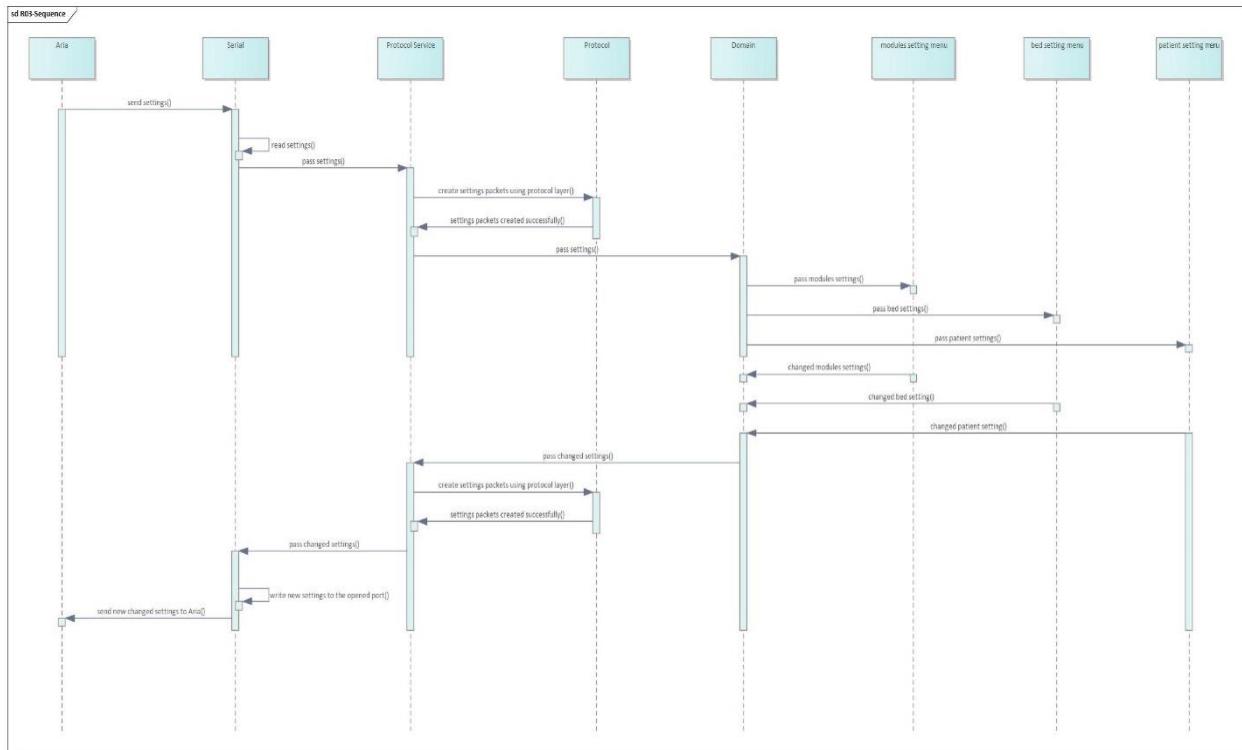
R02-Monitoring Aria's Modules Activity Diagram ۱۱.۳

Send And Receive Aria's Settings ۳.۲.۳

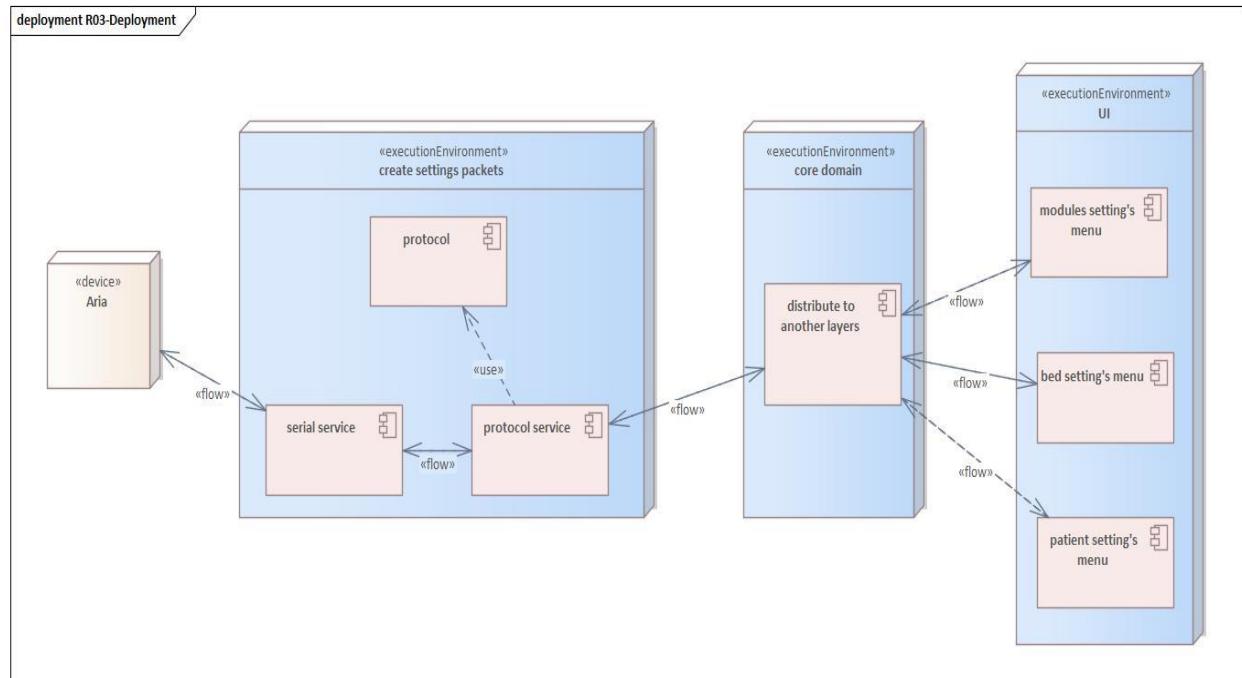
در این نیازمندی کاربر می‌خواهد که بتواند با ارتباط دوطرفه هم تنظیمات را ارسال کند و هم دریافت کند. دیاگرام‌های مربوط به این نیازمندی را مشاهده می‌کنید:



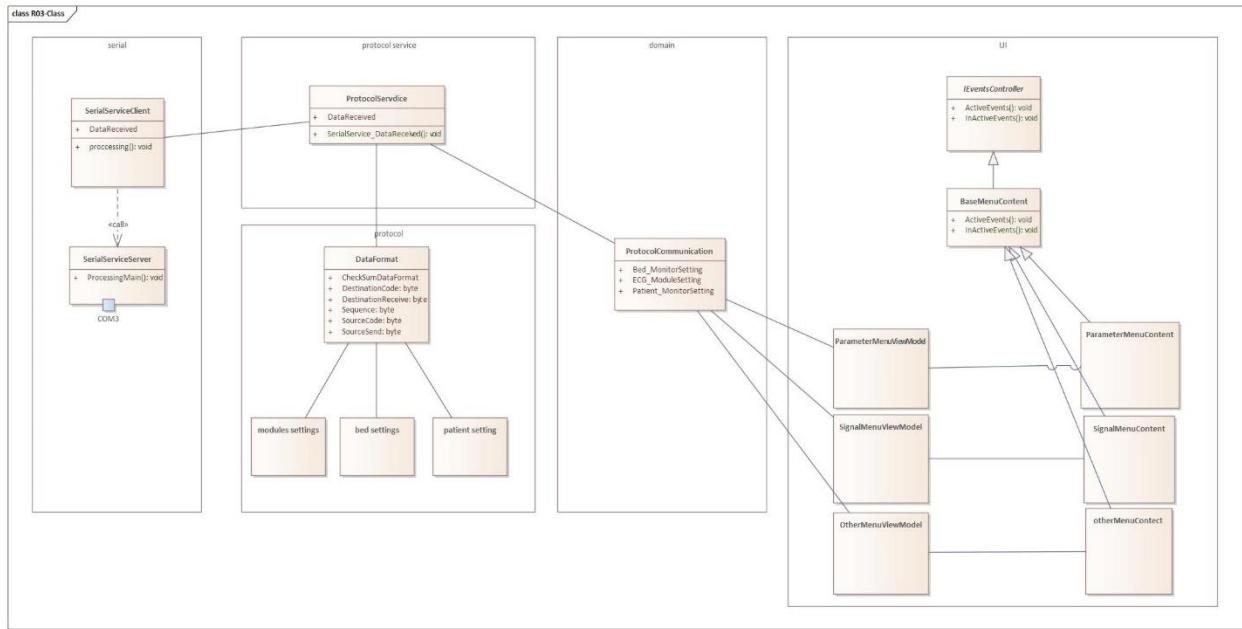
R03-Send And Receive Aria's Settings Use Case Diagram ۱۲.۳



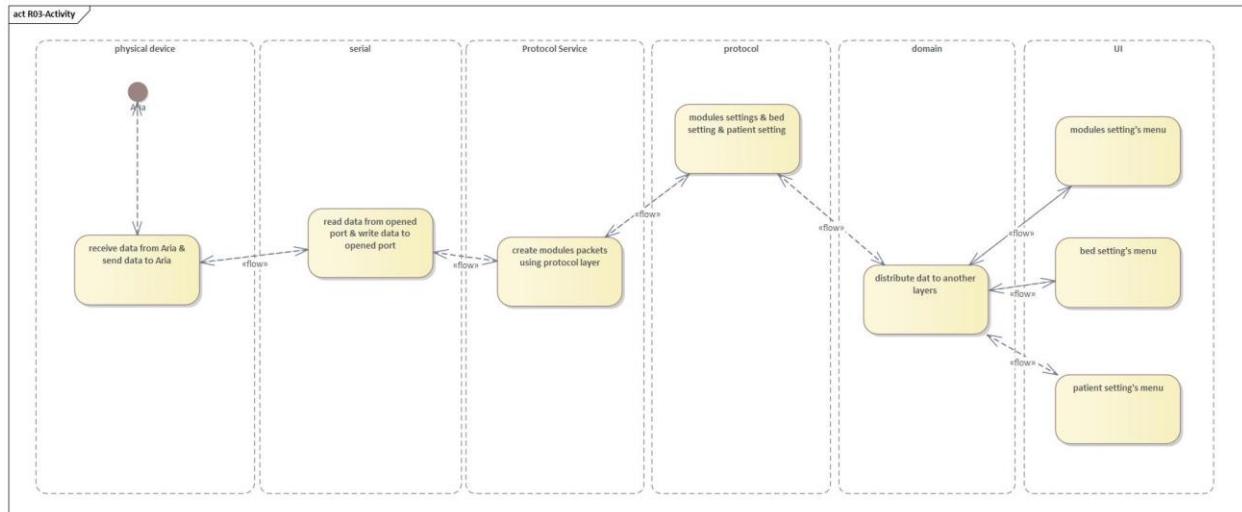
شکل ۱۳.۲ R03-Send And Receive Aria's Settings Sequence Diagram



شکل ۱۴.۳ R03-Send And Receive Aria's Settings Deployment Diagram



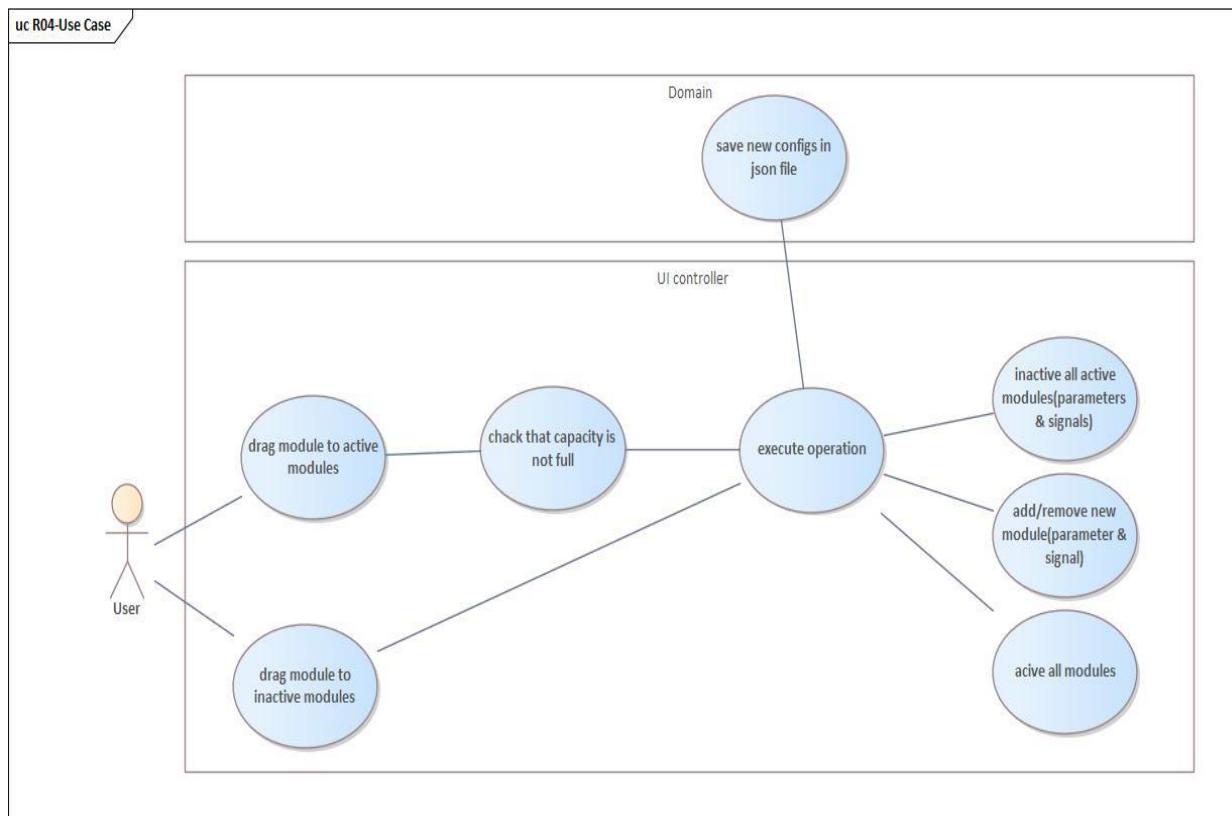
شکل ۱۵.۳ R03-Send And Receive Aria's Settings Class Diagram



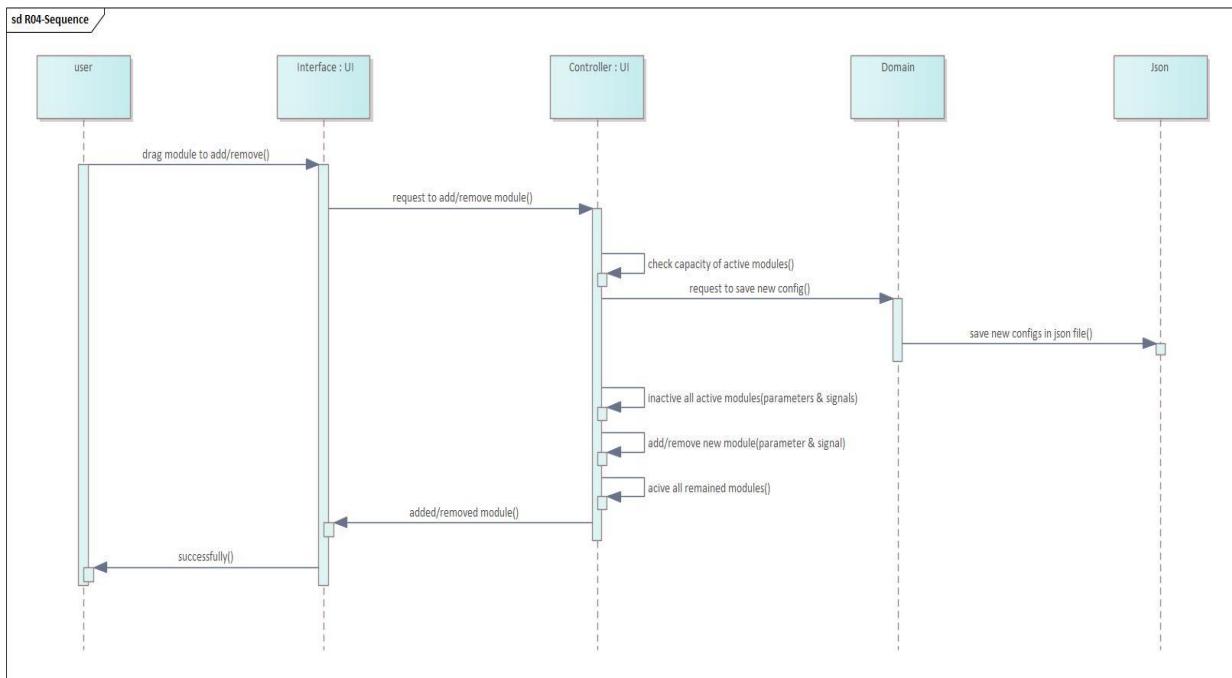
شکل ۱۶.۳ R03-Send And Receive Aria's Settings Activity Diagram

Customize Page ۴.۲.۳

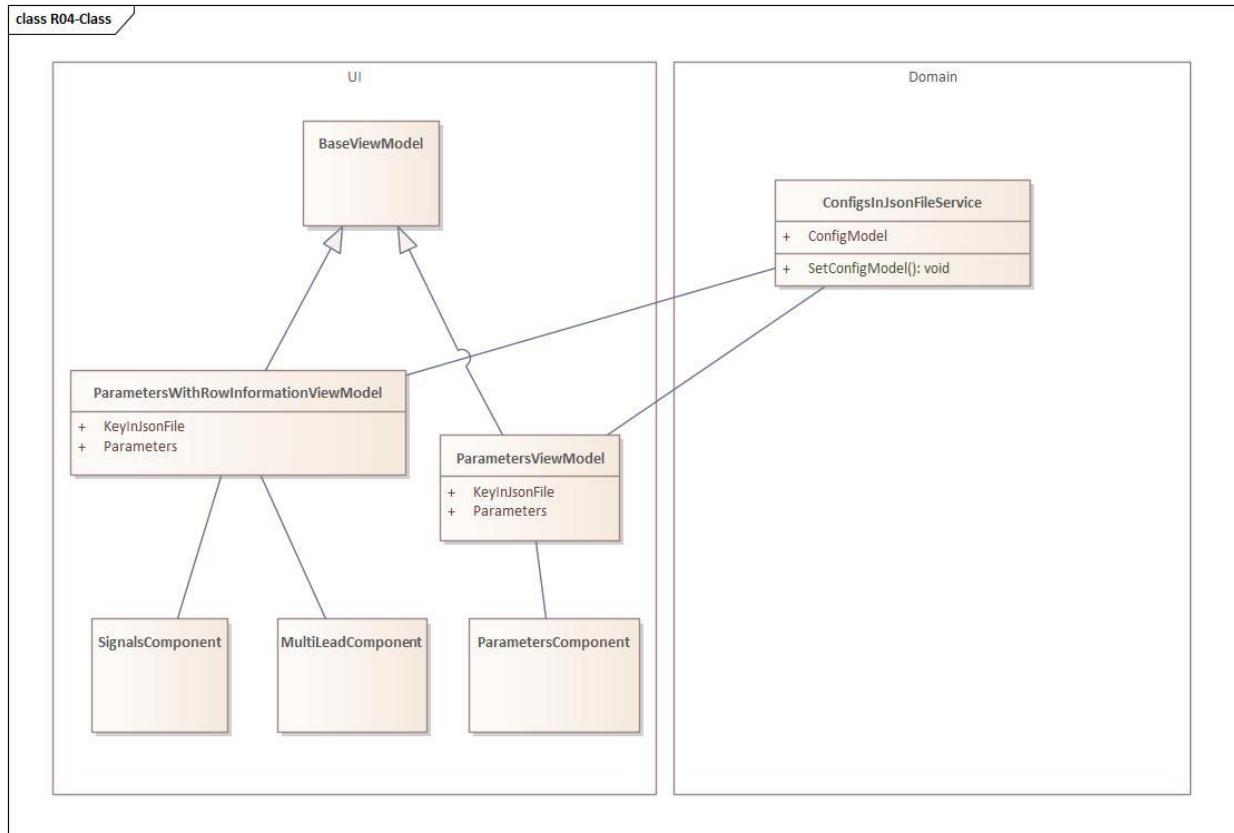
در این نیازمندی کاربر تمایل دارد که مطابق با نیاز خود صفحه را شخصی‌سازی کند، مازول‌ها را کم و زیاد کند و انتخاب کند که چه پارامترها و سیگنال‌هایی را می‌خواهد نظارت کند. دیاگرام‌های مربوط به این نیازمندی را مشاهده می‌کنید:



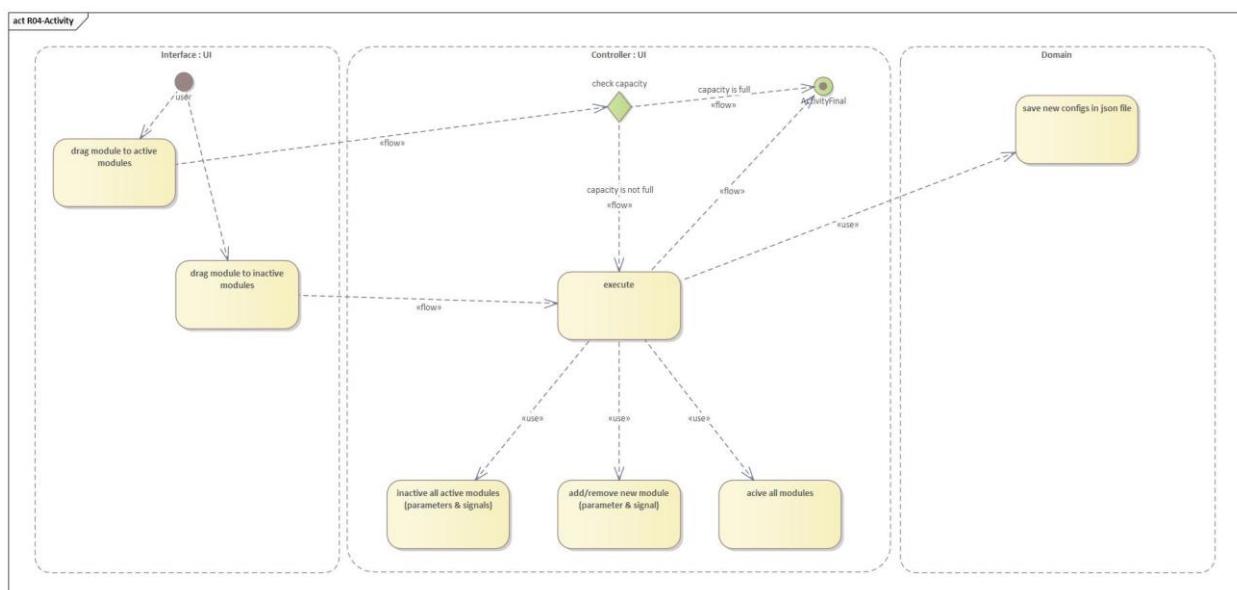
شکل ۱۷.۳ R04-Customize Page Use Case Diagram



شکل ۱۸.۳ R04-Customize Page Sequence Diagram



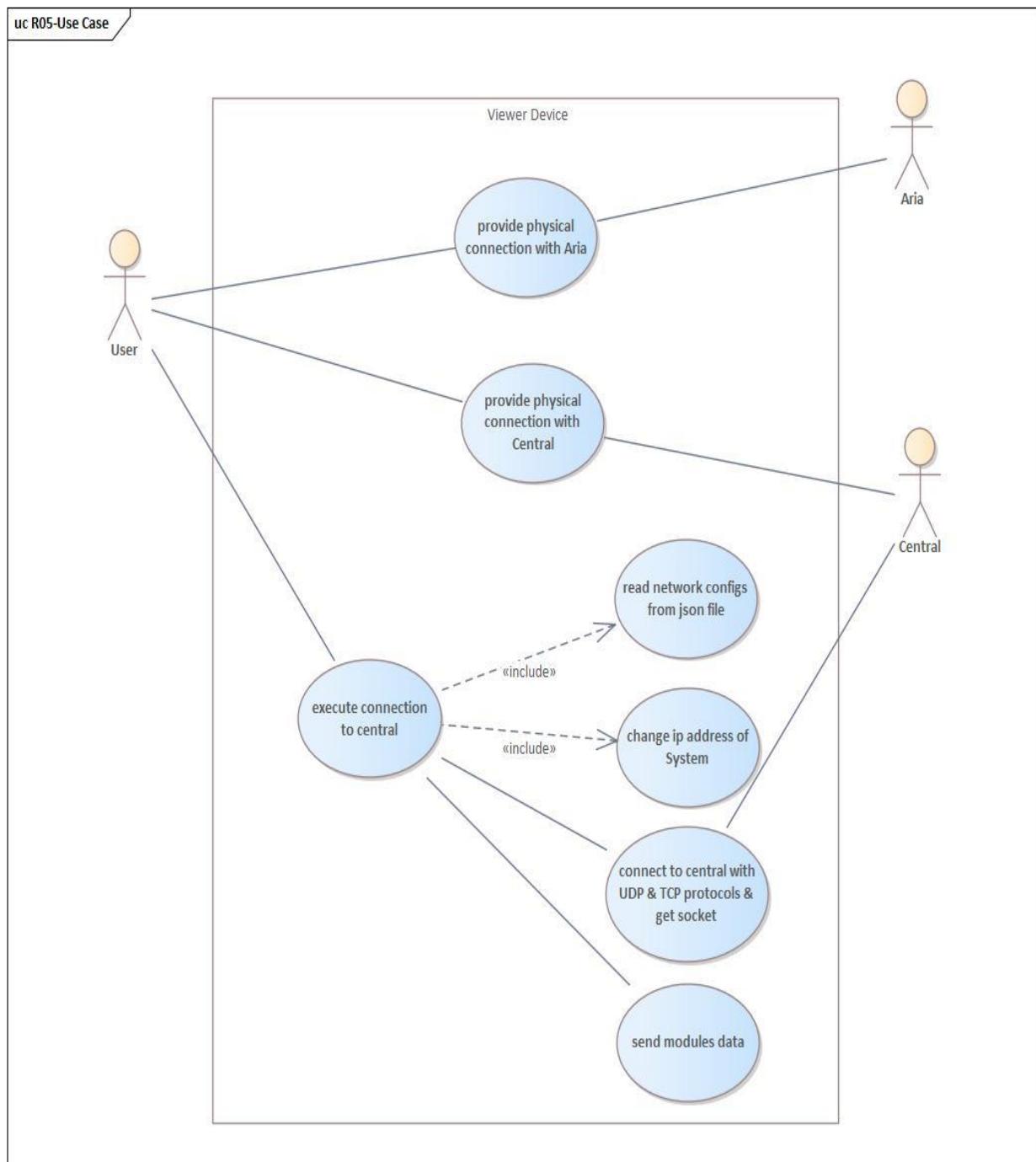
شکل ۱۹.۳ .R04-Customize Page Class Diagram



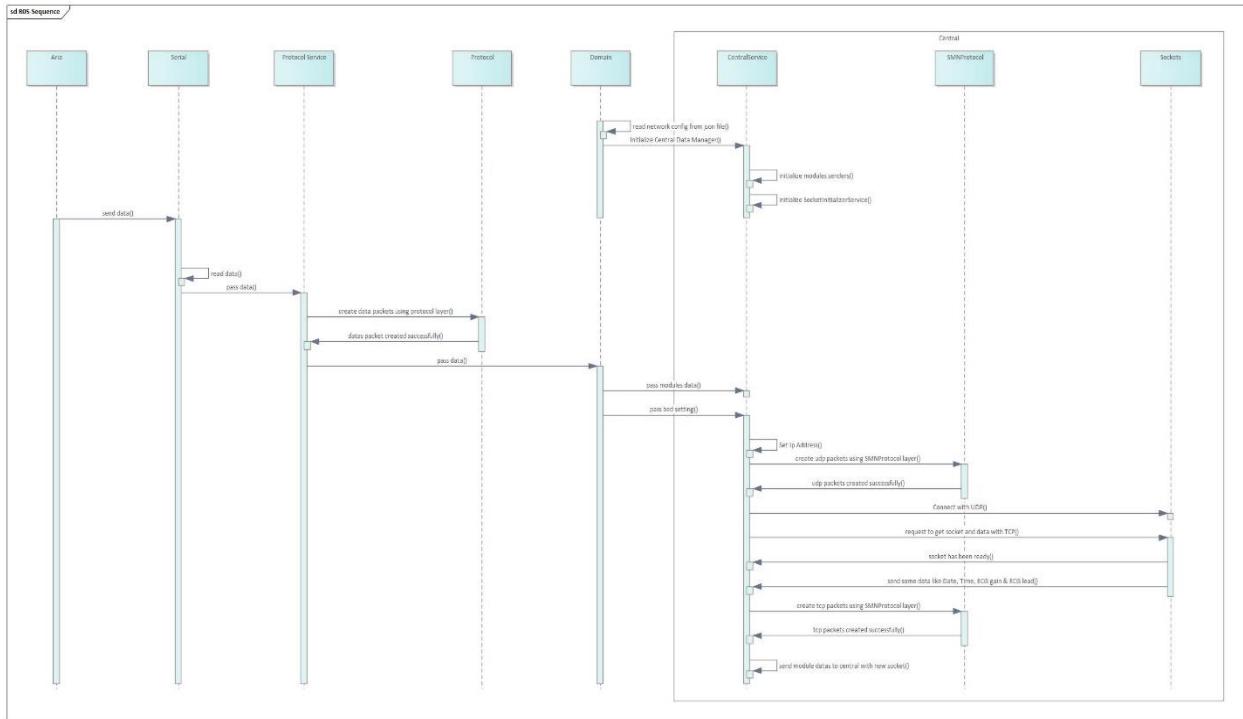
شکل ۲۰.۳ .R04-Customize Page Activity Diagram

Communication With Central ۵.۲.۳

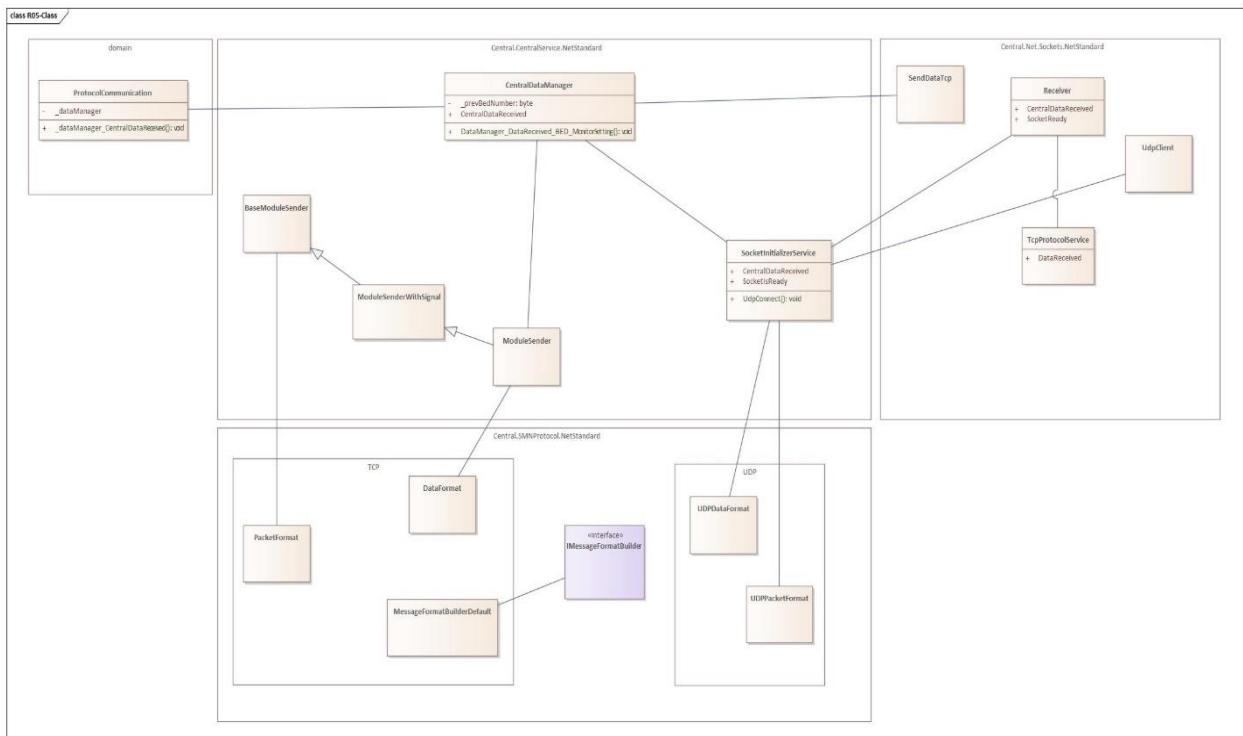
در این نیازمندی کاربر قصد دارد که بتواند با سیستم مانیتورینگ مرکزی یا سانترال ارتباط برقرار کند. دیاگرام‌های مربوط به این نیازمندی را مشاهده می‌کنید:



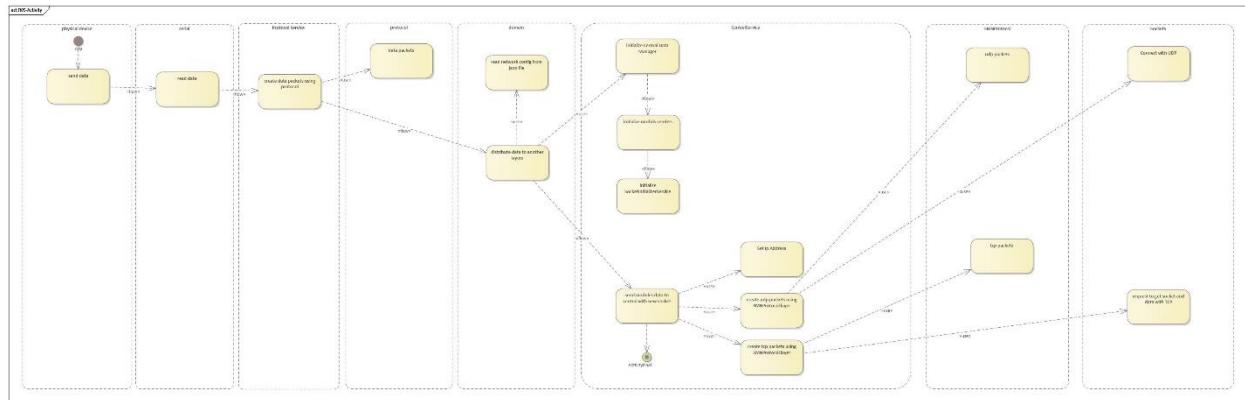
R05-Communicate With Central Use Case Diagram ۲۱.۳



شکل ۲۲.۳ R05-Communicate With Central Sequence Diagram



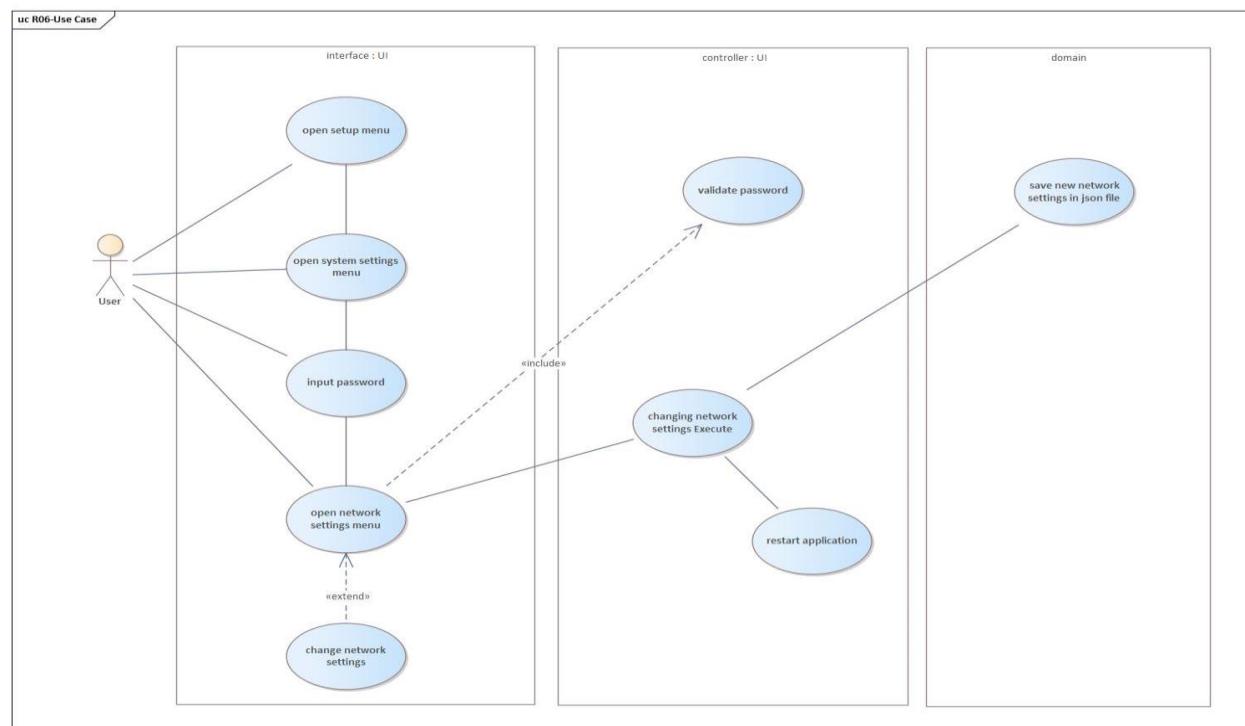
شکل ۲۲.۴ R05-Communicate With Central Class Diagram



شکل ۲۴.۳ R05-Communicate With Central Activity Diagram

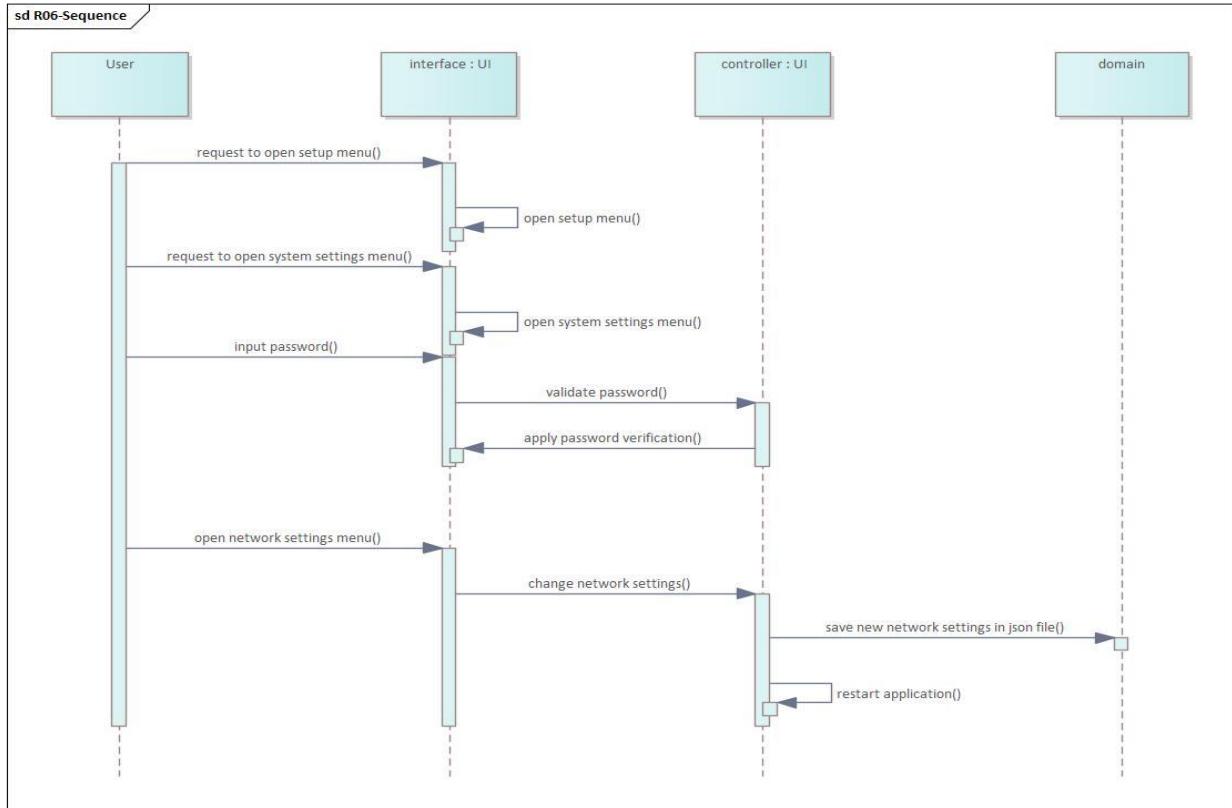
۶.۲.۳ Changing Network Setting By Password

در این نیازمندی کاربر می‌خواهد که بتواند با وارد کردن رمزعبور، تنظیمات شبکه از قبیل IP، شماره درگاه^۱ و ... را ویرایش کند. دیاگرام‌های مربوط به این نیازمندی را مشاهده می‌کنید:

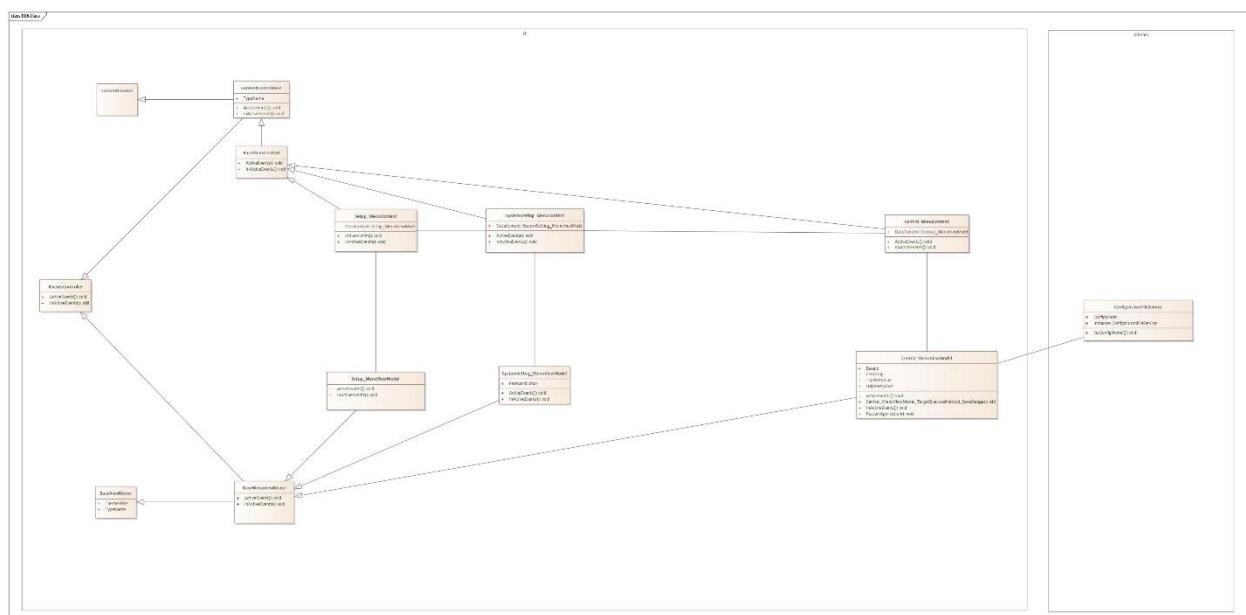


شکل ۲۵.۳ R06-Changing Network Setting By Password Use Case Diagram

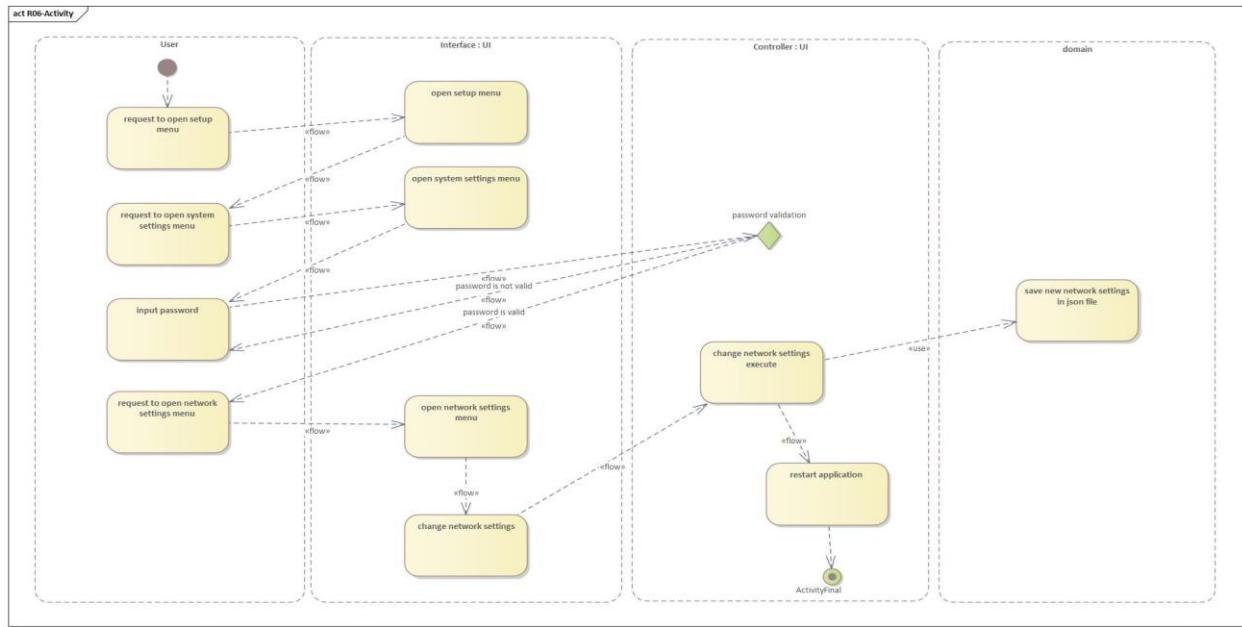
^۱ پورت



شکل ۲۶.۳ R06-Changing Network Setting By Password Sequence Diagram



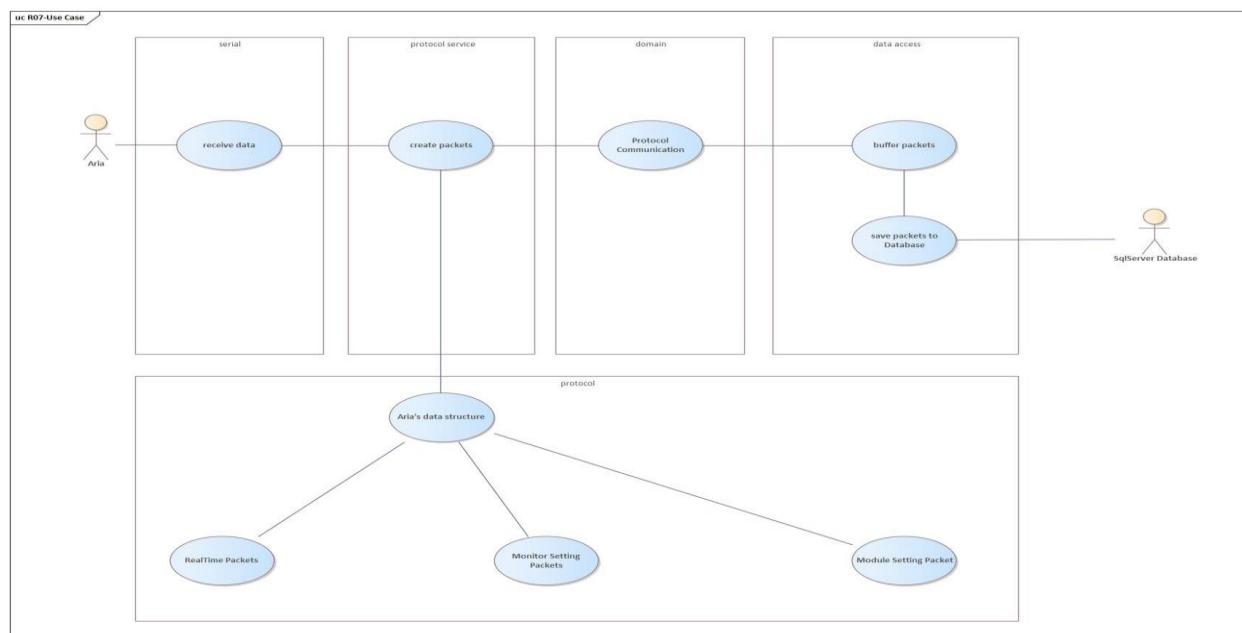
شکل ۲۷.۳ R06-Changing Network Setting By Password Class Diagram



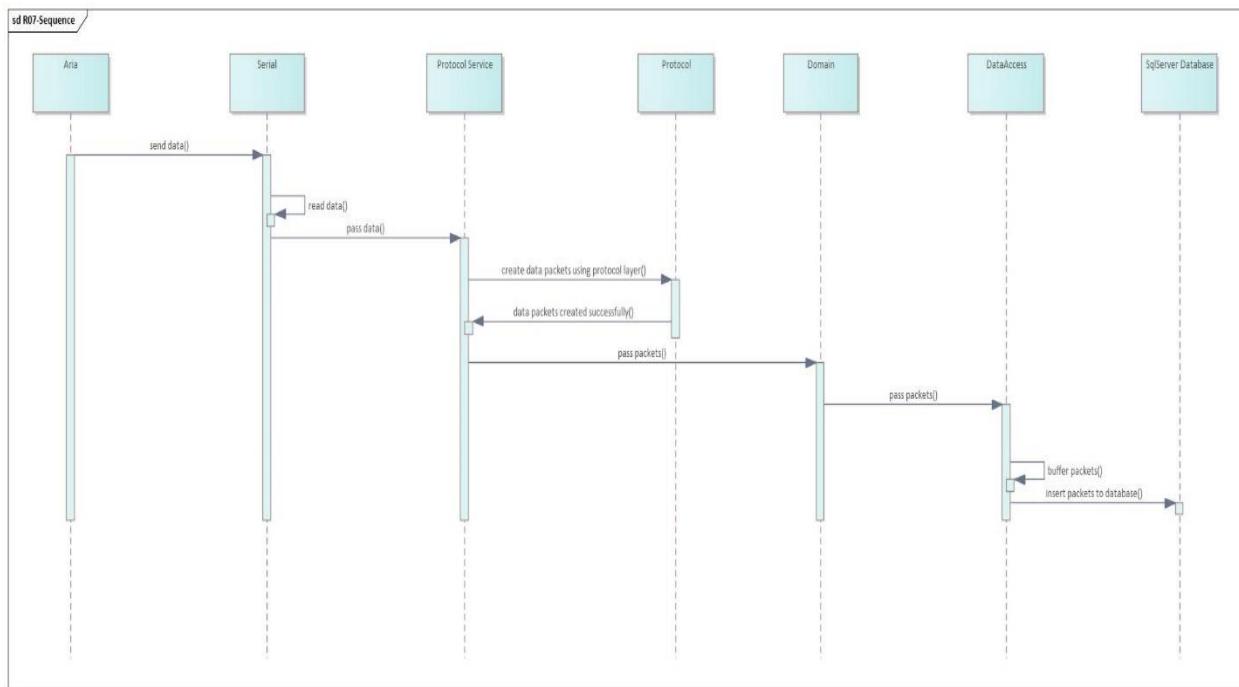
شکل ۲۸.۳ R06-Changing Network Setting By Password Activity Diagram

Storing Aria's Data ۷.۲.۳

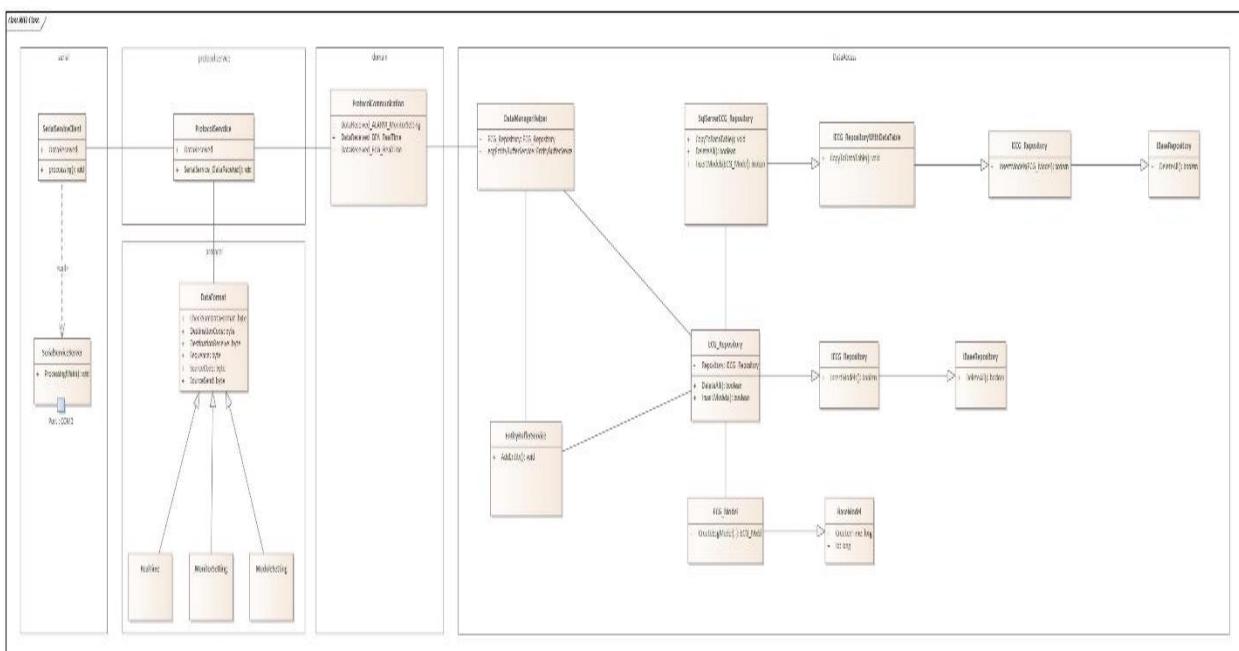
در این نیازمندی کاربر می‌خواهد که داده‌های مربوط به سیگنال‌ها، پارامترها و هشدارها در پایگاهداده ذخیره شود. دیاگرام‌های مربوط به این نیازمندی را مشاهده می‌کنید:



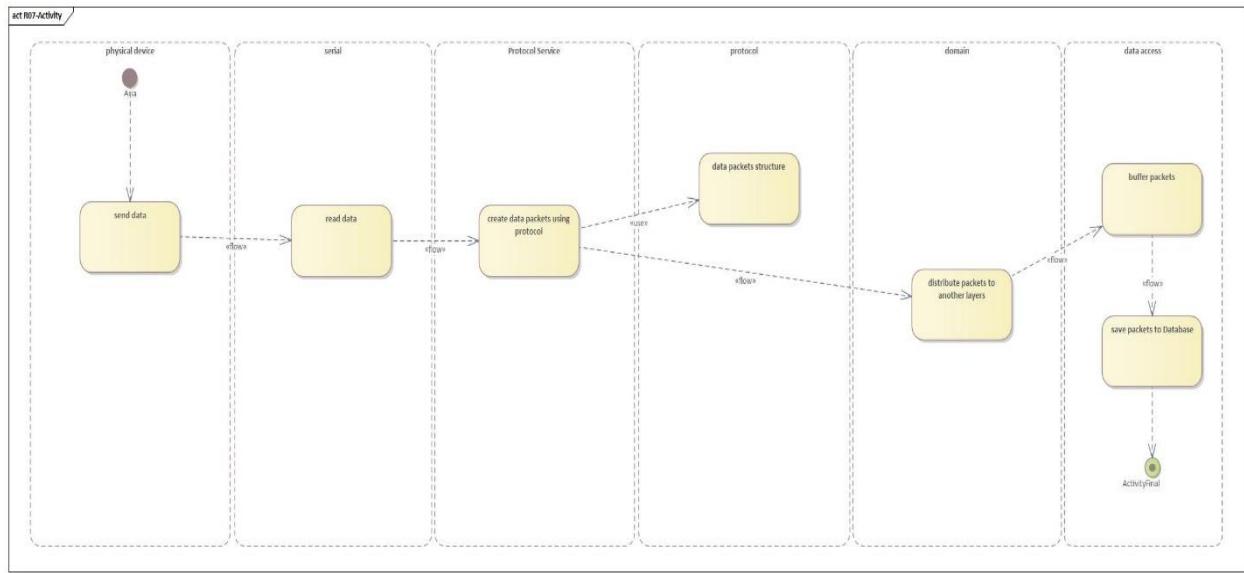
شکل ۲۹.۳ R07-Storing Aria's Data Use Case Diagram



R07-Storing Aria's Datas Sequence Diagram ٣٠.٣ شكل



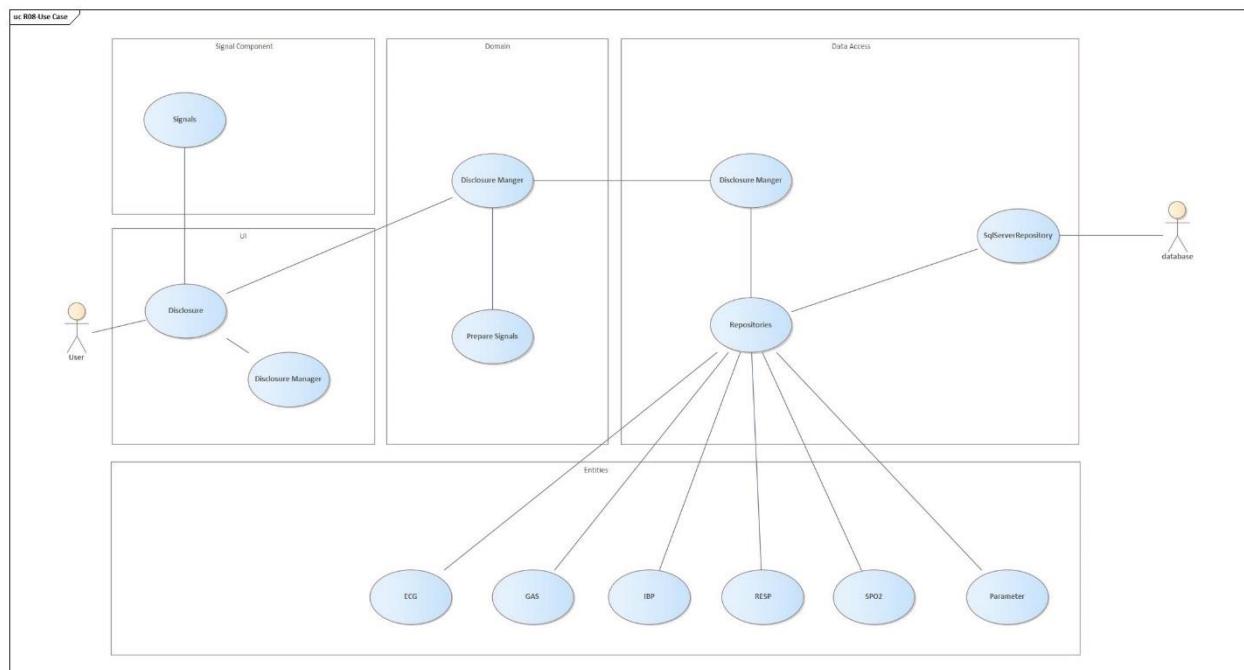
شکل ۳۱.۳ R07-Storing Aria's Datas Class Diagram



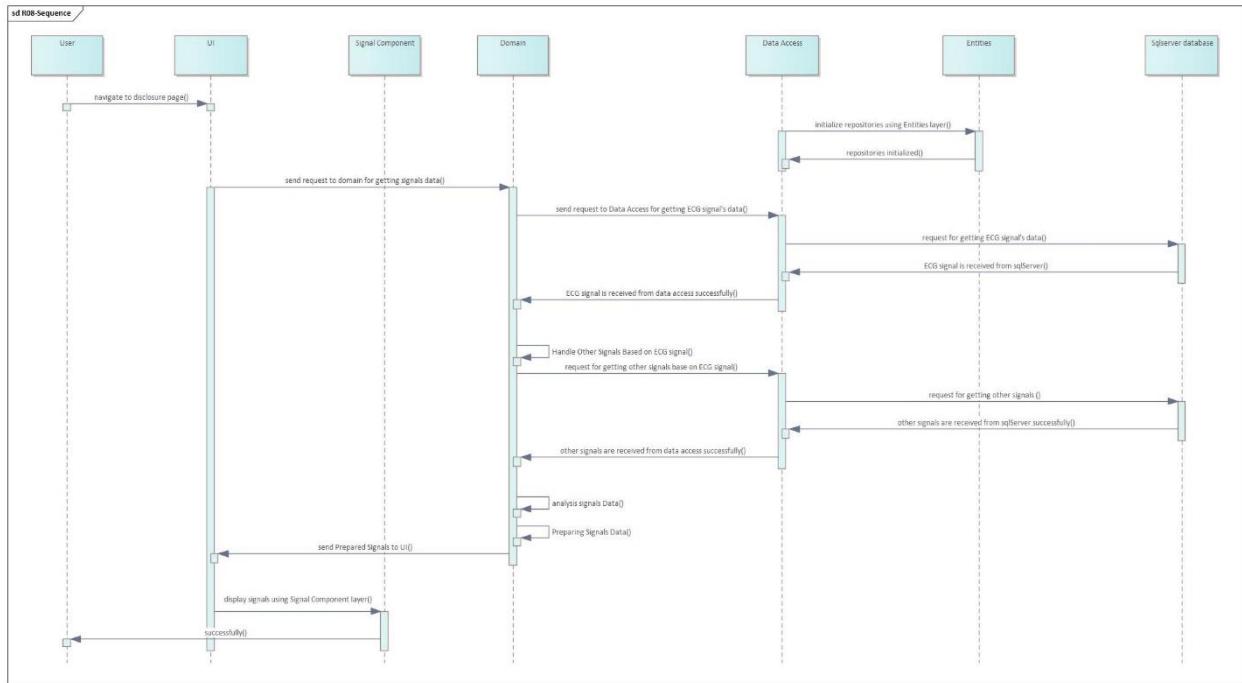
شکل ۳۲.۳ .R07-Storing Aria's Datas Activity Diagram ۳۲.۳

Aria's Signals History ۸.۲.۳

در این نیازمندی کاربر می‌خواهد که بتواند درون نرم‌افزار تمام اطلاعات مربوط به سیگنال‌های علائم حیاتی ذخیره شده آریا را نظارت کند. دیاگرام‌های مربوط به این نیازمندی را مشاهده می‌کنید:

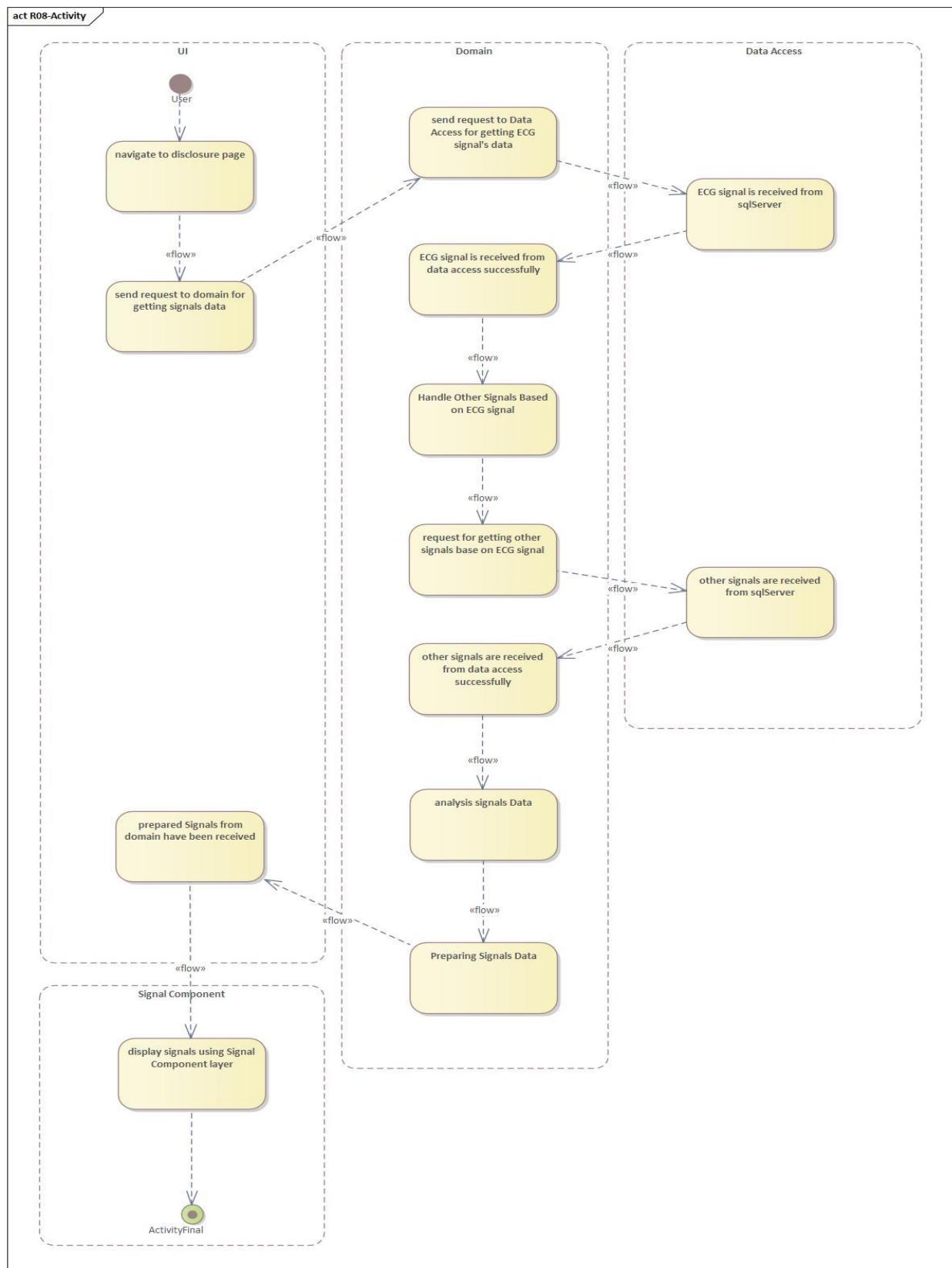


شکل ۳۳.۳ .R08-Aria's Signals History Use Case Diagram ۳۳.۳



R08-Aria's Signals History Sequence Diagram شکل ۳۴.۳

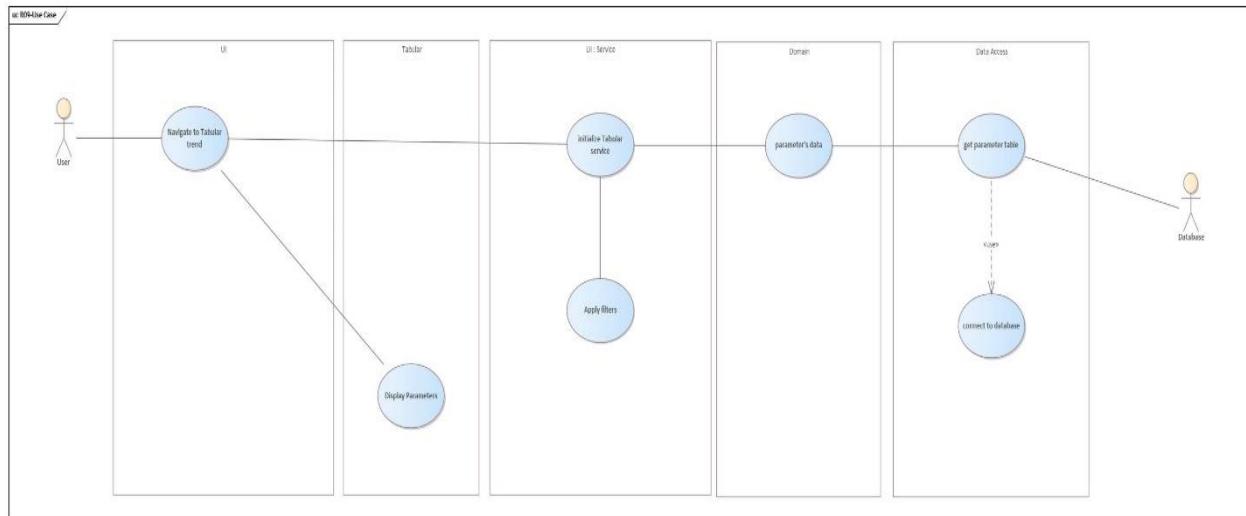
شکل ۳۵.۳ R08-Aria's Signals History Class Diagram



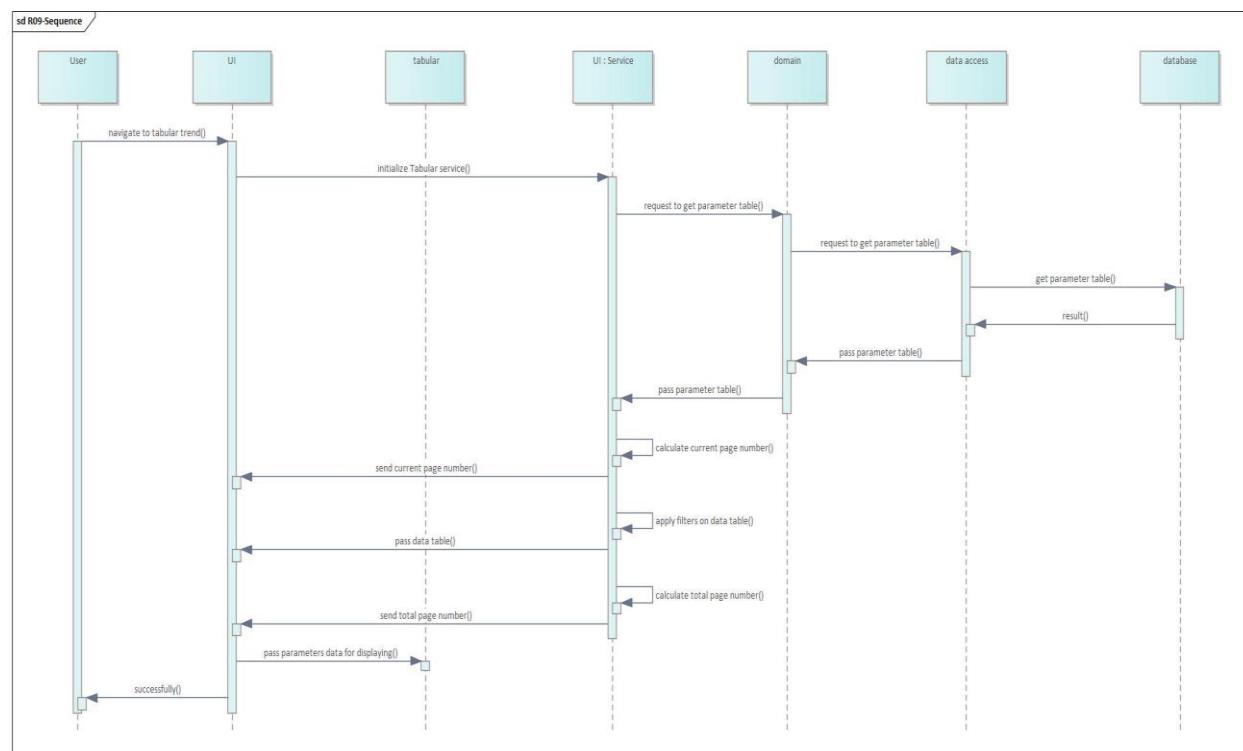
شکل ۳۶.۳ R08-Aria's Signals History Activity Diagram

Aria's Parameters History ۹.۲.۳

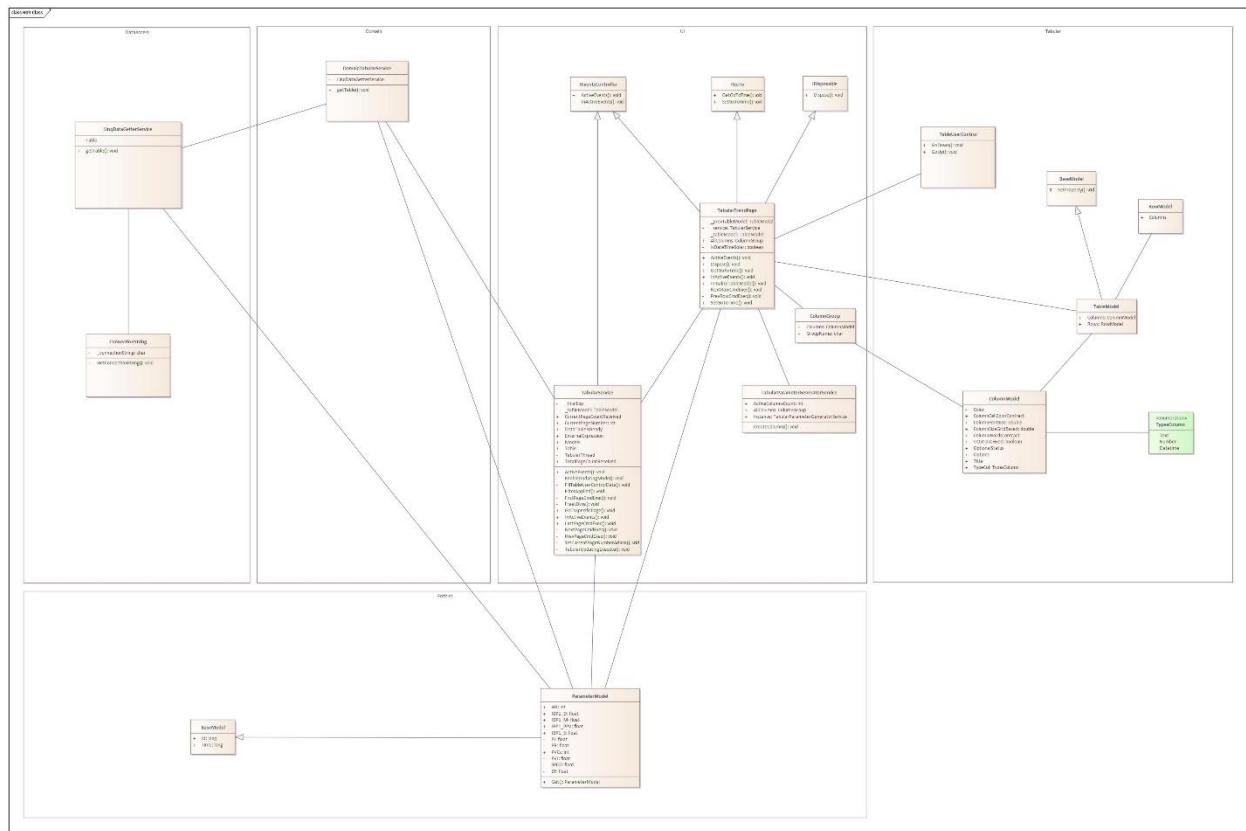
در این نیازمندی کاربر قصد دارد که بتواند درون نرم‌افزار تمام اطلاعات مربوط به پارامترهای علائم حیاتی ذخیره شده آریا را نظارت کند. دیاگرام‌های مربوط به این نیازمندی را مشاهده می‌کنید:



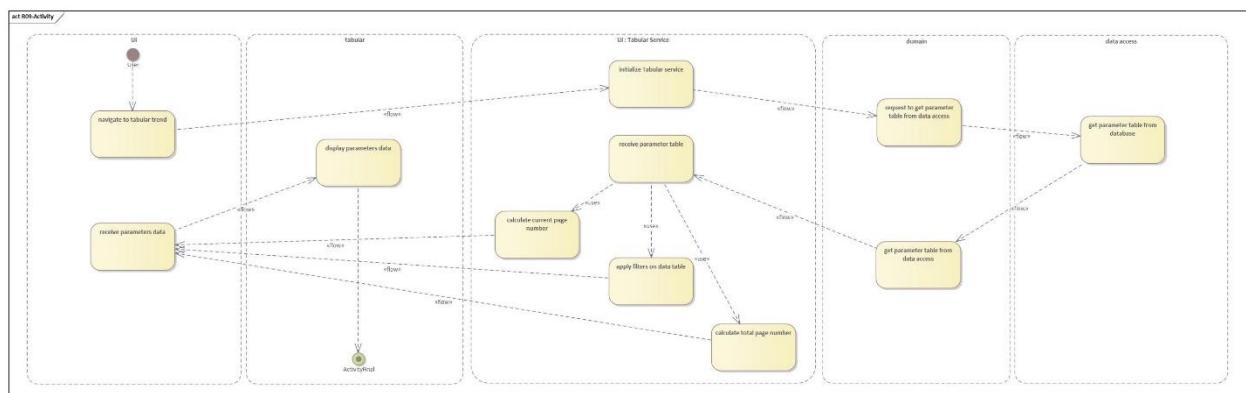
شکل ۳۷.۳ R09-Aria's Parameters History Use Case Diagram



شکل ۳۸.۳ R09-Aria's Parameters History Sequence Diagram



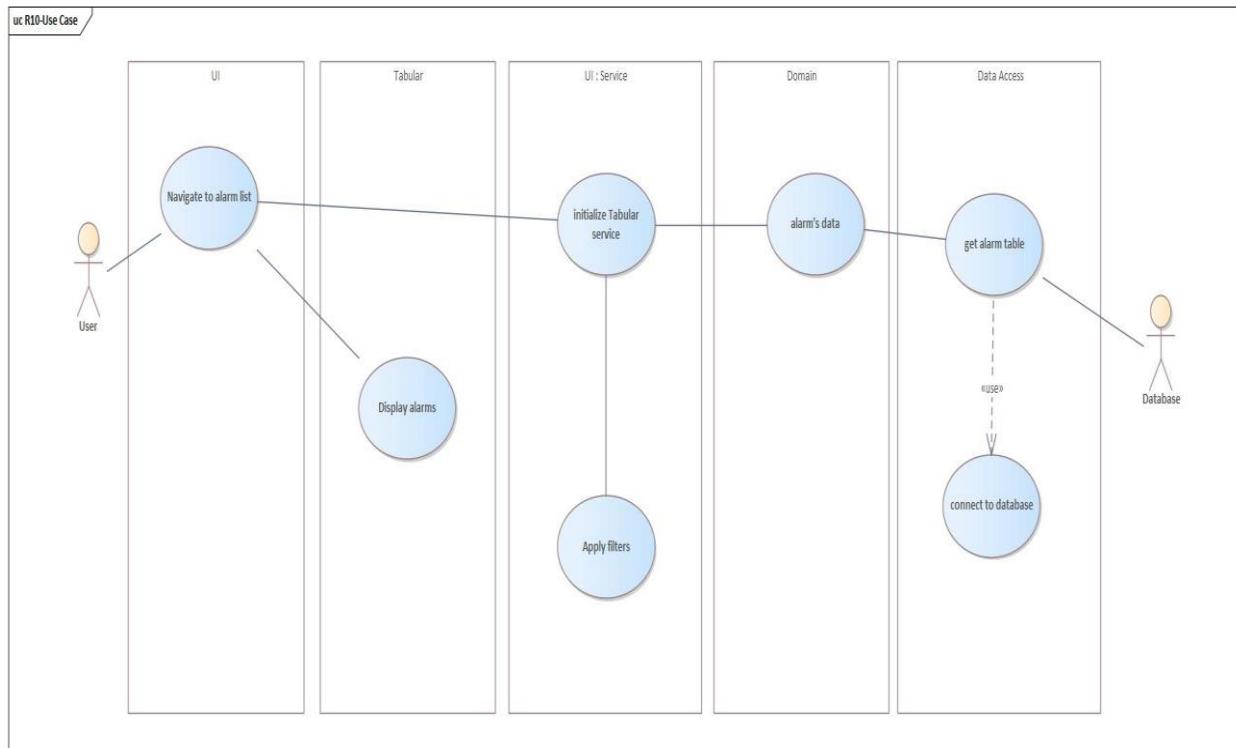
شکل ۳.۳۹ R09-Aria's Parameters History Class Diagram ۳.۳۹



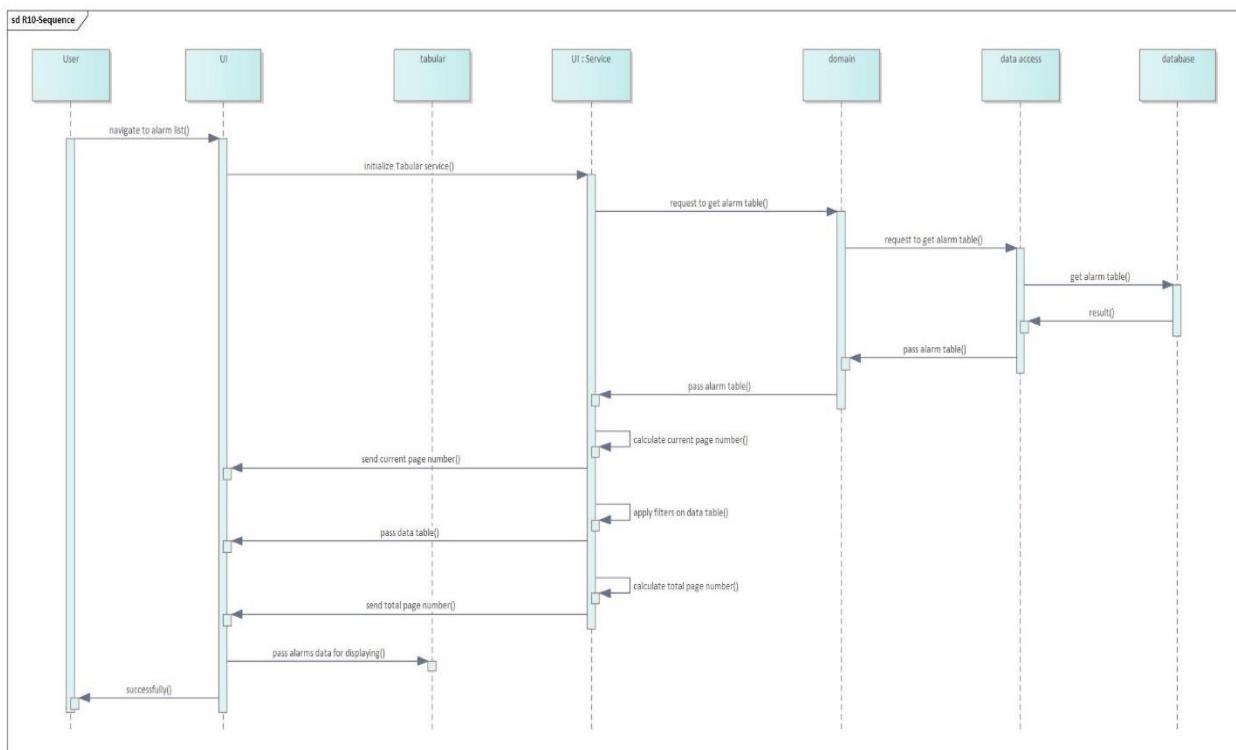
شکل ۴۰.۳ R09-Aria's Parameters History Activity Diagram ۴۰.۳

Aria's Alarms History ۱۰.۲.۳

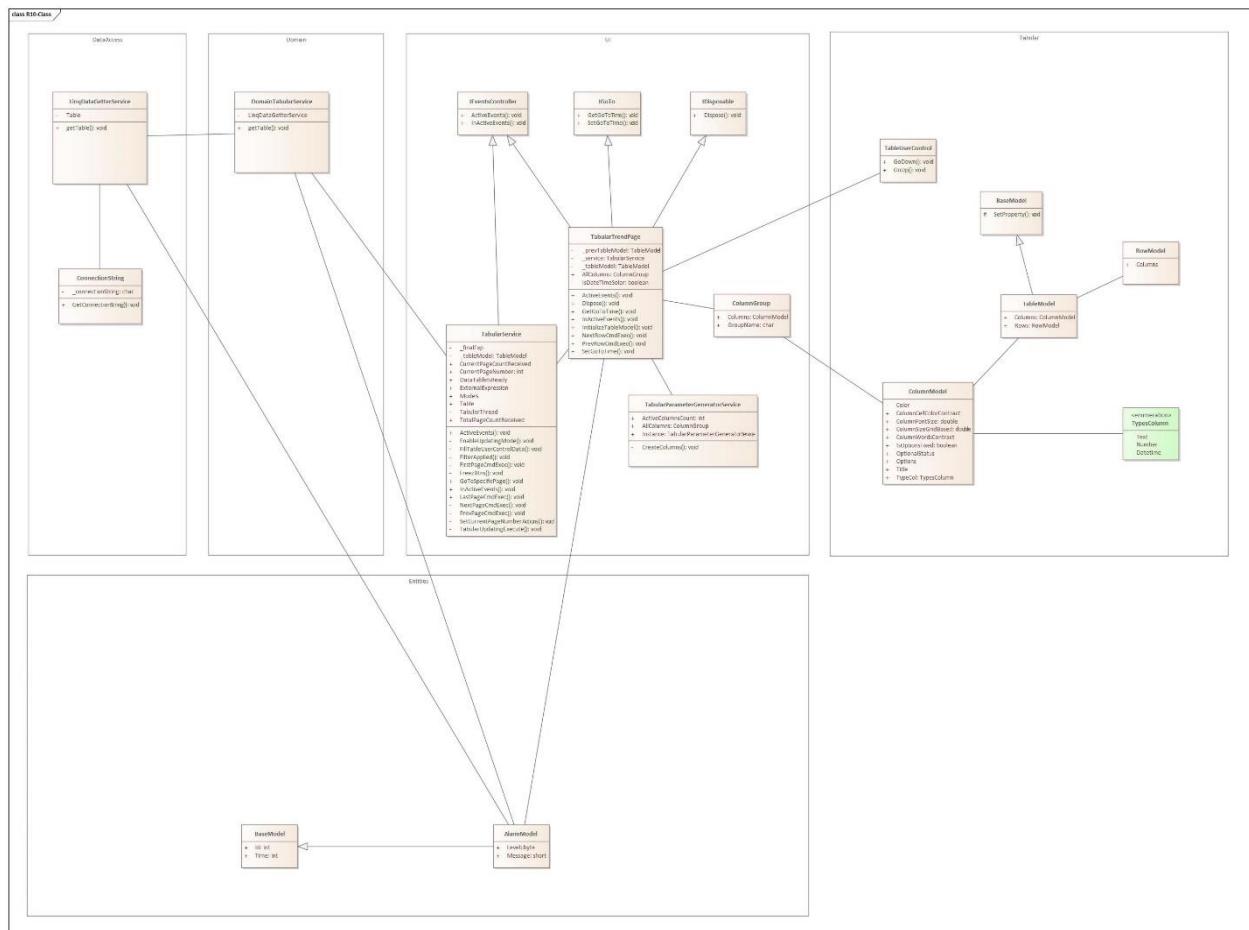
در این نیازمندی کاربر می‌خواهد که بتواند درون نرمافزار تمام اطلاعات مربوط به هشدارهای علائم حیاتی ذخیره شده آریا را نظارت کند. دیاگرام‌های مربوط به این نیازمندی را مشاهده می‌کنید:



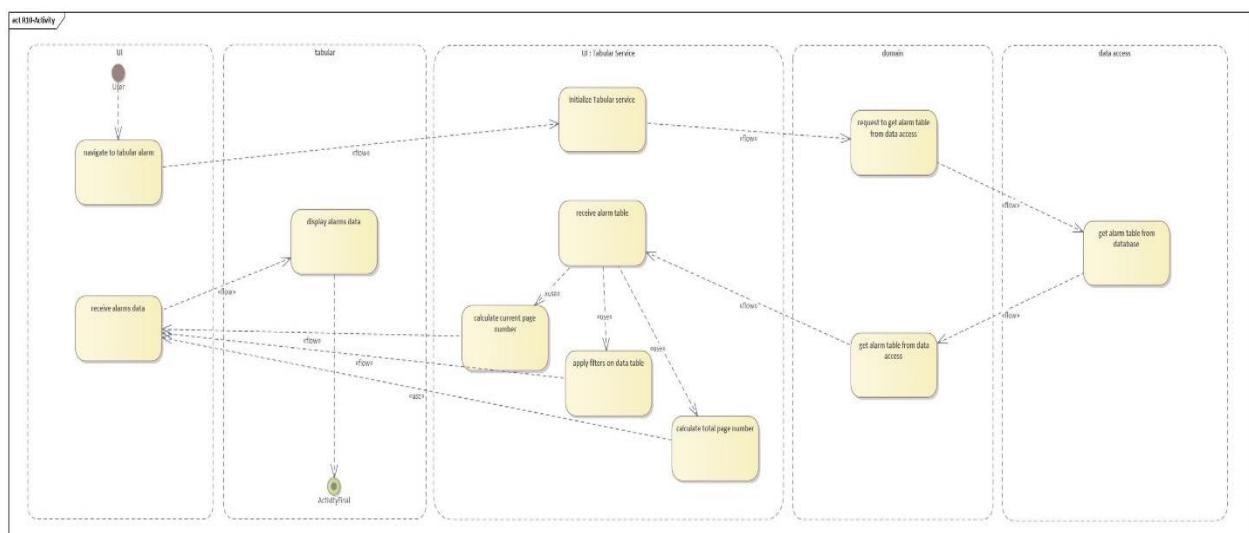
شکل ۴۱.۳ .R10-Aria's Alarms History Use Case Diagram



شکل ۴۲.۳ .R10-Aria's Alarms History Sequence Diagram



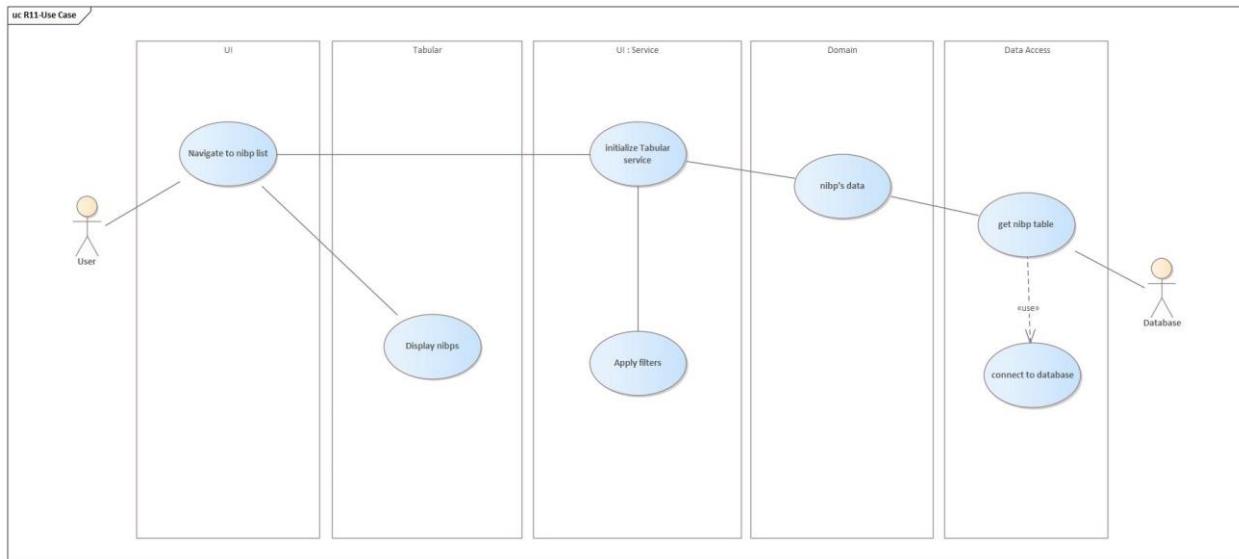
شکل ۴۳.۳ R10-Aria's Alarms History Class Diagram



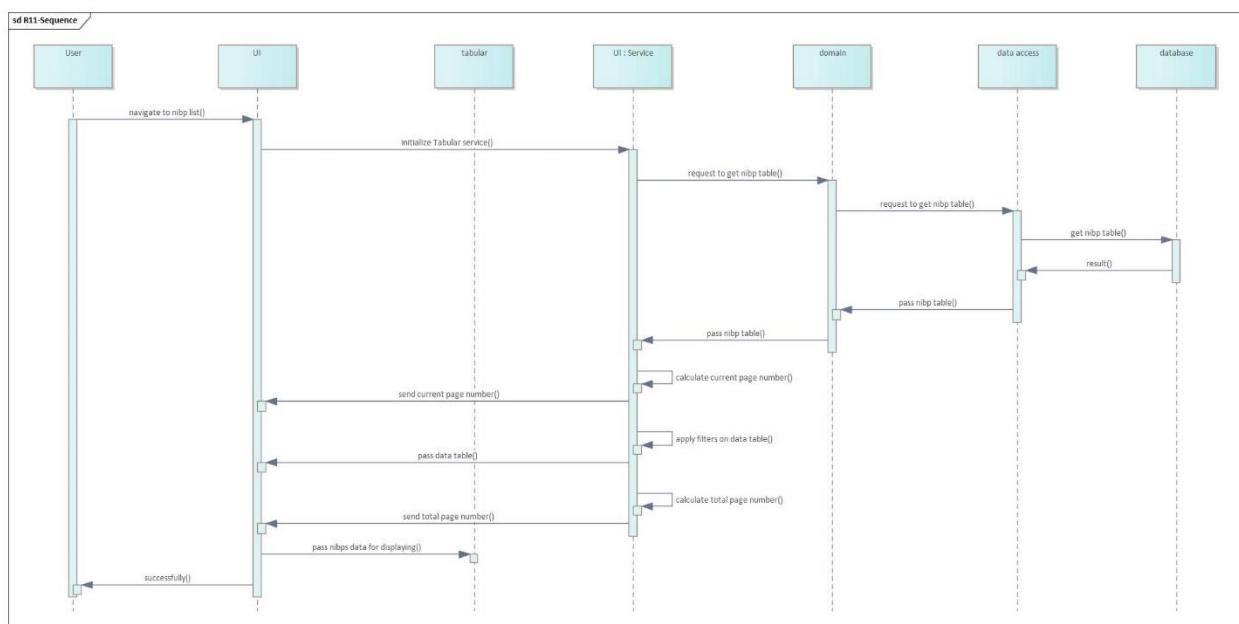
شکل ۴۴.۲ R10-Aria's Alarms History Activity Diagram

Aria's Nibps History ۱۱.۲.۳

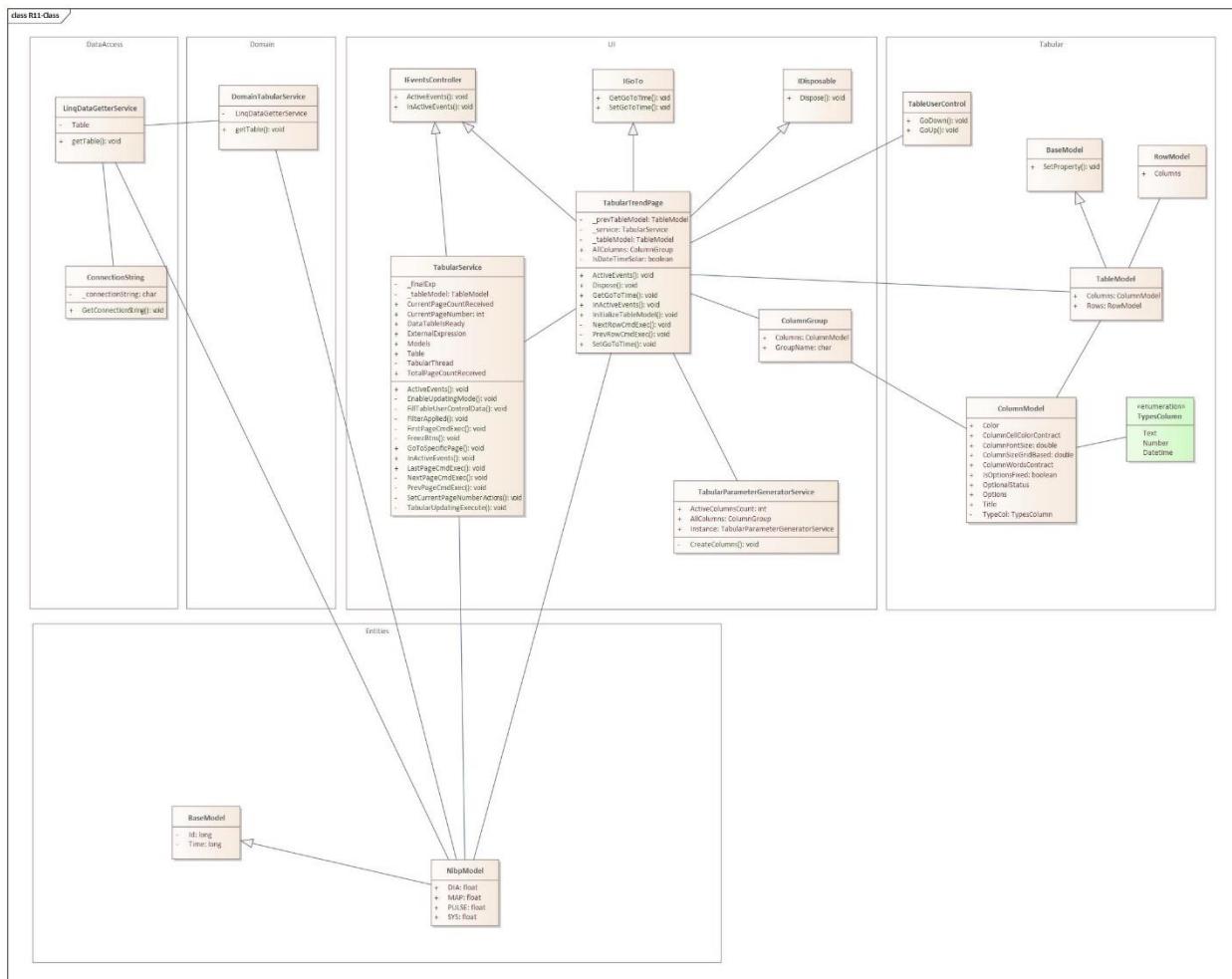
در این نیازمندی کاربر می‌خواهد بتواند درون نرم‌افزار تمام اطلاعات مربوط به Nibpi‌های ذخیره شده آریا را نظارت کند. دیاگرام‌های مربوط به این نیازمندی را مشاهده می‌کنید:



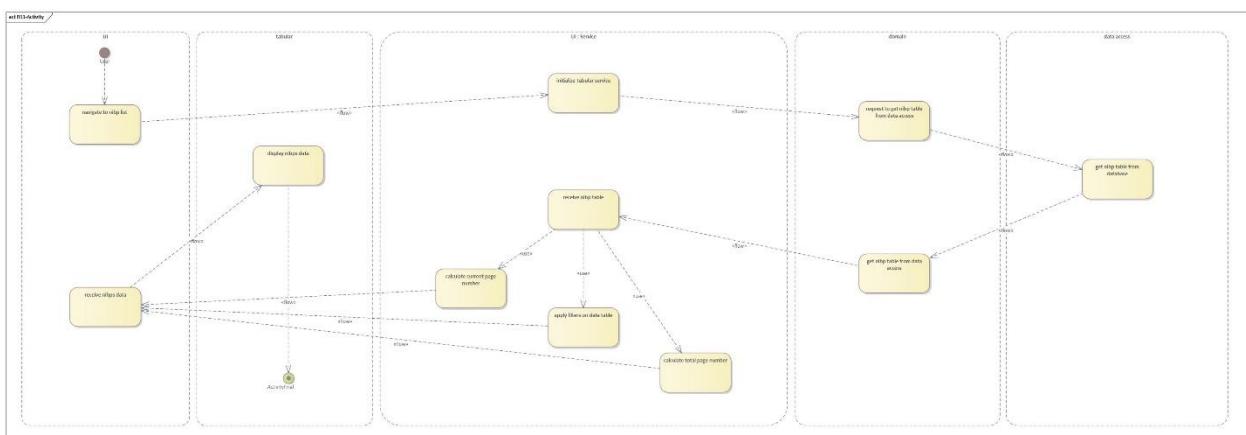
شکل ۴۵.۳ R11-Aria's Nibps History Use Case Diagram



شکل ۴۶.۳ R11-Aria's Nibps History Sequence Diagram



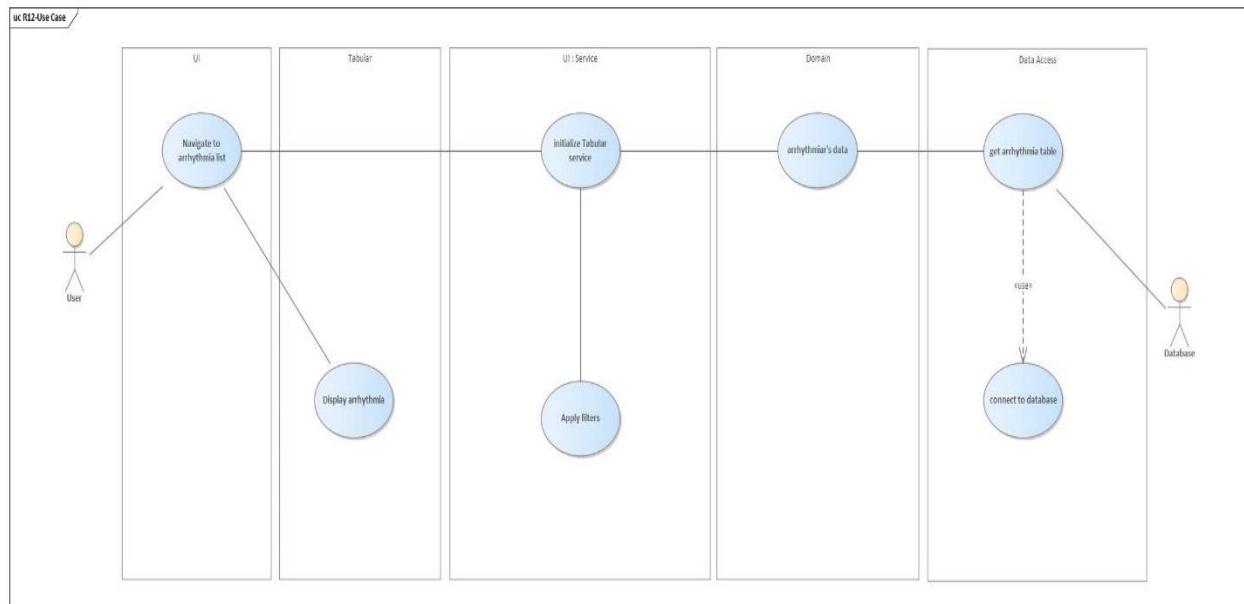
شکل ۴۷.۳ R11-Aria's Nibps History Class Diagram



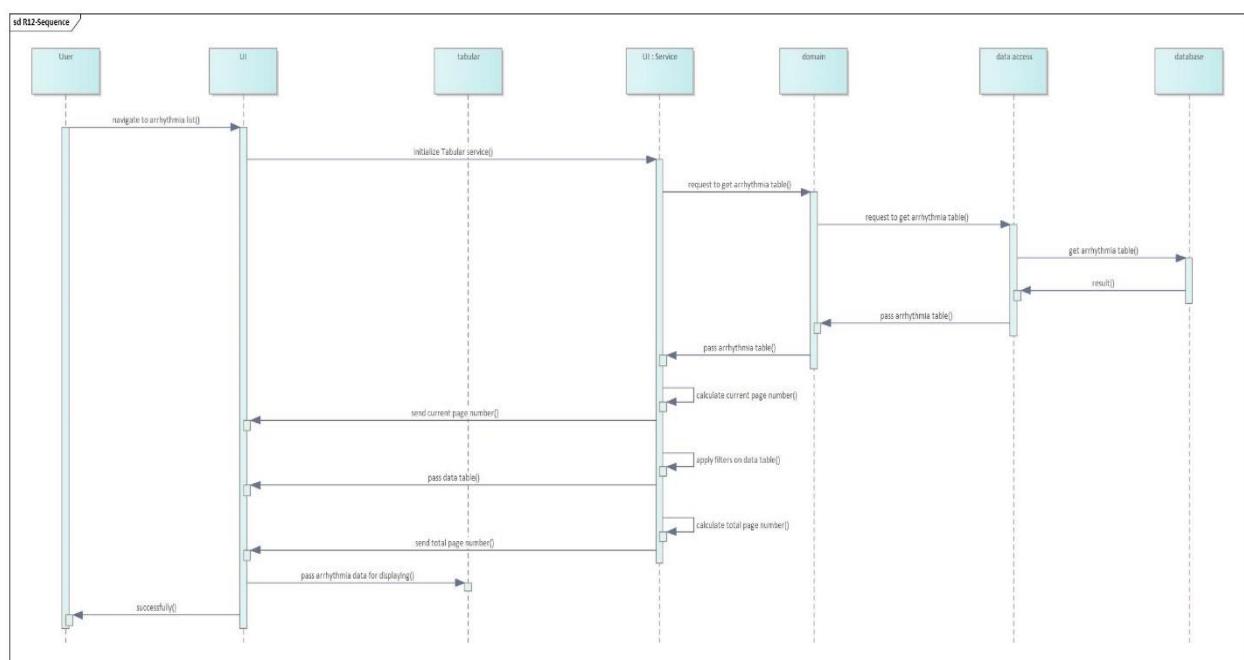
شکل ۴۸.۳ R11-Aria's Nibps History Activity Diagram

Aria's Arrhythmia History ۱۲.۲.۳

در این نیازمندی کاربر تمایل دارد که بتواند درون نرمافزار تمام اطلاعات مربوط به آریتمی‌های ذخیره شده آریا را نظارت کند. دیاگرام‌های مربوط به این نیازمندی را مشاهده می‌کنید:

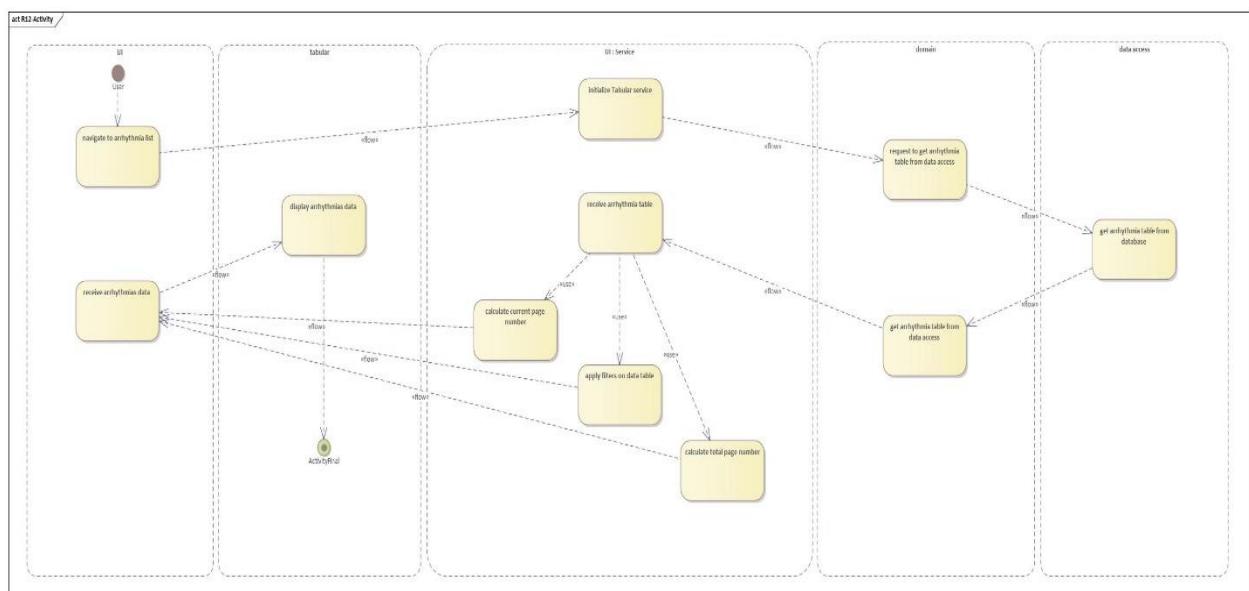


شکل ۴۹.۳ R12-Aria's Arrhythmia History Use Case Diagram



شکل ۵۰.۳ R12-Aria's Arrhythmia History Sequence Diagram

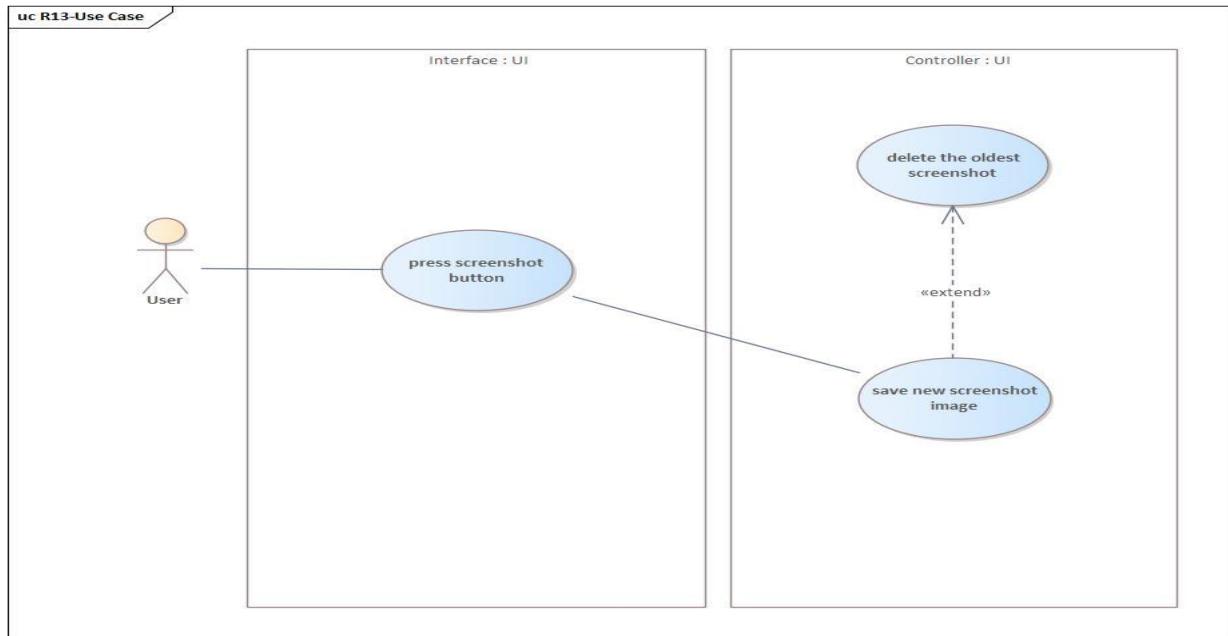
شكل ٥١.٣ R12-Aria's Arrhythmia History Class Diagram



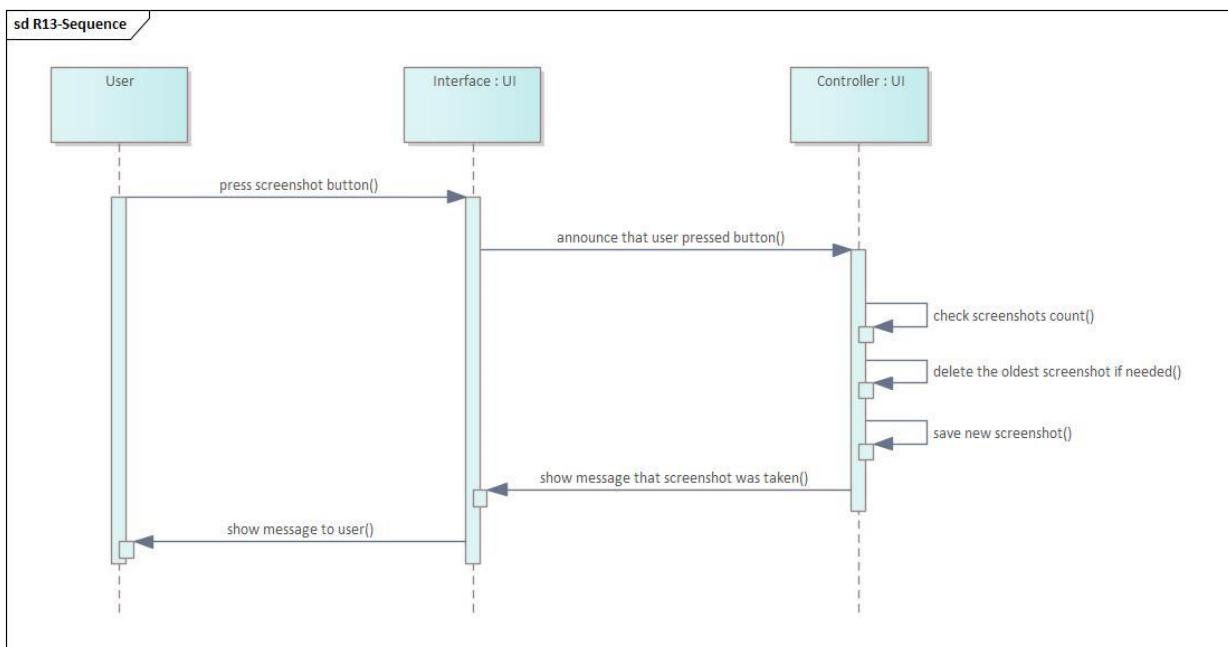
شكل ٥٢.٣ R12-Aria's Arrhythmia History Activity Diagram

Taking Screenshot ۱۳.۲.۳

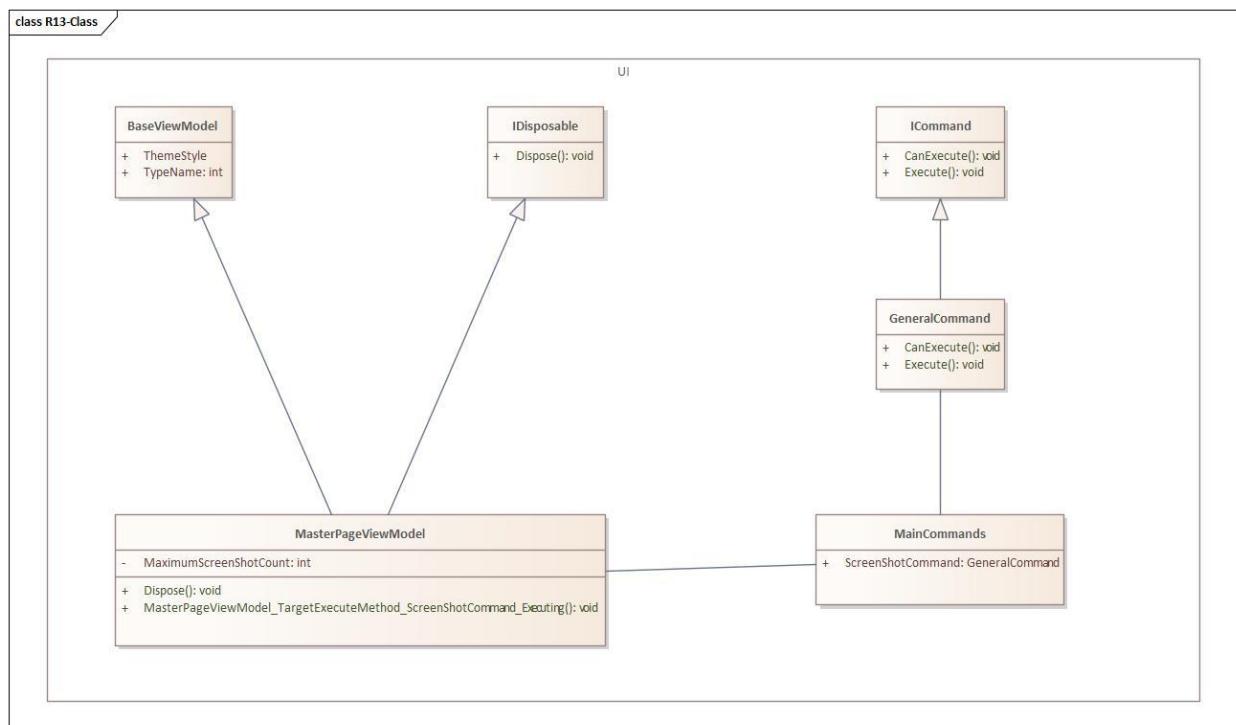
در این نیازمندی کاربر میخواهد که بتواند از طریق نرمافزار از صفحه نمایش عکس بگیرد و به USB منتقل کند. دیاگرام‌های مربوط به این نیازمندی را مشاهده می‌کنید:



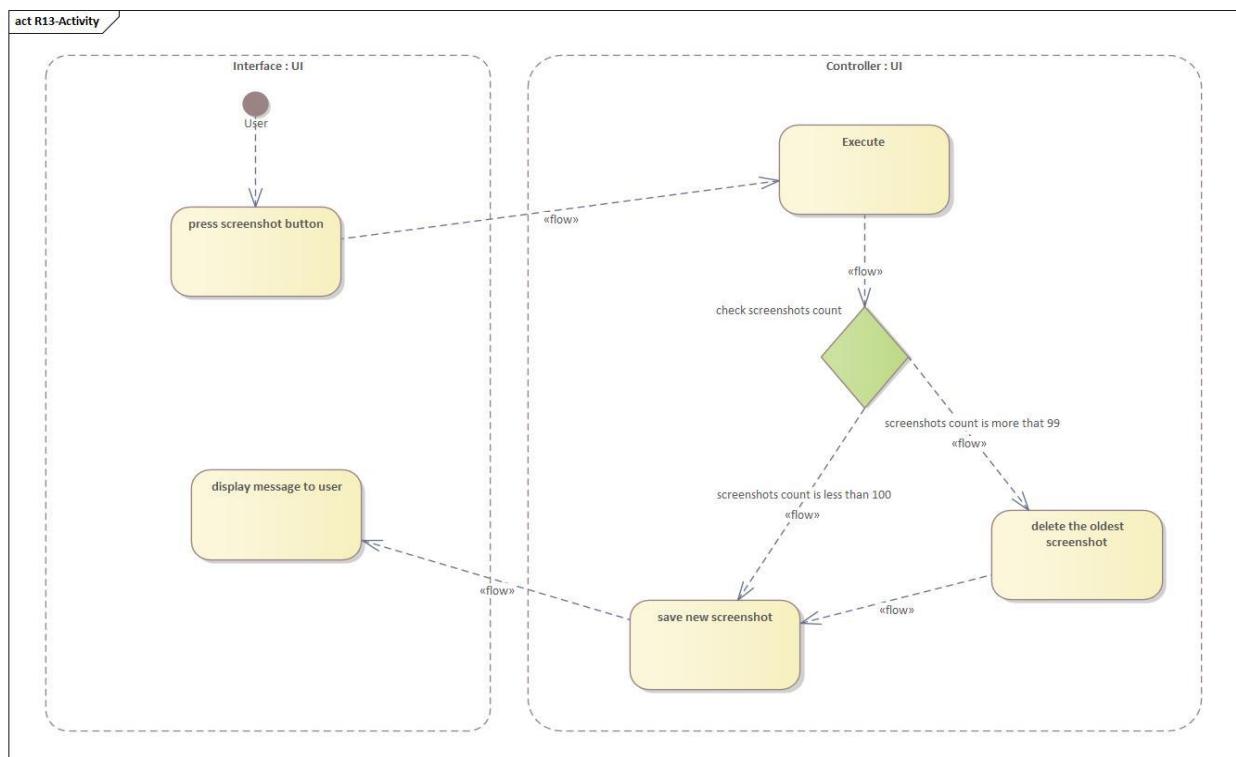
شکل ۱۳.۳ R13-Taking Screenshot Use Case Diagram



شکل ۱۴.۳ R13-Taking Screenshot Sequence Diagram



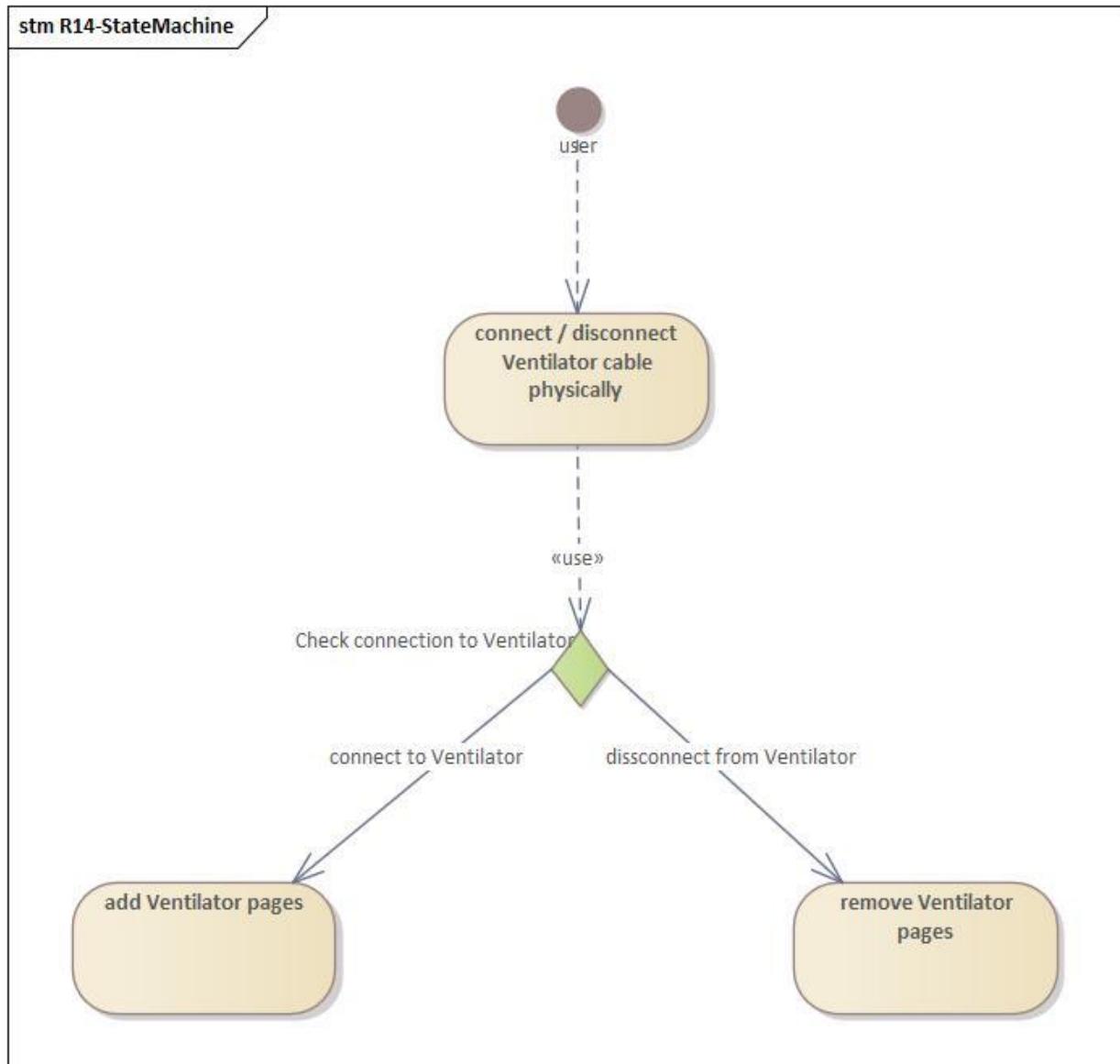
شکل ۵۵.۳ R13-Taking Screenshot Class Diagram



شکل ۵۶.۳ R13-Taking Screenshot Activity Diagram

Connection With Ventilator ۱۴.۲.۳

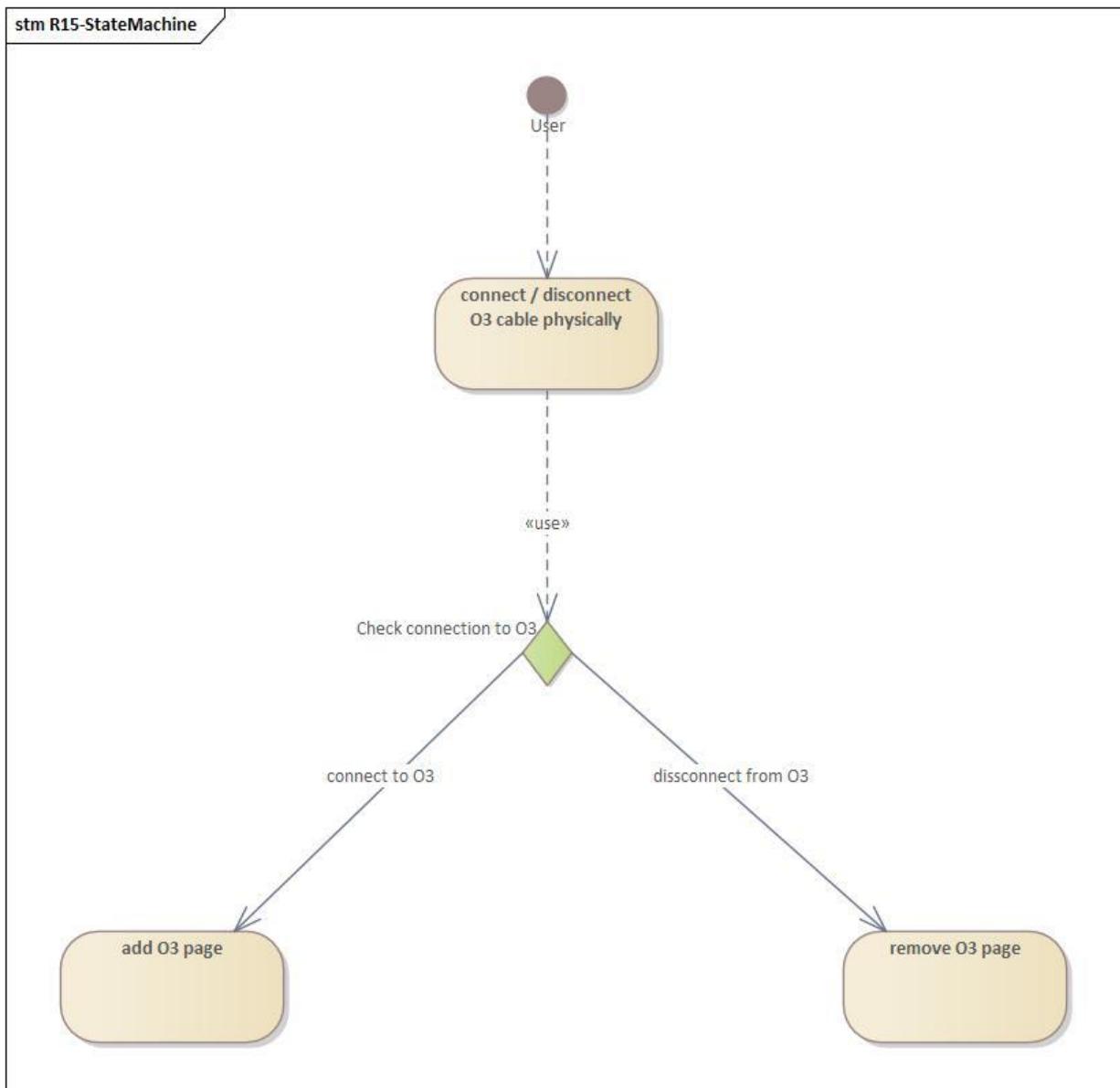
در این نیازمندی کاربر می‌خواهد که بتواند با دستگاه ونتیلاتور ارتباط برقرار کند و اطلاعات مربوط به پارامترها و سیگنال‌های آن را نظارت کند. دیاگرام مربوط به این نیازمندی را مشاهده می‌کنید:



شکل ۵۷.۳ .R14-Connection with Ventilator State Machine Diagram

Connection With O3 ۱۵.۲.۳

در این نیازمندی کاربر می‌خواهد که بتواند با ماژول O3 ارتباط برقرار کند و اطلاعات مربوط به پارامترها و سیگنال‌های آن را نظارت کند. دیاگرام مربوط به این نیازمندی را مشاهده می‌کنید:



R15-Connection with O3 State Machine Diagram ۵۸.۳

۳.۳ ابزارهای مورد استفاده

جهت پیاده‌سازی نرم‌افزار دستگاه پزشکی Viewer از زبان برنامه‌نویسی سی‌شارپ^۱ و تکنولوژی WPF استفاده می‌کنیم. دلیل اینکه از زبان برنامه‌نویسی سی‌شارپ استفاده می‌کنیم این است که می‌خواهیم

^۱ C Sharp(C#)

نرمافزار را تحت ویندوز اجرا کنیم و از آن جایی که هم ویندوز و هم زبان برنامه‌نویسی سی‌شارپ از محصولات شرکت بزرگ مایکروسافت^۲ هستند؛ سازگاری خوبی باهم دارند. در نتیجه کنترل منابع یعنی میزان استفاده از واحد کنترل مرکزی و حافظه بهتر، قابل اندازه‌گیری است. دلیل اینکه به جای استفاده از قابلیت‌های معروف سی‌شارپ مثل^۳ Windows Form از تکنولوژی WPF استفاده می‌کنیم این است که می‌خواهیم طراحی کاربرپسندتری داشته باشیم و طبق دفترچه اصلی Wpf که در وبسایت رسمی شرکت مایکروساف به آدرس اینترنتی^۴ learn.microsoft.com قرار دارد، تکنولوژی WPF با امکانات زیادی که جهت طراحی رابط کاربری در اختیار توسعه‌دهندگان قرار می‌دهد این امکان را می‌دهد که رابط کاربری کاربرپسندتری طراحی کنیم.

محیط توسعه کدهای برنامه هم از طریق نرم افزار^۵ Visual Studio است که محصول شرکت مایکروسافت است و بین برنامه‌نویسان، یکی از محبوب‌ترین محیط‌های توسعه برای زبان سی‌شارپ است. جهت ذخیره سازی داده‌های مربوط به علائم حیاتی بیمار از پایگاه داده SQL Server استفاده می‌کنیم که یکی از محبوب‌ترین و قدرتمندترین نرم‌افزارهای ساخت و مدیریت پایگاهداده محسوب می‌شود. SQL Server یک بانک اطلاعاتی از نوع پایگاهداده‌های رابطه‌ای است که توسط شرکت مایکروسافت ارائه شده است و وظیفه اصلی و حیاتی آن، ذخیره و بازیابی اطلاعات بر اساس درخواست نرم‌افزارهای دیگر می‌باشد. مهم‌ترین کاربرد SQL Server ساخت بانک اطلاعاتی با حجم بالا است که تعداد زیادی کاربر به صورت همزمان می‌توانند به آن دسترسی داشته باشند.

^۱ Windows Presentation Foundation

^۲ مایکروسافت یکی از بزرگ‌ترین شرکت‌ها در حوزه فناوری است.

^۳ Windows Forms فریمورکی به منظور ایجاد برنامه‌های گرافیکی برای ویندوز است.

^۴ آدرس اینترنتی دقیق دفترچه WPF در وبسایت مایکروسافت به نشانی زیر است:

<https://learn.microsoft.com/en-us/dotnet/desktop/wpf/?view=netdesktop-6.0>

^۵ ویژوال استودیو

۴.۳ معماری سیستم

در فصول گذشته معماری سخت‌افزاری سیستم را بررسی کردیم و ذکر کردیم که در این پژوهش به دنبال این هستیم که بر بستر کامپیوتر گیادا نرم‌افزار را اجرا کنیم که شامل ویژگی‌های سخت‌افزاری زیر است:

Processor: Intel(R) Celeron() CPU N3160 @ 1.60GHz 1.60 GHz

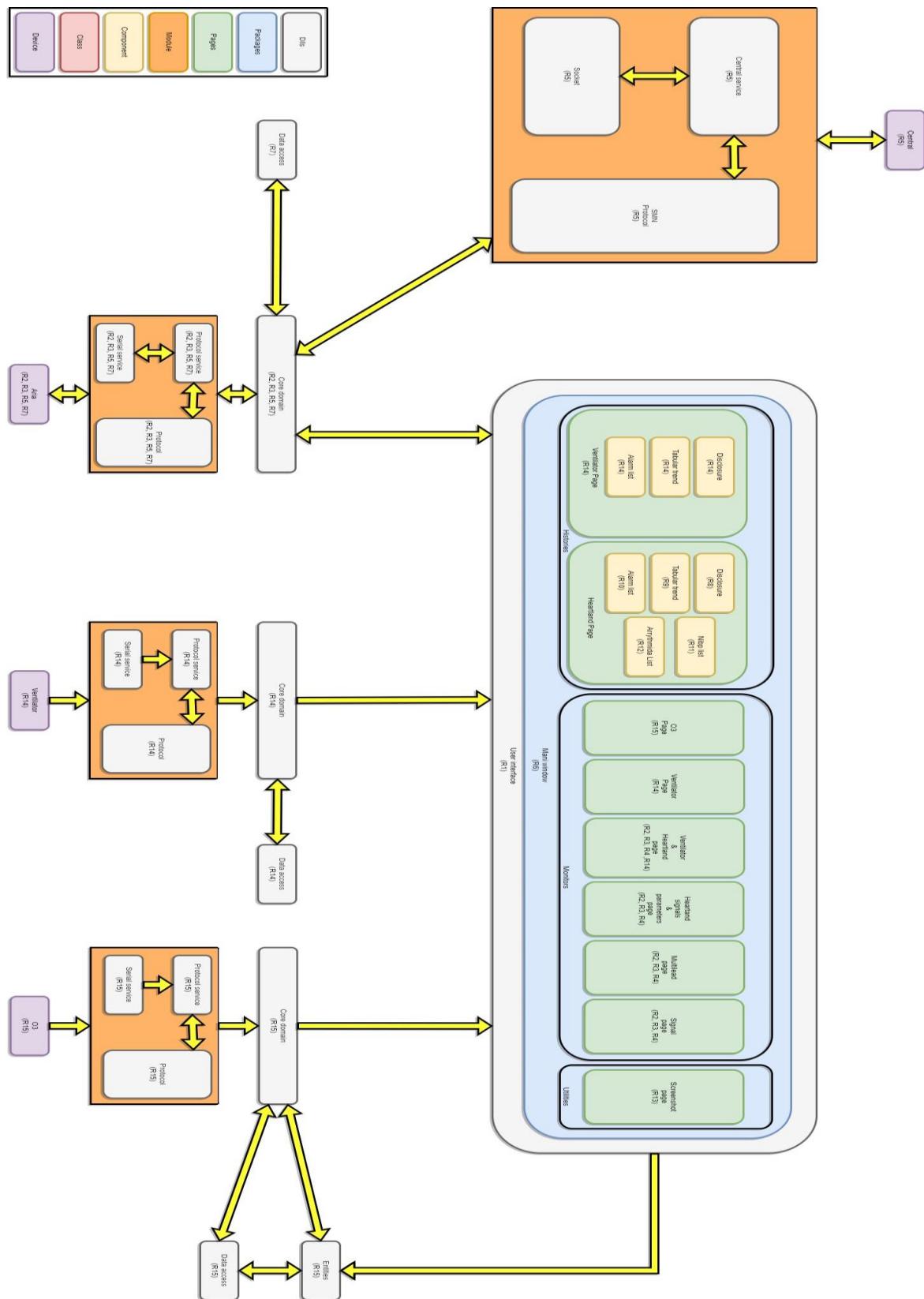
Installed memory (RAM): 2.00 GB (1.89 GB usable)

در سمت نرم‌افزار دستگاه Viewer، هسته این پروژه که آن را به نام قسمت بکاند می‌شناسیم وجود دارد که وظایف زیر را دارد:

- خواندن و نوشتمن داده روی درگاه باز شده جهت تعامل با دستگاه‌های جانبی.
- تبدیل داده‌های خام به اطلاعات طبق پروتکل‌های ارتباطی و بالعکس.
- توزیع اطلاعات، بین لایه‌های مختلف نرم‌افزار.
- ذخیره‌سازی در پایگاهداده.
- ارسال به سانترال.
- ارسال به لایه UI (رابط کاربری)^۱ جهت تعامل با کاربر.

تعامل با نرم‌افزار بدین صورت است که فرمان کاربر از طریق رابط کاربری به سمت لایه UI نرم‌افزار منتقل می‌شود؛ اگر فرمان از نوع ظاهری، مثل زوم کردن سیگنال باشد، در همین لایه UI اجرا می‌شود؛ اما اگر ظاهری نباشد به لایه بکاند ارسال می‌شود و قسمت بکاند نیز آن فرمان را یا به دستگاه‌های جانبی ارسال می‌کند یا درون نرم‌افزاری به اجرا درمی‌آورد. بعد از به نتیجه رسیدن فرمان، کاربر نتیجه تعامل خود با سیستم را از طریق رابط کاربری مشاهده می‌کند. شمای کلی معماری نرم‌افزاری به شکل زیر (۵۹) است.

^۱ User Interface



شکل ۵.۳ معماری نرم افزاری.

این معماری نرمافزاری، همان روش ما جهت به اجرا درآوردن نرمافزار بر بستر کامپیوتر گیادا است که نیازهای کاربر و نرمافزار را برطرف می‌کند. در تصویر مشخص شده است که عملیات مربوط به هر نیازمندی در چه جایگاهی از این معماری اجرا می‌شود. میخواهیم در یک سناریو نحوه کار Viewer با آریا، سپس با سانترال، سپس با ونتیلاتور و سپس با O3 را بررسی کنیم. طبق شکل (۳-۵۹) نحوه کار به شرح زیر است:

۱. دستگاه آریا به صورت فیزیکی از طریق کابل به سیستم Viewer متصل می‌شود.
۲. لایه Serial Service که وظیفه خواندن و نوشتن داده روی درگاه را دارد، داده‌های آریا را از روی درگاه باز شده می‌خواند.
۳. لایه Protocol Service داده‌ها را از لایه Serial Service دریافت می‌کند و با استفاده از لایه‌ی Protocol، داده‌های خام آریا را به اطلاعات معنادار تبدیل می‌کند.
۴. لایه‌ی Domain که هسته مرکزی این معماری است اطلاعات آریا را از لایه Protocol دریافت می‌کند و بعد از تغییرات لازمه، اطلاعات را بین لایه‌های دیگر از قبیل لایه Central UI و لایه Data access توزیع می‌کند.
۵. لایه UI اطلاعات آریا را از لایه‌ی Domain دریافت می‌کند و از طریق معماری MVVM به کاربر نمایش می‌دهد.
۶. لایه Data access اطلاعات آریا را از لایه‌ی Domain دریافت می‌کند و با روش^۱ در Ado.net پایگاهداده ذخیره می‌کند.
۷. لایه Central Service اطلاعات آریا را از لایه‌ی Domain دریافت می‌کند و پس از اینکه با استفاده از لایه Socket ارتباطات با سانترال را برقرار شد، با استفاده از لایه SMN Protocol آن اطلاعات را به سانترال ارسال می‌کند.
۸. دستگاه ونتیلاتور به صورت فیزیکی از طریق کابل به سیستم Viewer متصل می‌شود.

^۱ ADO.NET در حقیقت یک رابط برنامه نویسی است که مجموعه امکانات لازم به منظور برقراری اتصال با بانک های اطلاعاتی را در اختیار برنامه نویسان قرار می‌دهد.

۹. لایه Serial Service که وظیفه خواندن و نوشتن داده روی درگاه را دارد داده‌های ونتیلاتور را از روی درگاه باز شده می‌خواند.

۱۰. لایه Protocol Service داده‌ها را از لایه Serial Service دریافت می‌کند و با استفاده از لایه‌ی پروتکل، داده‌های خام ونتیلاتور را به اطلاعات معنادار تبدیل می‌کند.

۱۱. لایه‌ی Domain که هسته مرکزی این معماری است اطلاعات ونتیلاتور را از لایه Protocol دریافت می‌کند و بعد از تغییرات لازمه، اطلاعات را بین لایه‌های دیگر از قبیل لایه Service و لایه UI توزیع می‌کند.

۱۲. لایه UI اطلاعات ونتیلاتور را از لایه‌ی Domain دریافت می‌کند و از طریق معماری MVVM به کاربر نمایش می‌دهد.

۱۳. لایه Data access اطلاعات ونتیلاتور را از لایه‌ی Domain دریافت می‌کند و از طریق Ado.net در دیتابیس ذخیره می‌کند.

۱۴. مژول O3 به صورت فیزیکی از طریق کابل به سیستم Viewer متصل می‌شود.

۱۵. لایه Serial Service که وظیفه خواندن و نوشتن داده روی درگاه را دارد داده‌های O3 را از روی درگاه باز شده می‌خواند.

۱۶. لایه Protocol Service داده‌ها را از لایه Serial Service دریافت می‌کند و با استفاده از لایه‌ی پروتکل، داده‌های خام O3 را به اطلاعات معنادار تبدیل می‌کند.

۱۷. لایه‌ی Domain که هسته مرکزی این معماری است اطلاعات O3 را از لایه Protocol دریافت می‌کند و بعد از تغییرات لازمه، اطلاعات را بین لایه‌های دیگر از قبیل لایه Service و لایه UI توزیع می‌کند.

۱۸. لایه UI اطلاعات O3 را از لایه‌ی Domain دریافت می‌کند و از طریق معماری MVVM به کاربر نمایش می‌دهد.

۱۹. لایه Data access اطلاعات O3 را از لایه‌ی Domain دریافت می‌کند و از طریق Ado.net در دیتابیس ذخیره می‌کند.

۲۰. جهت ارسال داده به دستگاه جانبی آریا این مراحل پشت سر هم اجرا می‌شوند؛ در ابتدا کاربر عملیات مورد نظر را انجام می‌دهد، سپس فرمان از سمت لایه UI صادر می‌شود و به لایه Protocol مرکزی یعنی لایه Domain منتقل می‌شود، لایه Domain تغییرات را به لایه Protocol Service منتقل می‌کند و لایه Protocol Service با استفاده از لایه Protocol داده‌هایی که Serial Service می‌باشد به آریا ارسال کند را آماده می‌کند و این داده‌ها را از طریق لایه Serial Service روی درگاه باز شده مربوطه ارسال می‌کند.

۲۱. جهت ارسال داده به ماژول O3 مراحلی که در مورد قبلی ذکر شدند عیناً تکرار می‌شوند.

۲۲. زمانی که کاربر به صفحه تاریخچه مربوط به آریا مراجعه می‌کند، صفحه نمایش به دو قسمت تقسیم می‌شود؛ در یک قسمت عملیات مانیتورینگ همزمان انجام می‌شود یعنی مراحل ۱ تا ۷، در قسمت دیگر که مربوط به تاریخچه است کاربر می‌تواند تاریخچه‌ی سیگنال‌ها، پارامترها، هشدارها، NIBP‌ها و آریتمی‌ها را مشاهده کند. بعد از این که کاربر انتخاب کند از چه زمانی تا چه زمانی را می‌خواهد بررسی کند، درخواستش از لایه UI به لایه مرکزی یعنی لایه Domain منتقل می‌شود، سپس لایه Domain درخواست را به لایه Data access منتقل می‌کند و بعد از این که لایه Data access اطلاعات درخواست شده را از پایگاهداده خواند و آماده انتقال کرد؛ اطلاعات را به لایه Domain منتقل می‌کند و لایه Domain اطلاعات را جهت پاسخگویی و نمایش به لایه UI ارسال می‌کند.

۲۳. زمانی که کاربر به صفحه تاریخچه مربوط به ونتیلاتور مراجعه می‌کند مراحلی که در مورد قبلی ذکر شدند عیناً تکرار می‌شوند.

۲۴. زمانی که کاربر می‌خواهد تاریخچه ماژول O3 را نظارت کند به صفحه مربوطه در برنامه مراجعه می‌کند و مراحل مانند دو مورد قبلی عیناً تکرار می‌شوند.

۵.۳ معماری لایه UI

در تکنولوژی WPF از دو زبان برنامه‌نویسی استفاده می‌شود؛ یکی زبان نشانه‌گذاری XAML که جهت توسعه رابط کاربری استفاده می‌شود و دیگری زبان سی‌شارپ که جهت توسعه بکاند است یعنی روی رابط کاربری کار می‌کند.

در طول زمان که برنامه رشد می‌کند و دستخوش تغییرات می‌شود، پیچیدگی برنامه زیاد می‌شود. این مسائل باعث می‌شود که پیچیدگی پروژه افزایش پیدا کند و قابلیت توسعه و نگهداری کاهش یابد. از این رو، به جهت این که هزینه ایجاد تغییرات رابط کاربری افزایش پیدا نکند و برنامه قابل تست و اطمینان باشد ضروری است که از یک معماری (الگو) جهت توسعه رابط کاربری استفاده کنیم.

الگوی انتخاب شده الگوی MVVM است، زیرا این الگو کمک می‌کند که به طور واضح بخش منطق کد از بخش رابط کاربری کد جدا باشد و این مرزبندی باعث این می‌شود که کد برنامه قابل توسعه باقی بماند. از دیگر مزایای این نوع مرزبندی کاهش پیچیدگی، افزایش قابلیت نگهداری، افزایش قابلیت تست و اطمینان است. نحوه سازماندهی این الگو به شرح زیر است:

- Model: این بخش فقط وظیفه نگهداری داده‌ها را دارد.
- ViewModel: این بخش به عنوان ارتباط بین بخش‌های View و Model عمل می‌کند و شامل منطق رابط کاربری است.
- View: این بخش وظیفه نمایش رابط کاربری به کاربر را دارد و نحوه ارتباطش با بخش ViewModel از طریق روش اتصال داده‌ها^۱ است.

۶.۳ پیاده‌سازی و ارزیابی

هدف از این پژوهش طراحی نرمافزار دستگاه پزشکی Viewer جهت مانیتورینگ و ذخیره‌سازی علائم حیاتی بیمار است؛ در فصل بعد، این روش را با استفاده از ابزارهای استفاده شده که در این فصل معرفی

^۱ Data Binding

بر آن‌ها داشتیم (مثل ویژوال استودیو، زبان برنامه‌نویسی سی‌شارپ و ...) پیاده‌سازی خواهیم کرد و نرم‌افزار را مورد ارزیابی قرار خواهیم داد. [۲۳]

۴

فصل چهارم

تجزیه و تحلیل داده‌ها

تجزیه و تحلیل داده‌ها

سیستم Viewer قرار است که در درمانگاه‌ها مورد استفاده قرار بگیرد و پزشکان بنا به علامت حیاتی که از بیمار در سیستم نظارت می‌کنند، تصمیم‌گیری‌های اساسی انجام می‌دهند که موجب به احیای بیمار می‌شود. بنا به حساسیت این اطلاعات می‌باشد از صحت عملکرد سیستم اطمینان کسب کنیم.

در فصل قبل روش پیشنهادی این پژوهش را بیان کردیم و در این فصل قصد داریم به طور عملیاتی روشی که برای مدیریت (دربیافت، ارسال، خواندن، ذخیره سازی و نمایش) داده‌ها ارائه دادیم را پیاده‌سازی کنیم و سپس، روش پیشنهادی را مورد ارزیابی قرار دهیم.

۱.۴ پیاده‌سازی

در این بخش وارد مرحله پیاده‌سازی روش پیشنهاد شده‌ی این پژوهش می‌شویم و با استفاده از تکنولوژی‌های ذکر شده، به پیاده‌سازی آن می‌پردازیم. نکته قابل توجهی که بهتر است بدانید، این است که دستگاه Viewer در درجه اول جهت مانیتورینگ دستگاه جانبی آریا و ارسال داده‌های آن به سانترال توسعه یافته است و با گذر زمان، نیازهای بیزینسی^۱ تغییر پیدا کرده و ارتباط با دستگاه جانبی ونتیلاتور و مژول O3 هم به دستگاه Viewer اضافه شده است. برای اینکه کدهای برنامه بیش از حد، زیاد نشود و شلوغی آن‌ها باعث از دست رفتن قابلیت‌های توسعه و نگهداری نشود، به جای توسعه یک پروژه به توسعه سه پروژه می‌پردازیم که به شرح زیر هستند:

۱. یک پروژه با نام Ventilator جهت ارتباط با دستگاه ونتیلاتور که شامل تمامی لایه‌های معماری (لایه Serial تا لایه UI) است و به صورت مستقل قابل اجرا است.

۲. یک پروژه دیگر با نام O3 جهت ارتباط با مژول O3 که شامل تمامی لایه‌های معماری (لایه Serial تا لایه UI) است و به صورت مستقل قابل اجرا است.

۳. پروژه دیگر با نام HeartlandAriaTransfer جهت ارتباط با دستگاه آریا و ارسال داده‌های آن به سانترال که شامل تمامی لایه‌های معماری (لایه Serial تا لایه UI) است و به صورت مستقل

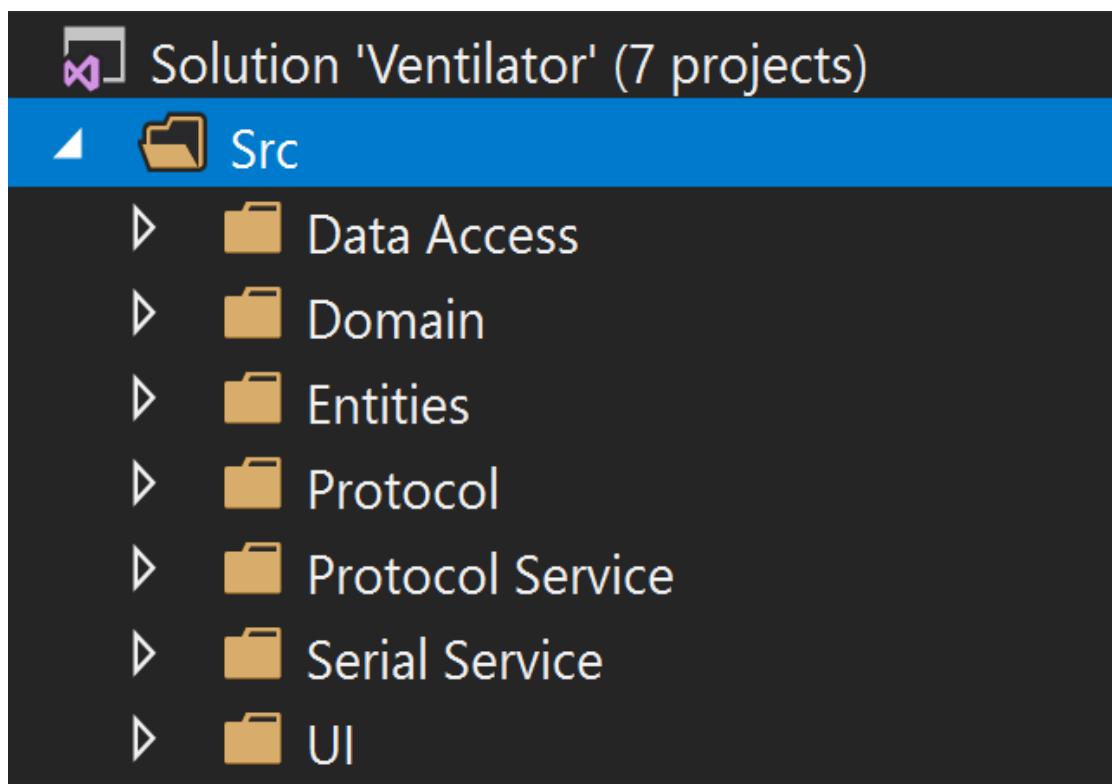
¹ business

قابل اجرا است. در این پروژه از دو پروژه‌ی قبلی استفاده می‌شود تا کاربر بتواند با تمامی دستگاه‌ها کار بکند.

از آن جایی که پروژه‌ها از لحاظ ساختاری مشابه به هم هستند، پروژه‌های Ventilator و O3 را به صورت مختصر و پروژه HeartlandAriaTransfer که شامل دو پروژه دیگر است را به صورت مفصل‌تر توضیح می‌دهیم.

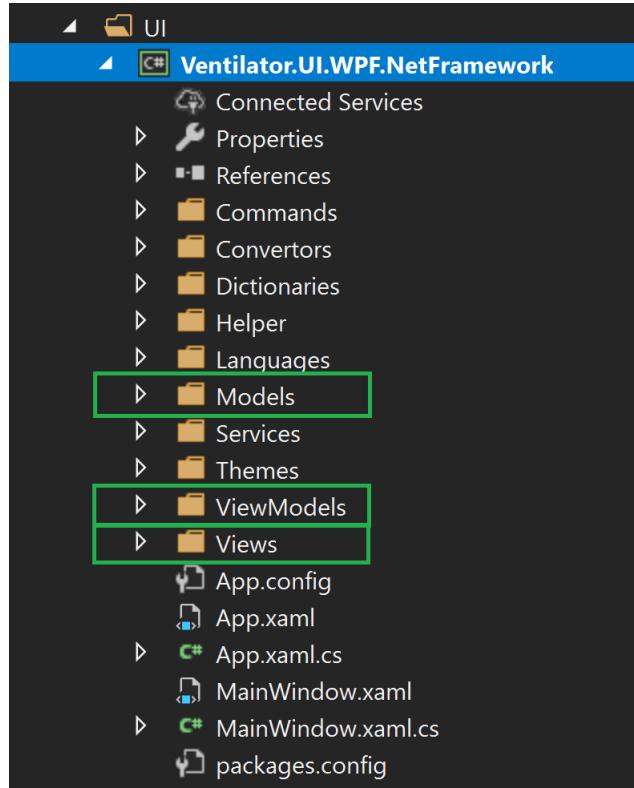
۱.۱.۴ پیاده‌سازی پروژه Ventilator

سناریو پیاده‌سازی پروژه Ventilator بدین صورت است که اول لایه Serial Service توسعه داده می‌شود تا در ابتدای کار بستر انتقال داده بین نرم‌افزار دستگاه Viewer و دستگاه ونتیلاتور برقرار شود. سپس لایه‌های Protocol Service و Protocol طبق پروتکل دستگاه ونتیلاتور توسعه داده می‌شود که وظیفه تفسیر داده‌ها به اطلاعات مورد استفاده در برنامه را دارد. سپس لایه Domain توسعه داده می‌شود که وظیفه توزیع اطلاعات بین لایه‌های UI و Data Access پروژه را دارد (شکل ۲.۴).



شکل ۱.۴ معماری نرم‌افزاری پروژه Ventilator

اطلاعات از UI به لایه Domain منتقل می‌شوند که این لایه وظیفه نمایش اطلاعات به کاربر را دارد. که طبق الگو MVVM توسعه داده شده است (شکل ۲.۴). علاوه بر سه پوشه‌ی Models، Views و ViewModels پوشه‌های دیگری نیز وجود دارند که همگی آنها جهت پیشبرد اهداف همان سه پوشه توسعه داده شده‌اند.



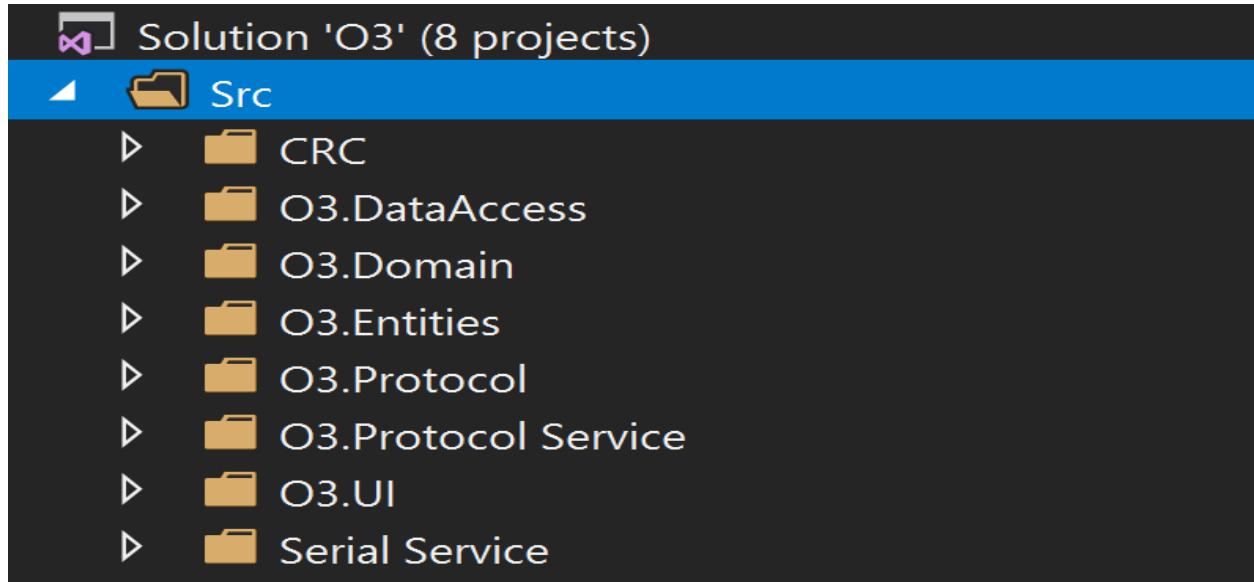
شکل ۲.۴ معماری نرم‌افزاری لایه UI پروژه Ventilator

اطلاعات از لایه Data Access به لایه Domain منتقل می‌شوند که این لایه وظیفه ذخیره و بازیابی اطلاعات از پایگاه داده SQL Server را دارد (اسکریپت مربوط به پایگاه داده Ventilator در بخش پیوست موجود است). علاوه بر لایه‌های ذکر شده، یک لایه دیگری به نام Entities وجود دارد که فقط شامل مدل‌های پایگاهداده‌ای است و استفاده آن در لایه‌های Data Access و UI است.

۲.۱.۴ پیاده‌سازی پروژه O3

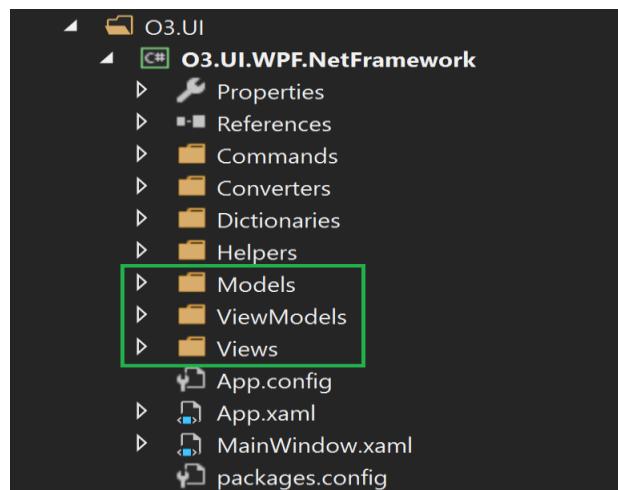
سناریو پیاده‌سازی پروژه O3 بدین صورت است که اول لایه Serial Service توسعه داده می‌شود تا در ابتدای کار بستر انتقال داده بین نرم‌افزار دستگاه Viewer و ماژول O3 برقرار شود. سپس لایه‌های

طبق پروتکل ماژول O3 توسعه داده می‌شود که وظیفه تفسیر داده‌ها به اطلاعات مورد استفاده در برنامه را دارد. سپس لایه Domain توسعه داده می‌شود که وظیفه توزیع اطلاعات بین لایه‌های UI و Data Access پروژه را دارد (شکل ۳.۴).



شکل ۳.۴ معناری نرم‌افزاری پروژه O3

اطلاعات از UI به لایه Domain منتقل می‌شوند که این لایه وظیفه نمایش اطلاعات به کاربر را دارد که طبق الگو MVVM توسعه داده شده است. علاوه بر سه پوششی Models و ViewModels و Views پوشش‌های دیگری نیز وجود دارند که همگی آنها جهت پیشبرد اهداف همان سه پوشش توسعه داده شده‌اند (شکل ۴.۴).



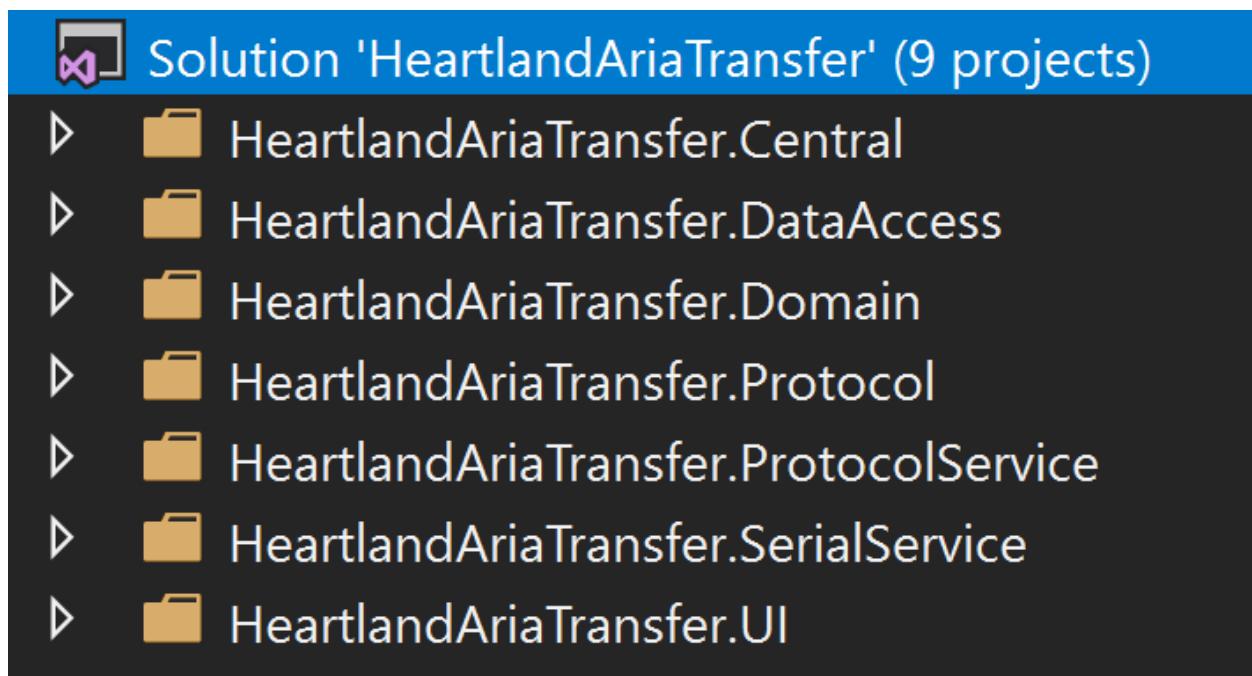
شکل ۴.۴ معناری نرم‌افزاری لایه UI پروژه O3

اطلاعات از لایه Domain به لایه Data Access منتقل می‌شوند که این لایه وظیفه ذخیره و بازیابی اطلاعات از پایگاه داده SQL Server را دارد (اسکریپت مربوط به پایگاه داده O3 در بخش پیوست موجود است). علاوه بر لایه‌های ذکر شده دو لایه دیگر وجود دارد؛ یک لایه به نام Entities شامل مدل‌های پایگاهداده‌ای است و استفاده آن در لایه‌های Data Access، Domain و UI است و لایه دیگر، لایه CRC است که وظیفه بررسی این مورد را دارد که آیا بسته‌های دریافتی صحیح دریافت شده‌اند یا خیر. استفاده آن در لایه Protocol است.

۳.۱.۴ پیاده‌سازی پروژه HeartlandAriaTransfer

این پروژه، پروژه اصلی است که وظایف زیر را دارد (شکل ۵.۴):

- ارتباط دوطرفه با آریا.
- ارتباط دوطرفه با سانترال.
- نمایش تاریخچه‌های آربا، ونتیلاتور و O3.
- ارتباط با پروژه‌های O3 و Ventilator.

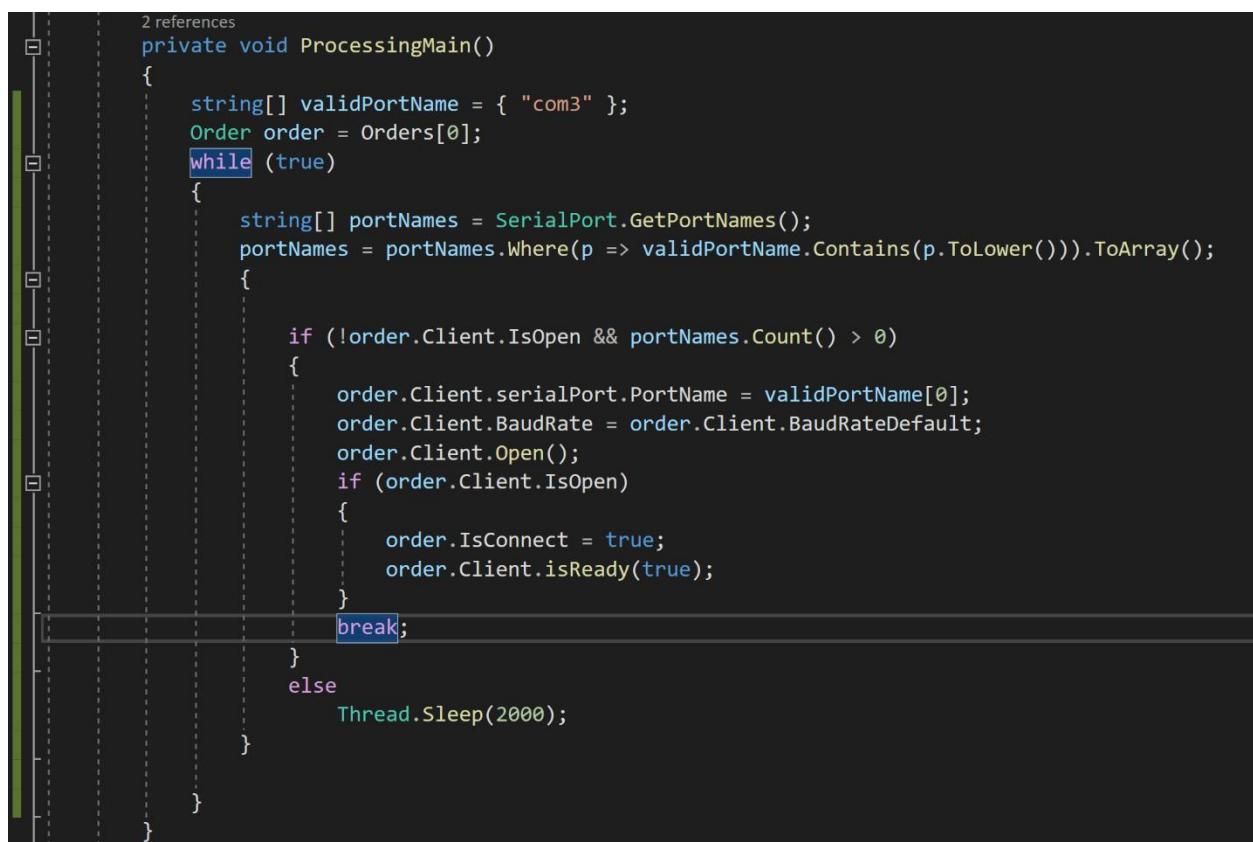


شکل ۵.۴ معماری نرمافزاری پروژه HeartlandAriaTransfer

معماری این پروژه هم مشابه پروژه‌های قبلی است که توضیح مختصری قبلاً داده‌ایم و قصد داریم در HeartlandAriaTransfer.SerialService این بخش به صورت مفصل‌تر توضیح بدهیم. در ابتدا لایه توسعه داده می‌شود تا در ابتدای کار بستر انتقال داده بین نرم‌افزار دستگاه Viewer و دستگاه آریا برقرار شود. این لایه وظایف زیر را دارد:

- باز کردن درگاه.
- خواندن داده‌های آریا از طریق درگاه باز شده.
- بستن درگاه.
- ارسال داده‌ها به آریا از طریق درگاه باز شده.

درگاه اختصاص داده شده به آریا COM3 می‌باشد که نحوه‌ی بازکردنش به شکل زیر (شکل ۶.۴) است.



```

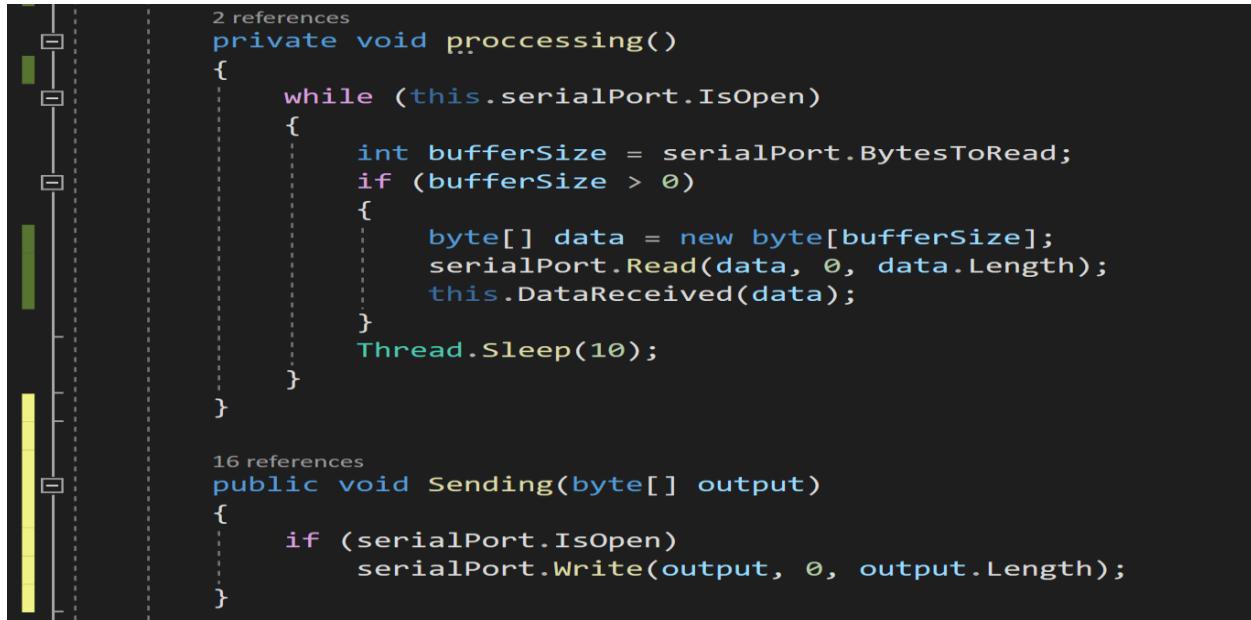
2 references
private void ProcessingMain()
{
    string[] validPortName = { "com3" };
    Order order = Orders[0];
    while (true)
    {
        string[] portNames = SerialPort.GetPortNames();
        portNames = portNames.Where(p => validPortName.Contains(p.ToLower())).ToArray();
        {

            if (!order.Client.IsOpen && portNames.Count() > 0)
            {
                order.Client.serialPort.PortName = validPortName[0];
                order.Client.BaudRate = order.Client.BaudRateDefault;
                order.Client.Open();
                if (order.Client.IsOpen)
                {
                    order.IsConnect = true;
                    order.Client.isReady(true);
                }
                break;
            }
            else
                Thread.Sleep(2000);
        }
    }
}

```

شکل ۶.۴ کد مربوط به باز کردن درگاه نرم‌افزار.

نحوه خواندن و نوشتمن داده‌ها روی درگاه باز شده هم به شکل زیر (شکل ۷.۴) است.



```

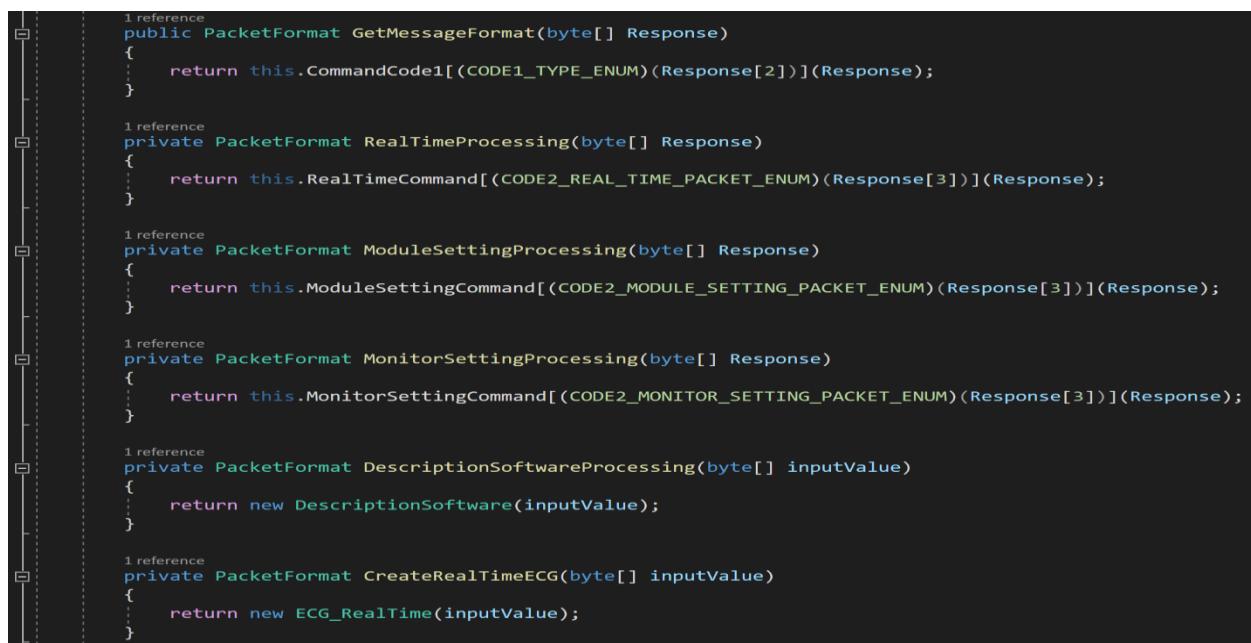
2 references
private void processing()
{
    while (this.serialPort.IsOpen)
    {
        int bufferSize = serialPort.BytesToRead;
        if (bufferSize > 0)
        {
            byte[] data = new byte[bufferSize];
            serialPort.Read(data, 0, data.Length);
            this.DataReceived(data);
        }
        Thread.Sleep(10);
    }
}

16 references
public void Sending(byte[] output)
{
    if (serialPort.IsOpen)
        serialPort.Write(output, 0, output.Length);
}

```

شکل ۷.۴ کد مربوط به خواندن و نوشتمن داده‌ها.

سپس طبق پروتکل دستگاه آریا لایه HeartlandAriaTransfer.Protocol توسعه داده می‌شود که وظیفه تبدیل داده‌های خام از جنس بایت به اطلاعات و بالعکس را دارد. در شکل زیر (شکل ۸.۴) بخشی از کدهای مربوط به این لایه را مشاهده می‌کنید.



```

1 reference
public PacketFormat GetMessageFormat(byte[] Response)
{
    return this.CommandCode1[(CODE1_TYPE_ENUM)(Response[2])](Response);
}

1 reference
private PacketFormat RealTimeProcessing(byte[] Response)
{
    return this.RealTimeCommand[(CODE2_REAL_TIME_PACKET_ENUM)(Response[3])](Response);
}

1 reference
private PacketFormat ModuleSettingProcessing(byte[] Response)
{
    return this.ModuleSettingCommand[(CODE2_MODULE_SETTING_PACKET_ENUM)(Response[3])](Response);
}

1 reference
private PacketFormat MonitorSettingProcessing(byte[] Response)
{
    return this.MonitorSettingCommand[(CODE2_MONITOR_SETTING_PACKET_ENUM)(Response[3])](Response);
}

1 reference
private PacketFormat DescriptionSoftwareProcessing(byte[] inputValue)
{
    return new DescriptionSoftware(inputValue);
}

1 reference
private PacketFormat CreateRealTimeECG(byte[] inputValue)
{
    return new ECG_RealTime(inputValue);
}

```

شکل ۸.۴ کد مربوط به تفسیر داده.

سپس لایه HeartlandAriaTransfer.ProtocolService توسعه داده می‌شود که وظیفه دریافت و مدیریت داده‌های دریافتی از لایه HeartlandAriaTransfer.SerialService و هچنین تبدیل داده‌ها به اطلاعات و بالعکس را، از طریق لایه HeartlandAriaTransfer.Protocol دارد. در شکل زیر (شکل ۹.۴) بخشی از کدهای مربوط به این لایه را مشاهده می‌کنید.

```

1 reference
private void SerialService_DataReceived(byte[] input)
{
    byte[] inputValid = new byte[input.Length + remainedInput.Length];
    Buffer.BlockCopy(remainedInput, 0, inputValid, 0, remainedInput.Length);
    Buffer.BlockCopy(input, 0, inputValid, remainedInput.Length, input.Length);
    if (inputValid.Length < 9)
    {
        remainedInput = new byte[inputValid.Length];
        Buffer.BlockCopy(inputValid, 0, remainedInput, 0, remainedInput.Length);
        return;
    }
    int i = 0;
    while (i < inputValid.Length - 2)
    {
        if (inputValid[i] == 170 && inputValid[i + 1] == 85)
        {
            if (i + 3 < inputValid.Length && (((inputValid[i + 2] << 8) + inputValid[i + 3]) + i + 5/*2Header+1Header*/ == 0x1A))
            {
                int lenPacket = ((inputValid[i + 2] << 8) + inputValid[i + 3]);
                byte[] packet = new byte[lenPacket + 3/*2Len+1CheckSum*/];
                Buffer.BlockCopy(inputValid, i + 2/*Header*/, packet, 0, packet.Length);
                int CkeckSum = 0;
                for (int check = 0; check < packet.Length - 1; check++)
                {
                    CkeckSum += packet[check];
                    if (check < packet.Length - 2 && packet[check] == 170 && packet[check + 1] == 85)
                    {
                    }
                }
                PacketFormat ReceivedMessageFormat = null;
                if ((byte)CkeckSum == packet[packet.Length - 1])
                {
                }
            }
        }
    }
}

```

شکل ۹.۴ کد مربوط به مدیریت داده‌ها.

سپس لایه HeartlandAriaTransfer.Domain که هسته مرکزی این معماری و پروژه است توسعه داده می‌شود که وظیفه توزیع اطلاعات بین لایه‌های HeartlandAriaTransfer.UI و HeartlandAriaTransfer.Central و HeartlandAriaTransfer.DataAccess پروژه را دارد. در شکل صفحه بعد (شکل ۱۰.۴) بخشی از کدهای مربوطه به این لایه را مشاهده می‌کنید.

شکل ۱۰.۴ کد مربوط به هسته مرکزی معماری نرم افزار.

اطلاعات از لایه Domain به لایه Data Access منتقل می‌شوند که این لایه وظیفه ذخیره و بازیابی اطلاعات از پایگاه داده SQL Server را دارد. یکی از مزایای این نوع مرزبندی بین لایه‌ها این است که اگر روزی نوع پایگاه داده را بخواهیم عوض کنیم، کل پروژه دستخوش تغییرات نمی‌شود، بلکه با تغییرات جزئی در لایه Data Access می‌توانیم این کار را انجام دهیم (اسکریپت مربوط به پایگاه داده HeartlandAriaTransfer در بخش پیوست موجود است).

در این پژوهه اطلاعات آریا می‌باشد که این امر توسط لایه HeartlandAriaTransfer.Central انجام می‌شود. لایه HeartlandAriaTransfer.Central مطالبه

شکل صفحه بعد (شکل ۱۱.۴)، دارای سه زیر لایه است:

- Central.CentralService.NetStandard
 - Central.Net.Sockets.NetStandard
 - Central.SMNPProtocol.NetStandard



شکل ۱۱.۴ لایه بندی بخش‌های مربوط به سانترال.

شرح کار این لایه‌ها بدین صورت است که لایه Central.CentralService.NetStandard وظیفه دریافت و مدیریت اطلاعات آریا جهت ارسال به سانترال دارد، لایه Central.Net.Sockets.NetStandard وظیفه برقراری ارتباط با سانترال از طریق شبکه را دارد و لایه Central.SMNPProtocol.NetStandard طبق پروتکل سانترال، وظیفه تبدیل اطلاعات آریا به داده‌های سانترال را دارد. در شکل زیر (شکل ۱۲.۴) بخشی از کدهای مربوط به لایه HeartlandAriaTransfer.Central را مشاهده می‌کنید.

```

8 references
internal abstract class BaseSender< TEntity > where TEntity : class
{
    private readonly Func< byte[], int> _senderMethods;
    private readonly NumberOfSubsystem _signal1SubsystemsNumber;
    private readonly EcgFilter _ecgFilter;
    protected int PacketCounter;
    protected List< DataFormat > FinalDataFormats;

    12 references
    public void SetEcgFilter(Enumeration.MODULE_SETTING_PACKET_ECG_FILTER filter) => this._ecgFilter.Value = (byte)filter;

    7 references
    protected BaseSender(Func< byte[], int> senderMethods)
    {
        this._senderMethods = senderMethods;
        this._signal1SubsystemsNumber = new NumberOfSubsystem() { Value = 13 };
        this._ecgFilter = new EcgFilter() { Value = 2 };
    }

    11 references
    protected virtual int SendData(List< DataFormat > data)
    {
        var signalPacket = new PacketFormat();
        signalPacket.AddDataFormat(this._signal1SubsystemsNumber);
        signalPacket.AddDataFormat(this._ecgFilter);
        foreach (var dataFormat in data)
            signalPacket.AddDataFormat(dataFormat);
        return this._senderMethods(signalPacket.GetPacketOutput());
    }
}

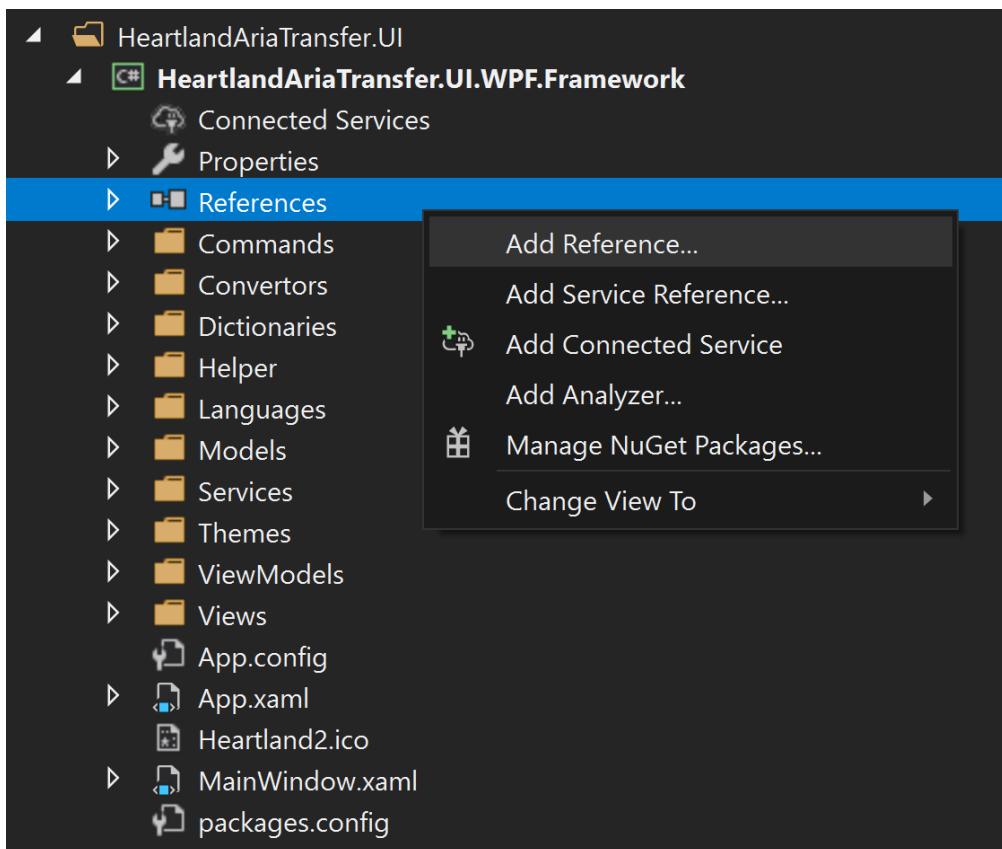
```

شکل ۱۲.۴ کد مربوط به سانترال.

اطلاعات از لایه Domain به لایه HeartlandAriaTransfer.UI منتقل می‌شود که وظیفه این لایه، نمایش اطلاعات و تعامل با کاربر است. در بخش بعدی نحوه کار لایه‌ی UI را به طور کامل توضیح می‌دهیم.

۲.۴ پیاده‌سازی لایه UI پروژه HeartlandAriaTransfer

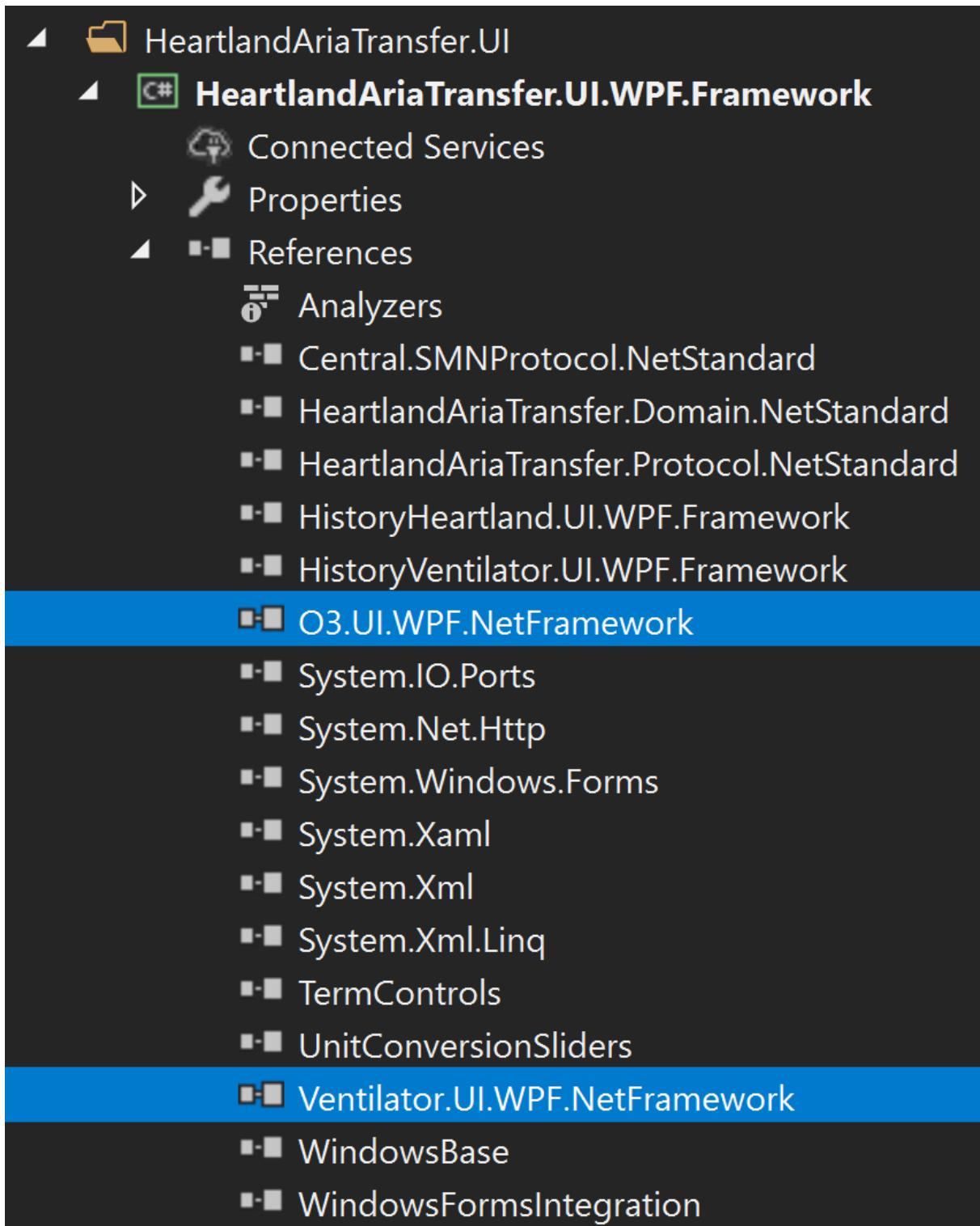
همان‌طور که در بخش‌های گذشته ذکر شد لایه UI با معماری MVVM پیاده‌سازی می‌شود و نحوه‌ی کار، کاملا مشابه با لایه UI دو پروژه‌ی Ventilator و O3 است که در بخش قبلی توضیح داده شده است. نکته قابل توجهی که حائز اهمیت است این است که لایه UI این پروژه مطابق شکل (شکل ۱۳.۴) به لایه‌های پروژه‌های Ventilator و O3، ارجاع^۱ دارد و می‌شود در این پروژه از رابط Visual Studio دو پروژه‌ی دیگرهم استفاده کنیم. برای این کار مطابق تصویر زیر، در نرمافزار Visual Studio در لایه UI روی کلمه References کلیک راست کرده و گزینه... Add Reference را انتخاب کرده و خروجی پروژه‌های Ventilator و O3 که یک فایل با پسوند .exe است را انتخاب می‌کنیم.



شکل ۱۳.۴ نحوه ارجاع به نرمافزارها جانبی.

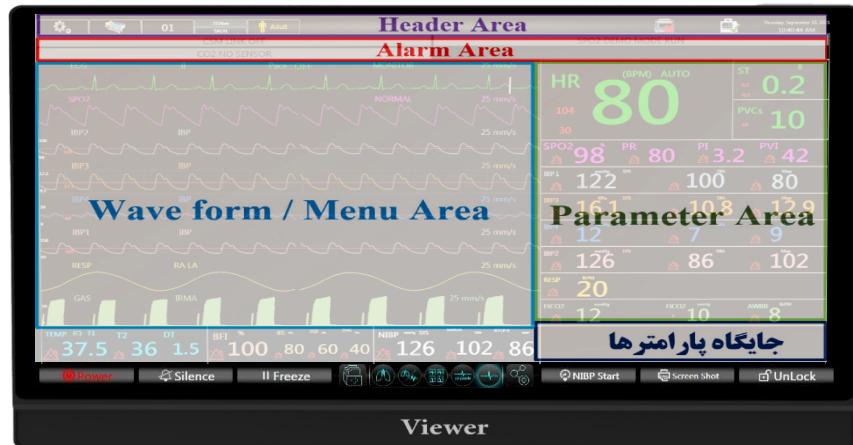
^۱ Reference

پس از انتخاب فایل خروجی، مطابق شکل (شکل ۱۴.۴)، پروژه‌های Ventilator و O3 به لیست اضافه می‌شوند.



شکل ۱۴.۴ لیست وابستگی‌ها.

در ادامه صفحاتی که در لایه‌ی UI ساخته شده‌اند و کاربر با آن‌ها تعامل می‌کند را بررسی می‌کنیم.
اکثر صفحات مطابق شکل زیر (شکل ۱۵.۴) دارای بخش‌های زیر هستند:



شکل ۱۵.۴ ساختار صفحه نرم‌افزار.

- بخش Header Area: نمایانگر مشخصات بیمار و شرایط کارکرد سیستم.
 - بخش Alarm Area: نمایش هشدارها.
 - بخش Parameter Area: محدوده نمایش پارامترهای عددی.
 - بخش Wave form / Menu Area: محدوده نمایش سیگنال‌ها.
 - جایگاه پارامترها: انتخاب پارامترهای مورد نظر با روش Drag & Drop^۱.
- نرم‌افزار شامل ۹ صفحه زیر است:

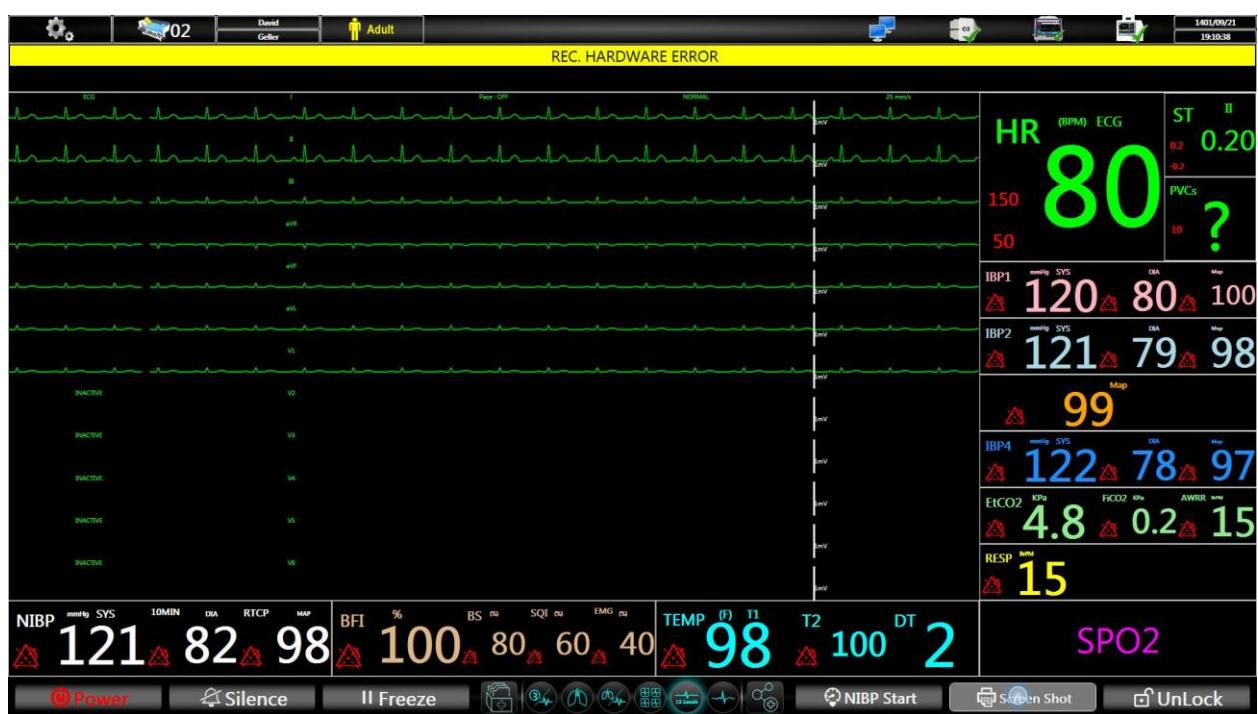
۱. صفحه اول جهت نظارت تمامی پارامترها و سیگنال‌های دستگاه جانبی آریا (شکل ۱۶.۴).

^۱ کشیدن و رها کردن



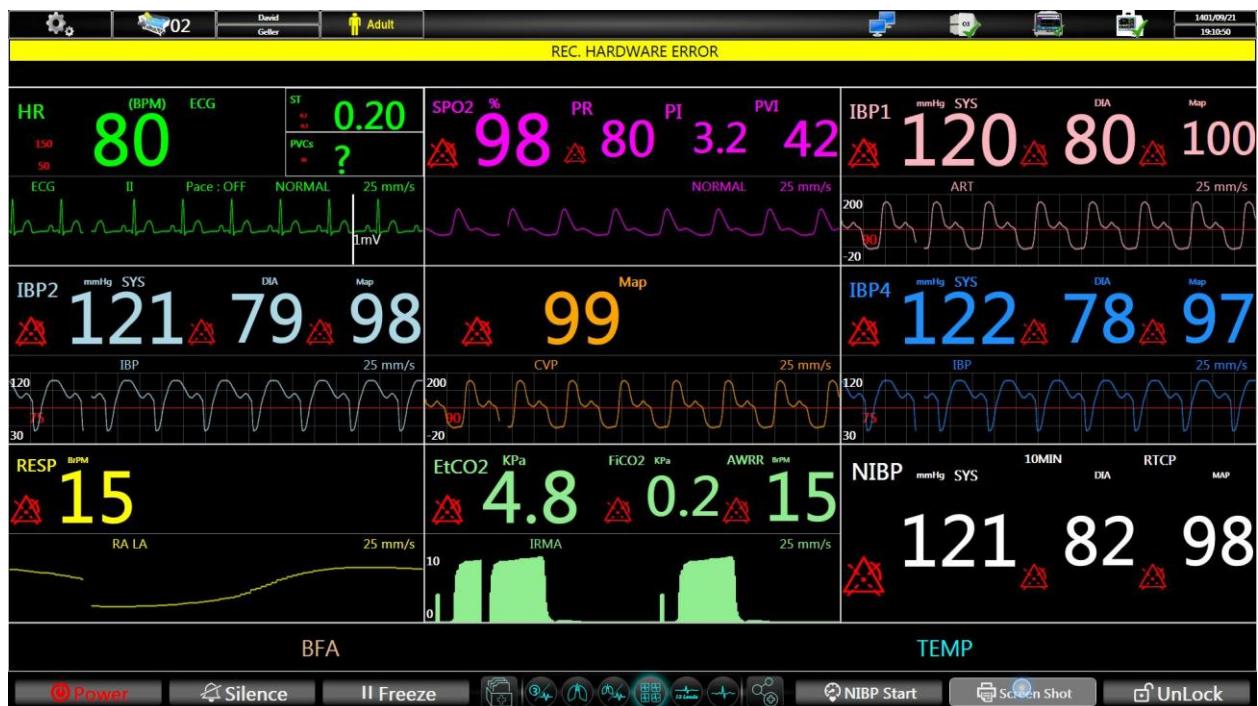
شکل ۱۶.۴ صفحه اول نرم‌افزار.

۲. صفحه دوم جهت نظارت تمامی پارامترها و تمامی سیگنال‌های مربوط به ECG دستگاه جانبی آریا (شکل ۱۷.۴).



شکل ۱۷.۴ صفحه دوم نرم‌افزار.

۳. صفحه سوم جهت نظارت تمامی پارامترها و سیگنال‌های دستگاه جانبی آریا به شکلی متفاوت از صفحه اول (شکل ۱۸.۴).



شکل ۱۸.۴ صفحه سوم نرم‌افزار.

۴. صفحه چهارم جهت نظارت تمامی پارامترها و سیگنال‌های دستگاه جانبی آریا و ونتیلاتور به صورت همزمان (شکل ۱۹.۴).



شکل ۱۹.۴ صفحه چهارم نرم‌افزار.

۵. صفحه پنجم جهت نظارت تمامی پارامترها و سیگنال‌های دستگاه جانبی ونتیلاتور (شکل ۲۰.۴).



شکل ۲۰.۴ صفحه پنجم نرم افزار.

۶. صفحه ششم جهت نظارت تمامی پارامترها و سیگنال‌های دستگاه جانبی آریا و مازول O3 به صورت همزمان (شکل ۲۱.۴).



شکل ۲۱.۴ صفحه ششم نرم افزار.

۷. صفحه هفتم جهت نظارت تمامی پارامترها و سیگنال‌های دستگاه جانبی آریا و تاریخچه پارامترها، سیگنال‌ها و هشدارها به صورت همزمان (شکل ۲۲.۴).



شکل ۲۲.۴ صفحه هفتم نرم‌افزار.

۸. صفحه هشتم جهت نظارت تمامی پارامترها و سیگنال‌های دستگاه جانبی آریا و تاریخچه پارامترها، سیگنال‌ها و هشدارهای ونتیلاتور به صورت همزمان (شکل ۲۳.۴).



شکل ۲۳.۴ صفحه هشتم نرم‌افزار.

۹. صفحه نهم جهت مشاهده عکس‌های گرفته شده از سیستم و مدیریت آنها (کپی و انتقال عکس‌ها به USB و حذف آنها) (شکل ۲۴.۴). (شکل ۲۴.۴)



شکل ۲۴.۴ صفحه نهم نرم‌افزار.

۳.۴ تجزیه و تحلیل داده‌ها (ارزیابی و کارایی)

همان طور که در فصل اول بررسی کردیم هدف از این پژوهش این است که نرم‌افزار دستگاه Viewer، بتواند بر بستر سخت‌افزاری کامپیوتر گیادا، با هر چهار دستگاه جانبی آریا، ونتیلاتور، مازول O3 و سانترال، به صورت همزمان ارتباط برقرار کند و بسته‌های انتقالی را به صورت صحیح و بدون خطا دریافت کند. یکی از مهمترین آزمون‌هایی که مورد ارزیابی قرار گرفته شده همین آزمون صحت دریافت داده‌ها از دستگاه‌های جانبی آریا، ونتیلاتور و مازول O3 است.

نرخ داده‌های دریافتی از دستگاه آریا که باید با نرم‌افزار Viewer و سخت افزار گیادا مورد پردازش قرار دهیم حداقل ۴۰ میلی‌ثانیه است یعنی هر ۴۰ میلی‌ثانیه یک بسته دریافت می‌شود. هر ۱ ثانیه معادل ۱۰۰۰ میلی‌ثانیه است پس در هر ثانیه ۲۵ بسته دریافت می‌کنیم: [۷]

$$1000 \div 40 = 25$$

در ساختار بسته‌های انتقالی دستگاه آریا بایتی به اسم Checksum وجود دارد که جمع داده‌های بسته است؛ بنابراین، بعد از دریافت هر بسته، داده‌های هر بسته را باهم جمع می‌کنیم و اگر با Checksum برابر باشد یعنی بسته به درستی دریافت شده است. خوشبختانه در تست طولانی مدتی که از سیستم Viewer گرفته شد، تمامی مقادیر Checksum‌ها درست بودند و این یعنی می‌شود از صحت داده‌های آریا در دستگاه Viewer مطمئن بود.

نرخ داده‌های دریافتی از دستگاه ونتیلاتور که باید مورد پردازش قرار دهیم حداقل ۱۰ میلی‌ثانیه است پس در هر ثانیه ۱۰۰ بسته دریافت می‌کنیم: [۸]

$$1000 \div 10 = 100$$

در ساختار بسته‌های انتقالی دستگاه ونتیلاتور بایتی به اسم CRC وجود دارد که جمع داده‌های بسته است؛ بنابراین، بعد از دریافت هر بسته، داده‌های هر بسته را باهم جمع می‌کنیم و اگر با Checksum برابر باشد یعنی بسته به درستی دریافت شده است. خوشبختانه در تست طولانی مدتی که از سیستم Viewer گرفته شد، تمامی مقادیر Checksum‌ها درست بودند و این یعنی می‌شود از صحت داده‌ها ونتیلاتورهم در دستگاه Viewer مطمئن بود.

نرخ داده‌های دریافتی از مژول O3 که باید مورد پردازش قرار دهیم کمتر از دستگاه‌های آریا و ونتیلاتور است و هر ۲ ثانیه یک بسته انتقال پیدا می‌کند [۹]. در ساختار بسته‌های انتقالی مژول O3 دو بایت به اسم CRC وجود دارد که جمع داده‌های بسته است؛ بنابراین، بعد از دریافت هر بسته، داده‌های هر بسته را باهم جمع می‌کنیم و اگر با CRC برابر باشد یعنی بسته به درستی دریافت شده است. خوشبختانه در تست طولانی مدتی که از سیستم Viewer گرفته شد، تمامی مقادیر Checksum‌ها درست بودند و این یعنی می‌شود از صحت داده‌ها مژول O3 هم در دستگاه Viewer مطمئن بود.

براساس یکی دیگر از پارامترهای سنجش کارآمدی نرمافزار که در فصل اول اشاره کردیم؛ یکی دیگر از آزمون‌های مهم، حصول اطمینان از کارکرد سیستم در طولانی مدت و شرایط حساس است. طبق روشی که آقای Guangye Chen و همکارانشون در سال ۲۰۲۲ میلادی ارائه کرده‌اند [۱۷] نحوه ارزیابی این آزمون بدین صورت است که میزان استفاده برنامه از منابع سیستم واحد کنترل مرکزی و حافظه را در طول زمان باید مورد بررسی قرار داد؛ اگر میزان استفاده از منابع با گذر زمان افزایش پیدا کند، حتی اگر مقدار ناچیزی افزایش پیدا کند و روند به صورت صعودی باشد، بالاخره بعد از چند شبانه

روز برنامه متوقف می‌شود. خوشبختانه در تست طولانی مدتی که از سیستم Viewer گرفته شد، میزان استفاده برنامه از منابع سیستم یکسان ماند و افزایشی نداشت، در نتیجه می‌توان به سیستم در شرایط حساس‌هم اطمینان کرد.

طبق پارامترهای سنجش قابلیت‌های توسعه و نگهداری، نرمافزاری این قابلیت‌ها را دارد که بتواند نیازهای متغیر مشتریان را تکامل یابد [۲۳]. همان‌طور که در بخش پیاده‌سازی (بخش ۱.۴) ذکر کردیم، دستگاه Viewer در درجه اول جهت مانیتورینگ دستگاه جانبی آریا و ارسال داده‌های آن به سانترال توسعه یافته بود و با گذر زمان، نیازهای مشتریان تغییر پیدا کرد و ارتباط با دستگاه جانبی ونتیلاتور و مازول O3 هم به دستگاه Viewer اضافه شد؛ در نتیجه روش ارائه شده این پژوهش قابلیت‌های توسعه و نگهداری دارد.

تمامی آزمون‌ها نرمافزاری یا سخت‌افزاری نیستند بلکه بخش زیادی از آزمون‌ها پزشکی هستند. آزمون‌های زیادی توسط افرادی که دیدگاه مهندسی پزشکی دارند گرفته شده است. برای مثال در برنامه نمایش سیگنال‌ها با سرعت‌های $12/5$ ، 25 ، 50 میلی‌متر بر ثانیه امکان پذیر است. سرعت 25 میلی‌متر بر ثانیه، از دیدگاه مهندس پزشکی یعنی اینکه فاصله‌ی 1 تا 5 میلی‌متر در یک ثانیه می‌باشد. آزمون‌هایی که افراد متخصص و مهندسین پزشکی از سیستم گرفته‌اند مشکلی نبوده.

نرمافزار دستگاه Viewer در مقایسه با نرمافزار دستگاه‌های مشابه خارجی مثل دستگاه‌های سری BeneVision که در فصل اول (شکل ۲) به آن اشاره کردیم؛ علاوه بر ارتباط با دستگاهی مثل آریا و ارسال داده‌های آن به سانترال، مانیتورینگ دو دستگاه جانبی دیگر یعنی ونتیلاتور و مازول O3 هم انجام می‌دهد که جزو نوآوری‌های این پژوهش بود.[۲۱]

^۱ نقطه بالایی سیگنال

۵

فصل پنجم

بحث و نتیجه‌گیری همراه با مطالعات آتی

بحث و نتیجه‌گیری همراه با مطالعات آتی

در این پژوهش به دنبال این بودیم که علاوه بر کار تحقیقاتی، به کار عملیاتی‌هم بپردازیم و فقط به یافته‌های علمی اکتفا نکنیم و آن‌ها را در محیط عملیلاتی پیاده‌سازی کنیم. به دلیل حساسیت بالا در حوزه پزشکی سیستم باید انواع و اقسام آزمون‌ها را پشت سر می‌گذشت تا بتوان از آن به عنوان یک سیستم مانیتورینگ بلاذرنگ در درمانگاه‌ها و مراکز پزشکی استفاده کرد؛ البته در طول توسعه و طراحی سیستم از شبیه سازهایی که وظیفه تولید کردن داده‌های مشابه با داده‌های واقعی علائم حیاتی بیمار را دارند و نیاز ما را برای تست رفع میکرند استفاده می‌شد، اما در نهایت به دلیل حساسیت بالا در حوزه سلامت و پزشکی و همچنین برای اخذ نشان و گواهینامه^۱ CE برای صادرات، این سیستم در محیط عملیاتی یعنی همراه با بیمار واقعی هم مورد ارزیابی قرار گرفت.

از نقاط عطف نرم‌افزار دستگاه Viewer نسبت به محصول بقیه رقبا مثل شرکتهای [6] در کشور آمریکا، [5] در کشور چین و ... می‌توان به موارد زیر اشاره کرد:

- مانیتورینگ همزمان دستگاه ونتیلاتور در کنار علائم حیاتی بیمار حاصل از دستگاه آریا.
- مانیتورینگ همزمان مازول O3 در کنار علائم حیاتی بیمار.
- امکان نظارت تاریخچه علائم حیاتی بیمار.
- ... و ...

۱.۵ یافته‌های پژوهش

بر اساس مطالعات صورت گرفته که بخشی از آن‌ها را در فصل دوم (پیشینه پژوهش) بررسی کردیم به یافته‌های علمی معتبری در حوزه نرم‌افزار و برنامه‌نویسی دست پیدا کردیم. بعد از اینکه مرحله تحقیقات پروژه انجام شد، وارد مرحله توسعه شدیم و یافته‌های علمی مرحله قبل را در این مرحله پیاده سازی کردیم و مورد ارزیابی (فصل چهارم) قرار دادیم. از یافته‌های این پژوهش می‌توان به موارد زیر اشاره کرد:

^۱ استاندارد اروپا

- مرزبندی مناسب، کلید اساسی در طراحی معماری نرمافزار کاربردی است.
- فراهم شدن یک دیتاست با ارزش از علائم حیاتی بیمار است که می‌تواند بسیار مورد کاربرد قرار بگیرد؛ مثلاً اگر بیماری سکته مغزی کرده است، با در دست داشتن تاریخچه‌ی علائم حیاتی آن بیمار و بررسی آن‌ها می‌توان به نتایج مهمی در حوزه پزشکی رسید که چه پارامترهایی باعث سکته بیمار شده است.
- با استفاده از نرمافزار پیشنهاد شده، می‌توان روشی جهت انتقال داده‌ها (ارسال و دریافت) در حوزه‌های مختلف مثل مخابرات و ... ارائه داد.

۲.۵ محدودیت‌ها و مشکلات پژوهش

از عمدۀ ترین ارکان تحقیق و پژوهش دسترسی به آمار و اطلاعات است اما دسترسی به کتب، مجلات، آمار، بانک‌های اطلاعاتی و ... به دلیل وجود تحریم‌ها و مسئله فیلترینگ^۱ در کشور به راحتی ممکن نیست. علاوه‌براین هدف از این پژوهش این بود که وارد فاز عملیاتی شویم و پروژه برنامه‌نویسی پیاده‌سازی بشود، به همین دلیل یافتن منابعی که مرتبط باشند به راحتی ممکن نبود زیرا سازمان‌هایی که محصولاتی مشابه دستگاه Viewer را پیاده سازی کرده‌اند اطلاعات مربوط به محصولات را نشر نمی‌دهند و جزو حریم خصوصی سازمان‌ها محسوب می‌شود.

۳.۵ پیشنهادهایی در خصوص پژوهش‌های آتی

پایان این پژوهش، می‌تواند شروع پژوهش‌های مهم دیگری باشد؛ به دلیل دستیابی به دیتاست‌هایی از داده‌های مربوط به علائم حیاتی بیماران، می‌توان پژوهش‌های زیاد دیگری را در صنعت پزشکی در پیش گرفت که از دیتاست‌های فراهم شده‌ی حاصل این پژوهش استفاده کنند؛ مثلاً، با تحلیل وضعیت سیگنال قلب و دیگر پارامترهای علائم حیاتی می‌توان با دقت بالایی به این نتیجه رسید که بیمار در چه زمانی نیاز به شوک دارد و موقعی که نیاز به شوک پیدا شد دستگاهی تهیه شود که به صورت خودکار

^۱ فیلترینگ، عبارت است از محدود کردن دسترسی کاربران اینترنت به وب‌گاه‌ها و خدمات اینترنتی.

شوك را به بیمار وارد کند. خیلی از بیمارانی که فوت کردند اگر کمی زودتر(در حد ۵۰۰ میلی ثانیه) شوک وارد می‌شد شاید احیا می‌شدند و زنده می‌مانندند. از آنجایی که دقیق دستگاه در محاسبات از انسان بیشتر است، احتمال اعمال شوک به بیمار در زمان دقیق‌تر از طریق دستگاه بیشتر است. علاوه بر این در نسخه‌های بعدی نرم‌افزار دستگاه **Viewer** می‌توان مانیتورینگ دستگاه جدید دیگری را اضافه کرد و همچنان قابلیت‌هایی از قبیل چاپ^۱ را هم اضافه کرد تا کاربر بتواند در صورت دلخواه، وضعیت بیمار را روی کاغذ هم داشته باشد و از آن استفاده‌های گوناگون بکند.

^۱ Print

منابع و مراجع

- Hoa Hong Nguyen, Farhaan Mirza, Muhammad Asif Naeem, Minh Nguyen - “A review on iot healthcare monitoring applications and a vision for transforming sensor data into real-time clinical feedback” - *Proceedings of the 2017 IEEE 21st international conference on computer supported cooperative work in design* - 2017 [١]
- Y. Nait Malek, A. Kharbouch, H. Khoukhi, M. Bakhouya, V. De Florio, D. Ouadghiri, S. Latre, C. Blondia - “On the use of iot and big data technologies for realtime monitoring and data processing” - *The 7th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH 2017)* - 2017 [٢]
- Hulya Vural, Murat Koyuncu - “Does domain driven design lead to finding the optimal modularity of a microservice” - *IEEE*, Electronic ISSN: 2169-3536, INSPEC Accession Number: 20360932, Page(s): 32721 , 32733 – 2021 [٣]
- Rajalakshmi Krishnamurthi, Adarsh Kumar , Dhanalekshmi Gopinathan, Anand Nayyar, and Basit Qureshi - “An overview of iot sensor data processing, fusion, and analysis techniques” - *Department of Computer Science and Engineering, Jaypee Institute of Information Technology, Noida 201309, India* – 2020 [٤]
- N. Srinidhi, S. Dilip Kumar, K. Venugopal - “Network optimizations in the internet of things” - *Engineering Science and Technology, an International Journal* 22 (2019) 1-21 : Received 27 February 2018, Revised 7 September 2018, Accepted 9 September 2018, Available online 22 September 2018, Version of Record 10 February 2019 - 2018 [٥]
- M. Arun Kumar , R. Vimala b, K.R. Aravind Britto - “A cognitive technology based healthcare monitoring system and medical data transmission” - *Measurement* 146 (2019) 322–332 : Received 12 November 2018, Revised 16 January 2019, Accepted 5 March 2019, Available online 2 May 2019, Version of Record 1 July 2019 - 2019 [٦]
- Saadatco company - Reza Shalbaf, Kamran Gholami, Mahmoud Merati - “Aria protocol document: global provider of high quality and cost effective patient monitoring systems accompanying patients - 2021 [٧]
- Saadatco company - Hamid Azizzadeh, Hesam Ekhtiyar, Amir Hossein Mehrnam, Mojtaba Rezaei - Ventilator protocol document: Global provider of high quality and cost effective patient monitoring systems accompanying patients - 2021 [٨]

- Masimo company - O3 protocol document: Global medical technology company that develops and produces a wide array of industry-leading monitoring technologies - 2018 [٩]
- Saadat company - Nahid Asgari, Amir Hosseing Panahi, Kamran Gholami, Ahmad Mansourikhah, Mehrdad Ghodousi, Hamid Vaghari, Mahmoud Merati - Central protocol document: Global provider of high quality and cost effective patient monitoring systems accompanying patients - 2021 [١٠.]
- Eric Evans – Domain driven design book - Printed in the United States of America. Published simultaneously in Canada - 2003 [١١]
- Saadat company – Sepehr Karimi - Mohammad Hossein Nezhadhendi - Ahmad Mansourikhah - Ventilator project – 2022 [١٢]
- Saadat company – Ahmad Mansourikhah, Sepehr Karimi, Mohammad Hossein Nezhadhendi - O3 project – 2022 [١٣]
- Saadat company – Mohammad Hossein Nezhadhendi, Ahmad Mansourikhah, Sepehr Karimi - HeartlandAriaTransfer project – 2022 [١٤]
- Saadat company – Sepehr Karimi, Mohammad Hossein Nezhadhendi – Elham Yousefi - Ahmad Mansourikhah - Project requirements -2021 [١٥]
- Saadat company – Ahmad Mansourikhah, Elham Yousefi, Sepehr Karimi, Mohammad Hossein Nezhadhendi - Viewer device manual – 2022 [١٦]
- Nigel Tan, Robert F. Bird, Guangye Chen, Scott V. Luedtke, Brian J. Albright and Michela Taufer - “Analysis of vector particle in cell(VPIC) memory usage optimizationson cutting edge computer architectures” - IEEE - Electronic ISSN: 2169-3536, INSPEC Accession Number: 20360932 - 2022 [١٧]
- Karyna Isaieva, Marc Fauvel, Nicolas Weber, Pierre-andré Vuissoz, Jacques Felblinger, Julien Oster, Freddy Odille - “A hardware and software system for MRI applications requiring external device data” - IEEE - Electronic ISSN: 2169-3536, INSPEC Accession Number: 20360932 - 2022 [١٨]
- David Heidrich, Andreas Schreiber, Sebastian Oberdörfer - “Towards generating labeled property graphs for comprehending C# based software projects” - IEEE - Electronic ISSN: 2169-3536, INSPEC Accession Number: 20360932 – 2022 [١٩]
- Saadat company - Ali Mahmoodi, Amir Hosseing Panahi - Previous viewer project – 2017 [٢٠.]
- Mindray Company - Mindray BeneVision N-Series - Global medical technology company that develops and produces a wide array of industry - leading [٢١]

monitoring technologies - 2022

Epiloksa achdan, Dana kusumo - “Effect of MVVM architecture pattern on android based application performance” - *Jurnal Media Informatica* - ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online) - Page 1949-1955 - 2022 [٢٢]

Roger S. Pressman, Bruce Maxim - Software engineering: A practitioner's approach book - Printed in the United States of America. Published ACM - 2019 [٢٣]

پیوست ها

(پ-۱) اسکریپت مربوط به پایگاهداده Ventilator :

```
USE [VentilatorDB]
GO
/***** Object: Table [dbo].[Alarm]  Script Date: 1/23/2022 11:25:58 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Alarm](
    [id] [bigint] IDENTITY(1,1) NOT NULL,
    [Time] [bigint] NOT NULL,
    [Flag_Alarms_A1] [tinyint] NOT NULL,
    [Flag_Alarms_A2] [tinyint] NOT NULL,
    [Flag_Alarms_A3] [tinyint] NOT NULL,
    [Flag_Alarms_A4] [tinyint] NOT NULL,
    [Flag_Alarms_A5] [tinyint] NOT NULL,
    [Reserved] [smallint] NOT NULL,
CONSTRAINT [PK_Alarm] PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
= ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[AlarmCode]  Script Date: 1/23/2022 11:25:59 AM ****/
```

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[AlarmCode](
    [Id] [bigint] IDENTITY(1,1) NOT NULL,
    [Time] [bigint] NOT NULL,
    [Code] [smallint] NOT NULL,
    [AlarmLevel] [tinyint] NOT NULL,
CONSTRAINT [PK_AlarmCode] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
= ON) ON [PRIMARY]
) ON [PRIMARY]
GO
***** Object: Table [dbo].[AlarmSetting]  Script Date: 1/23/2022 11:25:59 AM
*****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[AlarmSetting](
    [id] [bigint] IDENTITY(1,1) NOT NULL,
    [Time] [bigint] NOT NULL,
    [Flag_Alarm] [tinyint] NOT NULL,
    [PpeakAlarmHigh] [tinyint] NOT NULL,
    [VtiLimitAlarm] [smallint] NOT NULL,
    [RateAlarmHigh] [tinyint] NOT NULL,
    [RateAlarmLow] [tinyint] NOT NULL,
    [ApneaTimeAlarm] [tinyint] NOT NULL,
    [VteAlarmHigh] [smallint] NOT NULL,
    [VteAlarmLow] [smallint] NOT NULL,
```

```

[VeAlarmHigh] [smallint] NOT NULL,
[VeAlarmLow] [smallint] NOT NULL,
[LeakAlarmHigh] [tinyint] NOT NULL,
[PEEPAlarmHigh] [tinyint] NOT NULL,
[PEEPAlarmLow] [tinyint] NOT NULL,
[Reserved] [smallint] NOT NULL,
CONSTRAINT [PK_AlarmSetting] PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
= ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[AlgorithmSettingS1]  Script Date: 1/23/2022 11:25:59
AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[AlgorithmSettingS1](
    [id] [bigint] IDENTITY(1,1) NOT NULL,
    [Time] [bigint] NOT NULL,
    [Flag_Set_S1] [tinyint] NOT NULL,
    [O2Setting] [tinyint] NOT NULL,
    [RateSetting] [tinyint] NOT NULL,
    [PeepSetting] [tinyint] NOT NULL,
    [PControlSetting] [tinyint] NOT NULL,
    [PSupportSetting] [tinyint] NOT NULL,
    [PHighSetting] [tinyint] NOT NULL,
    [PLowSetting] [tinyint] NOT NULL,
    [VentMode] [tinyint] NOT NULL,
    [TiSetting] [real] NOT NULL,
    [FTrigSetting] [tinyint] NOT NULL,

```

```
[PTrigSetting] [real] NOT NULL,  
[FPSetting] [tinyint] NOT NULL,  
[THighSetting] [real] NOT NULL,  
[TLowSetting] [real] NOT NULL,  
[TVSetting] [smallint] NOT NULL,  
[IBWSetting] [tinyint] NOT NULL,  
[AutoTlow] [tinyint] NOT NULL,  
[Reserved1] [tinyint] NOT NULL,  
[Reserved2] [smallint] NOT NULL,  
  
CONSTRAINT [PK_AlgorithmSettingS1] PRIMARY KEY CLUSTERED  
(  
    [id] ASC  
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS  
= ON) ON [PRIMARY]  
) ON [PRIMARY]  
GO  
***** Object: Table [dbo].[AlgorithmSettingS2]  Script Date: 1/23/2022 11:25:59  
AM *****  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
CREATE TABLE [dbo].[AlgorithmSettingS2](  
    [id] [bigint] IDENTITY(1,1) NOT NULL,  
    [Time] [bigint] NOT NULL,  
    [Flag_Set_S2] [tinyint] NOT NULL,  
    [Flag_Set_AlgorithmSet] [tinyint] NOT NULL,  
    [SighFactorSetting] [tinyint] NOT NULL,  
    [SighEverySetting] [tinyint] NOT NULL,  
    [SighNumSetting] [tinyint] NOT NULL,  
    [FlowPattSetting] [tinyint] NOT NULL,  
    [EsenSetting] [tinyint] NOT NULL,  
    [RiseTimeSetting] [tinyint] NOT NULL,
```

```
[BaseFlowSetting] [tinyint] NOT NULL,  
[PauseSetting] [tinyint] NOT NULL,  
[NebuTime] [tinyint] NOT NULL,  
[SuctionTime] [tinyint] NOT NULL,  
[HumidType] [tinyint] NOT NULL,  
[PimaxTime] [tinyint] NOT NULL,  
[Flag_Set_HW] [tinyint] NOT NULL,  
[PatientCat] [tinyint] NOT NULL,  
CONSTRAINT [PK_AlgorithmSettingsS2] PRIMARY KEY CLUSTERED  
(  
    [id] ASC  
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS  
= ON) ON [PRIMARY]  
) ON [PRIMARY]  
GO  
***** Object: Table [dbo].[ApneaSetting]  Script Date: 1/23/2022 11:25:59 AM  
*****  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
CREATE TABLE [dbo].[ApneaSetting](  
    [id] [bigint] IDENTITY(1,1) NOT NULL,  
    [Time] [bigint] NOT NULL,  
    [Flag_Apnea] [tinyint] NOT NULL,  
    [RateApneaSetting] [tinyint] NOT NULL,  
    [PControlApneaSetting] [tinyint] NOT NULL,  
    [TiApneasetting] [tinyint] NOT NULL,  
    [Reserved] [tinyint] NOT NULL,  
CONSTRAINT [PK_ApneaSetting] PRIMARY KEY CLUSTERED  
(  
    [id] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
= ON) ON [PRIMARY]

) ON [PRIMARY]
GO
***** Object: Table [dbo].[Parameter]  Script Date: 1/23/2022 11:25:59 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Parameter](
    [Id] [bigint] IDENTITY(1,1) NOT NULL,
    [Time] [bigint] NOT NULL,
    [Flag_Parameters_P1] [tinyint] NOT NULL,
    [Flag_Parameters_P2] [tinyint] NOT NULL,
    [TViParam_SendToHL] [real] NOT NULL,
    [RateTotalParam_SendToHL] [tinyint] NOT NULL,
    [RateSpontParam_SendToHL] [tinyint] NOT NULL,
    [LeakParam_SendToHL] [tinyint] NOT NULL,
    [CstatParam_SendToHL] [tinyint] NOT NULL,
    [RinsParam_SendToHL] [smallint] NOT NULL,
    [O2Param_SendToHL] [tinyint] NOT NULL,
    [FpeakExhParam_SendToHL] [real] NOT NULL,
    [RSBIParam_SendToHL] [smallint] NOT NULL,
    [VeParam_SendToHL] [real] NOT NULL,
    [VeSpontParam_SendToHL] [real] NOT NULL,
    [TiParam_SendToHL] [real] NOT NULL,
    [TeParam_SendToHL] [real] NOT NULL,
    [FpeakInhParam_SendToHL] [real] NOT NULL,
    [PpauseParam_SendToHL] [smallint] NOT NULL,
    [PMeanParam_SendToHL] [real] NOT NULL,
    [PEEPPParam_SendToHL] [real] NOT NULL,
    [AutoPEEPPParam_SendToHL] [smallint] NOT NULL,
```

```

[IE1Param_SendToHL] [real] NOT NULL,
[IE2Param_SendToHL] [real] NOT NULL,
[Ti_TtotParam_SendToHL] [real] NOT NULL,
[HL1Param_SendToHL] [smallint] NOT NULL,
[HL2Param_SendToHL] [smallint] NOT NULL,
[P01Param_SendToHL] [smallint] NOT NULL,
[PiMaxParam_SendToHL] [tinyint] NOT NULL,
[P01_PiMaxParam_SendToHL] [smallint] NOT NULL,
[C20_CParam_SendToHL] [smallint] NOT NULL,
[WOBimpParam_SendToHL] [smallint] NOT NULL,
[RCeParam_SendToHL] [real] NOT NULL,
[RexpParam_SendToHL] [smallint] NOT NULL,
[CdynParam_SendToHL] [tinyint] NOT NULL,
[PendInsp_SendToHL] [real] NOT NULL,
[TVeParam_SendToHL] [real] NOT NULL,
[PpeakParam_SendToHL] [real] NOT NULL,
CONSTRAINT [PK_Parameter] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
= ON) ON [PRIMARY]
) ON [PRIMARY]
GO
***** Object: Table [dbo].[Signal]  Script Date: 1/23/2022 11:25:59 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Signal](
    [id] [bigint] IDENTITY(1,1) NOT NULL,
    [Time] [bigint] NOT NULL,
    [Ventilation_Phase1] [tinyint] NULL,

```

```

[Ventilation_Phase2] [tinyint] NULL,
[Ventilation_Phase3] [tinyint] NULL,
[Ventilation_Phase4] [tinyint] NULL,
[Pressure] [varbinary](32) NOT NULL,
[Flow] [varbinary](32) NOT NULL,
[Volume] [varbinary](32) NOT NULL,
CONSTRAINT [PK_Signal] PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
= ON) ON [PRIMARY]
) ON [PRIMARY]
GO

```

ب-۲) اسکریپت مربوط به پایگاهداده O3 :

```

USE [O3_db]
GO
***** Object: Table [dbo].[Sensor1]  Script Date: 2/12/2022 9:21:54 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Sensor1](
    [Datetime] [bigint] NOT NULL,
    [Value] [tinyint] NOT NULL
) ON [PRIMARY]
GO
***** Object: Table [dbo].[Sensor2]  Script Date: 2/12/2022 9:21:54 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Sensor2](

```

```
[Datetime] [bigint] NOT NULL,  
[Value] [tinyint] NOT NULL  
) ON [PRIMARY]  
GO
```

(پ-۳) اسکریپت مربوط به پایگاهداده HeartlandAriaTransfer :

```
USE [HeartlandAriaTransferDB]  
GO  
***** Object: Table [dbo].[tbl_alarms] Script Date: 4/4/2022 1:45:04 PM *****  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
CREATE TABLE [dbo].[tbl_alarms](  
    [id] [bigint] IDENTITY(1,1) NOT NULL,  
    [date_time] [bigint] NOT NULL,  
    [code] [smallint] NOT NULL,  
    [level_alarm] [tinyint] NOT NULL,  
CONSTRAINT [PK_tbl_alarms] PRIMARY KEY CLUSTERED  
(  
    [id] ASC  
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS  
= ON) ON [PRIMARY]  
) ON [PRIMARY]  
GO  
***** Object: Table [dbo].[tbl_arr] Script Date: 4/4/2022 1:45:04 PM *****  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
CREATE TABLE [dbo].[tbl_arr](  
    [id] [bigint] IDENTITY(1,1) NOT NULL,  
    [date_time] [bigint] NOT NULL,
```

```

[hr_a] [int] NOT NULL,
[hr_b] [int] NOT NULL,
[hr_r] [int] NOT NULL,
[hr_t] [int] NOT NULL,
[hr_v] [int] NOT NULL,
[hr_value] [int] NOT NULL,
[is_alarm] [tinyint] NOT NULL,
[is_beat] [tinyint] NOT NULL,
[is_relearn] [tinyint] NOT NULL,
[pvcs_value] [int] NOT NULL,
CONSTRAINT [PK_tbl_arr] PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
= ON) ON [PRIMARY]
) ON [PRIMARY]
GO
***** Object: Table [dbo].[tbl_co2]  Script Date: 4/4/2022 1:45:04 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[tbl_co2](
    [date_time] [bigint] NOT NULL,
    [signal] [varbinary](max) NOT NULL,
    [scale_hight] [real] NOT NULL,
    [scale_low] [real] NOT NULL,
    [atmospher] [real] NOT NULL,
CONSTRAINT [PK_tbl_co2_1] PRIMARY KEY CLUSTERED
(
    [date_time] ASC

```

```

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
= ON) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
***** Object: Table [dbo].[tbl_ecg]  Script Date: 4/4/2022 1:45:04 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[tbl_ecg](
    [id] [bigint] IDENTITY(1,1) NOT NULL,
    [date_time] [bigint] NOT NULL,
    [signal_I] [varbinary](max) NOT NULL,
    [signal_II] [varbinary](max) NOT NULL,
    [signal_III] [varbinary](max) NOT NULL,
    [signal_AVR] [varbinary](max) NOT NULL,
    [signal_AVF] [varbinary](max) NOT NULL,
    [signal_AVL] [varbinary](max) NOT NULL,
    [signal_V1] [varbinary](max) NOT NULL,
    [signal_V2] [varbinary](max) NOT NULL,
    [signal_V3] [varbinary](max) NOT NULL,
    [signal_V4] [varbinary](max) NOT NULL,
    [signal_V5] [varbinary](max) NOT NULL,
    [signal_V6] [varbinary](max) NOT NULL,
    [lead_source] [tinyint] NOT NULL,
    [pace] [bit] NOT NULL,
    [gain_value] [tinyint] NOT NULL,
    [trace_ecg_filter] [tinyint] NOT NULL,
CONSTRAINT [PK_tbl_ecg] PRIMARY KEY CLUSTERED
(
    [id] ASC
)

```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
= ON) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
***** Object: Table [dbo].[tbl_ibp1]  Script Date: 4/4/2022 1:45:04 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[tbl_ibp1](
    [date_time] [bigint] NOT NULL,
    [signal] [varbinary](max) NOT NULL,
    [signal_high_limit] [real] NOT NULL,
    [signal_low_limit] [real] NOT NULL,
CONSTRAINT [PK_tbl_ibp1_1] PRIMARY KEY CLUSTERED
(
    [date_time] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
= ON) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
***** Object: Table [dbo].[tbl_ibp2]  Script Date: 4/4/2022 1:45:04 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[tbl_ibp2](
    [date_time] [bigint] NOT NULL,
    [signal] [varbinary](max) NOT NULL,
    [signal_high_limit] [real] NOT NULL,
    [signal_low_limit] [real] NOT NULL,
CONSTRAINT [PK_tbl_ibp2_1] PRIMARY KEY CLUSTERED
```

```
(  
    [date_time] ASC  
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS  
= ON) ON [PRIMARY]  
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]  
GO  
***** Object: Table [dbo].[tbl_ibp3]  Script Date: 4/4/2022 1:45:04 PM *****/  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
CREATE TABLE [dbo].[tbl_ibp3](  
    [date_time] [bigint] NOT NULL,  
    [signal] [varbinary](max) NOT NULL,  
    [signal_high_limit] [real] NOT NULL,  
    [signal_low_limit] [real] NOT NULL,  
CONSTRAINT [PK_tbl_ibp3_1] PRIMARY KEY CLUSTERED  
(  
    [date_time] ASC  
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS  
= ON) ON [PRIMARY]  
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]  
GO  
***** Object: Table [dbo].[tbl_ibp4]  Script Date: 4/4/2022 1:45:04 PM *****/  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
CREATE TABLE [dbo].[tbl_ibp4](  
    [date_time] [bigint] NOT NULL,  
    [signal] [varbinary](max) NOT NULL,  
    [signal_high_limit] [real] NOT NULL,
```

```

[signal_low_limit] [real] NOT NULL,
CONSTRAINT [PK_tbl_ibp4_1] PRIMARY KEY CLUSTERED
(
    [date_time] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
= ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
***** Object: Table [dbo].[tbl_nibp]  Script Date: 4/4/2022 1:45:04 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[tbl_nibp](
    [id] [bigint] IDENTITY(1,1) NOT NULL,
    [date_time] [bigint] NOT NULL,
    [sistol] [real] NOT NULL,
    [diastol] [real] NOT NULL,
    [map] [real] NOT NULL,
    [pulse] [int] NOT NULL,
CONSTRAINT [PK_tbl_nibp] PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
= ON) ON [PRIMARY]
) ON [PRIMARY]
GO
***** Object: Table [dbo].[tbl_resp]  Script Date: 4/4/2022 1:45:04 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

```
CREATE TABLE [dbo].[tbl_resp](  
    [date_time] [bigint] NOT NULL,  
    [signal] [varbinary](max) NOT NULL,  
CONSTRAINT [PK_tbl_resp_1] PRIMARY KEY CLUSTERED  
(  
    [date_time] ASC  
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS  
= ON) ON [PRIMARY]  
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]  
GO  
***** Object: Table [dbo].[tbl_spo2] Script Date: 4/4/2022 1:45:04 PM *****  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
CREATE TABLE [dbo].[tbl_spo2](  
    [date_time] [bigint] NOT NULL,  
    [signal] [varbinary](max) NOT NULL,  
CONSTRAINT [PK_tbl_spo2_1] PRIMARY KEY CLUSTERED  
(  
    [date_time] ASC  
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS  
= ON) ON [PRIMARY]  
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]  
GO  
***** Object: Table [dbo].[tbl_trend] Script Date: 4/4/2022 1:45:04 PM *****  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
CREATE TABLE [dbo].[tbl_trend](  
    [id] [bigint] IDENTITY(1,1) NOT NULL,
```

```
[date_time] [bigint] NOT NULL,  
[ecg_heart_rate_value] [int] NOT NULL,  
[spo2_value] [real] NOT NULL,  
[spo2_pr_value] [real] NOT NULL,  
[spo2_pi_value] [real] NOT NULL,  
[spo2_pvi_value] [real] NOT NULL,  
[resp_value] [int] NOT NULL,  
[ibp1_sistol_value] [real] NOT NULL,  
[ibp1_diastol_value] [real] NOT NULL,  
[ibp1_map_value] [real] NOT NULL,  
[ibp1_ppv_value] [tinyint] NOT NULL,  
[ibp2_sistol_value] [real] NOT NULL,  
[ibp2_diastol_value] [real] NOT NULL,  
[ibp2_map_value] [real] NOT NULL,  
[ibp2_ppv_value] [tinyint] NOT NULL,  
[ibp3_sistol_value] [real] NOT NULL,  
[ibp3_diastol_value] [real] NOT NULL,  
[ibp3_map_value] [real] NOT NULL,  
[ibp3_ppv_value] [tinyint] NOT NULL,  
[ibp4_sistol_value] [real] NOT NULL,  
[ibp4_diastol_value] [real] NOT NULL,  
[ibp4_map_value] [real] NOT NULL,  
[ibp4_ppv_value] [tinyint] NOT NULL,  
[temp_t1_value] [real] NOT NULL,  
[temp_t2_value] [real] NOT NULL,  
[temp_dt_value] [real] NOT NULL,  
[bfa_bfi_value] [int] NOT NULL,  
[bfa_bs_value] [int] NOT NULL,  
[bfa_sqi_value] [int] NOT NULL,  
[bfa_emg_value] [int] NOT NULL,  
[gas_etco2_value] [real] NOT NULL,  
[gas_fico2_value] [real] NOT NULL,  
[gas_awrr_value] [real] NOT NULL,
```

```
[st_value] [real] NOT NULL,  
[arr_pvcs_value] [int] NOT NULL,  
[gas_atmospher_value] [real] NOT NULL,  
CONSTRAINT [PK_tbl_trend] PRIMARY KEY CLUSTERED  
(  
    [id] ASC  
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS  
= ON) ON [PRIMARY]  
) ON [PRIMARY]  
GO
```

Abstract

Abstract

Today, in the modern world and in the era of technology, technology plays an important role in most industries. Medical industry is one of these industries in which technology is used a lot. In the medical industry, all kinds of devices with various technologies have been developed, which each of these devices has its own specific application; For example, a device has the task of reviving the patient by controlling the way of breathing and it does this by giving oxygen and controlling the inhalation and exhalation of the patient, or in another example, a device is obligated for reviving the patient by giving him a shock. Many of these devices must interact and send data to each other. The connection of these devices is complicated and difficult. Also, due to the fact that we are dealing with human life, sending, receiving and managing transmission data is a very important topic that has been addressed in this research in order to take an important step in the medical industry in order to revive patients as best as possible with the help of technology. In this research, the research method includes two phases of research and implementation, and we have implemented the scientific findings in the research phase using C# programming language and WPF technology in the implementation phase and evaluated its efficiency. In this regard, Viewer medical device software has been developed for monitoring and storing the patient's vital signs, which allows communication and interaction with four other medical devices, and the user (doctor, nurse, etc.) can observe and check the patient's vital signs in different parts of the program.

Key Words: Software Architecture, Data Transmission, Medical Devices, Programming, Network, Healthcare Monitoring, Patient Monitoring



SADRA INSTITUTE OF HIGHER EDUCATION

Computer Department

MSc Thesis

Title

An efficient technique for the software of Viewer medical device to monitoring and storing the patient's vital signs

**By
Karmi. S, MsC.**

**Supervisor
Askarinejad. Z, PhD.**

October 2022