



فهرست مسائل

۱	مسئله ۱
۱	بخش اول
۲	بخش دوم
۵	بخش سوم
۷	مسئله ۲
۹	مسئله ۳

پاسخ مسئله‌ی ۱.

بخش اول

الف

نیازمندی‌های غیرعملکردی (NFRs) به مجموعه‌ای از ویژگی‌ها و معیارهایی اشاره دارند که نحوه‌ی عملکرد یک سیستم را تعریف می‌کنند. این نیازمندی‌ها به جای تعیین اینکه سیستم چه کارهایی باید انجام دهد، بر روی کیفیت و نحوه‌ی انجام آن‌ها تمرکز دارند. نیازمندی‌های غیرعملکردی شامل ویژگی‌هایی مانند عملکرد، امنیت، قابلیت استفاده، و قابلیت اطمینان هستند که در موفقیت کلی یک سیستم نرم‌افزاری تأثیرگذارند.

تفاوت بین نیازمندی‌های عملکردی و غیرعملکردی:

- **نیازمندی‌های عملکردی:** این نیازمندی‌ها مشخص می‌کنند که سیستم باید چه وظایف یا عملکردهایی را انجام دهد. یعنی چه کاری باید انجام شود.
- **نیازمندی‌های غیرعملکردی:** این نیازمندی‌ها مشخص می‌کنند که سیستم چگونه باید این وظایف را انجام دهد. یعنی چطور کاری باید انجام شود.

مثال‌ها:

- **نیازمندی عملکردی:** سیستم باید به کاربران اجازه دهد که با ایمیل و رمز عبور وارد شوند.
- **نیازمندی غیرعملکردی:** سیستم باید قادر باشد که در کمتر از ۲ ثانیه وارد شود.
- **نیازمندی عملکردی:** سیستم باید به کاربران اجازه دهد تا سوابق پزشکی خود را به روز کنند.
- **نیازمندی غیرعملکردی:** سیستم باید اطمینان حاصل کند که سوابق پزشکی به طور امن و بدون از دست رفتن داده‌ها به روز می‌شود.
- **نیازمندی عملکردی:** سیستم باید به کاربران اجازه دهد پیام‌هایی به پزشکان ارسال کنند.
- **نیازمندی غیرعملکردی:** سیستم باید قابلیت اطمینان ۹۹٪ برای خدمات پیام‌رسانی را فراهم کند.

ب

نیازمندی‌های غیرعملکردی مانند عملکرد، امنیت و مقیاس‌پذیری نقشی حیاتی در موفقیت پروژه دارند زیرا به طور مستقیم بر توانایی سیستم در برآورده کردن انتظارات کاربران و بقای بلندمدت آن تأثیر می‌گذارند.

- **عملکرد:** نیازمندی‌های غیرعملکردی اطمینان می‌دهند که سیستم به زمان‌های پاسخ‌دهی و توان پردازش مناسب برسد. عملکرد ضعیف می‌تواند منجر به ناراحتی کاربران، زمان‌های تأخیر و عدم کارایی سیستم شود. برای مثال، تأخیر در بارگذاری سوابق پزشکی در یک سیستم بهداشتی می‌تواند منجر به تأخیر در مشاوره‌های پزشکی یا تجویز دارو شود.
- **امنیت:** نیازمندی‌های امنیتی برای حفاظت از داده‌های حساس در برابر دسترسی‌های غیرمجاز، حملات سایبری و نقض حریم خصوصی ضروری هستند. امنیت قوی می‌تواند از نشت داده‌ها جلوگیری کرده و اعتماد کاربران را حفظ کند.
- **مقیاس‌پذیری:** مقیاس‌پذیری برای سیستم‌هایی که انتظار دارند حجم بیشتری از کاربران یا داده‌ها را در آینده مدیریت کنند، حیاتی است. اگر سیستم نتواند به درستی مقیاس‌پذیر شود، ممکن است با افزایش تعداد کاربران یا حجم داده‌ها دچار افت عملکرد یا حتی خرابی شود. مقیاس‌پذیری این اطمینان را می‌دهد که سیستم به طور یکپارچه و بدون مشکل گسترش یابد.

بخش دوم

الف

با توجه به مشکلات گزارش شده از سوی کاربران، نیازمندی‌های غیرعملکردی زیر نقض شده‌اند:

- **نقض عملکرد:** تأخیر در بارگذاری سوابق پزشکی، به ویژه تصاویر و ویدیوها، در حین کنفرانس‌های ویدیویی نشان‌دهنده نقض عملکرد است. نیازمندی‌های عملکردی ایجاب می‌کنند که بارگذاری داده‌ها در مدت زمان معقول انجام شود. عدم رعایت این نیازمندی باعث تأخیر قابل توجهی در عملکرد سیستم می‌شود.
- **نقض قابلیت دسترسی پذیری:** نیاز به قطع و وصل مجدد اتصال زمانی که کاربران سوابق پزشکی خود را مشاهده یا به‌روزرسانی می‌کنند، نشان‌دهنده نقض در دسترسی سیستم است. نیازمندی‌های قابلیت دسترسی ایجاب می‌کنند که سیستم بدون قطع شدن یا وقفه مکرر در دسترس باشد.
- **نقض امنیت:** دسترسی تصادفی برخی از کاربران به سوابق پزشکی بیماران دیگر نقض نیازمندی‌های امنیتی است. این نیازمندی‌ها ایجاب می‌کنند که داده‌های بیمار باید به‌طور کامل از هم جدا شده و از دسترسی غیرمجاز محافظت شوند.

ب

برای ارزیابی اینکه نیازمندی‌های غیرعملکردی به درستی در سیستم پیاده‌سازی شده‌اند، می‌توان از شاخص‌های زیر استفاده کرد:

• عملکرد:

- شاخص: زمان پاسخ‌دهی برای بارگذاری سوابق و کنفرانس‌های ویدیویی.
- چگونه ارزیابی می‌کند: این شاخص زمان لازم برای بارگذاری فایل‌ها و راه‌اندازی کنفرانس ویدیویی را اندازه‌گیری می‌کند و اطمینان می‌دهد که زمان‌های تأخیر برای بارگذاری فایل‌ها از حد مجاز (مثلاً ۵ ثانیه) بیشتر نشود.

• قابلیت دسترسی:

- شاخص: درصد زمان در دسترس بودن سیستم.
- چگونه ارزیابی می‌کند: این شاخص زمان در دسترس بودن سیستم را بر اساس درصد زمانی که سیستم بدون وقفه یا خرابی کار می‌کند، اندازه‌گیری می‌کند. هدف این است که در دسترس بودن سیستم ۹۹٪ باشد.

• امنیت:

- شاخص: تعداد حوادث دسترسی غیرمجاز.
- چگونه ارزیابی می‌کند: این شاخص تعداد دفعاتی که کاربران غیرمجاز به داده‌ها دسترسی پیدا کرده‌اند را اندازه‌گیری می‌کند. باید تعداد این حوادث برابر صفر باشد تا امنیت سیستم تضمین شود.

ج

چندین عامل ممکن است باعث نقض نیازمندی‌های غیرعملکردی شوند:

- عملکرد: ممکن است سیستم برای مدیریت فایل‌های بزرگ (تصاویر و ویدیوها) بهینه‌سازی نشده باشد که باعث تأخیر در بارگذاری می‌شود. همچنین ممکن است سیستم تحت فشار تعداد زیادی از درخواست‌ها قرار گیرد و منابع به درستی تخصیص نیابند.
- قابلیت دسترسی: قطع شدن سرویس به دلیل مشکلات سخت‌افزاری، ناتوانی در مقیاس‌پذیری یا عدم پشتیبانی از زیرساخت مناسب می‌تواند باعث بروز اختلال در دسترسی سیستم شود.
- امنیت: عدم پیاده‌سازی مکانیسم‌های امنیتی قوی، مانند رمزنگاری مناسب یا تأیید هویت، می‌تواند باعث نقض امنیت و دسترسی غیرمجاز به داده‌ها شود.

د

برای بهبود نیازمندی‌های غیرعملکردی سیستم، چندین راه‌حل فنی و مدیریتی وجود دارد که می‌توانند به بهبود عملکرد، امنیت، و مقیاس‌پذیری سیستم کمک کنند. در این بخش، به شرح راه‌حل‌های فنی و مدیریتی می‌پردازیم:

۱. بهبود عملکرد

• راه‌حل فنی: استفاده از فشرده‌سازی داده‌ها

- توضیح: استفاده از الگوریتم‌های فشرده‌سازی پیشرفته مانند JPEG برای تصاویر و H.264 برای ویدیوها می‌تواند حجم داده‌ها را کاهش دهد و زمان بارگذاری را به حداقل برساند.

– تأثیر: این راه حل باعث کاهش زمان بارگذاری و انتقال داده ها و در نتیجه بهبود عملکرد سیستم خواهد شد.

• راه حل فنی: استفاده از بارگذاری همزمان

– توضیح: بارگذاری همزمان تصاویر و ویدیوها از طریق استفاده از چندین ترد می تواند سرعت بارگذاری را بهبود دهد.

– تأثیر: با این روش، زمان بارگذاری کلی کاهش یافته و کارایی سیستم بهتر خواهد شد.

۲. بهبود قابلیت دسترسی پذیری

• راه حل فنی: استفاده از سیستم های Caching

– توضیح: برای بهبود دسترسی به داده ها، می توان از سیستم های کشینگ مانند Redis یا سیستم های توزیع شده استفاده کرد که داده ها را در نزدیکی کاربر ذخیره می کنند.

– تأثیر: این روش باعث کاهش بار بر روی سرور و افزایش سرعت دسترسی به اطلاعات خواهد شد.

• راه حل مدیریتی: پیاده سازی نظارت مستمر

– توضیح: با استفاده از ابزارهای نظارت مانند Prometheus یا Grafana، عملکرد سیستم و وضعیت منابع به طور مداوم بررسی و بهینه سازی می شود.

– تأثیر: این روش باعث جلوگیری از خرابی های غیرمنتظره و بهبود دسترسی به سیستم خواهد شد.

۳. بهبود امنیت

• راه حل فنی: پیاده سازی رمزنگاری End-to-End

– توضیح: رمزنگاری End-to-End باعث می شود که تنها فرستنده و گیرنده قادر به مشاهده داده ها باشند. این روش از اطلاعات حساس کاربران در برابر حملات محافظت می کند.

– تأثیر: این راه حل امنیت داده ها را به شدت افزایش می دهد و از دسترسی غیرمجاز جلوگیری می کند.

• راه حل فنی: پیاده سازی MFA

– توضیح: با استفاده از احراز هویت چندعاملی، دسترسی به سیستم تنها با وارد کردن کد ارسال شده به گوشی همراه یا ایمیل ممکن می شود.

– تأثیر: این روش لایه امنیتی اضافی ایجاد می کند و خطر دسترسی غیرمجاز به سیستم را به حداقل می رساند.

۴. بهبود مقیاس پذیری

• راه حل فنی: استفاده از معماری میکروسرویس

– توضیح: با استفاده از معماری میکروسرویس ها، سیستم به بخش های کوچکتری تقسیم می شود که می توانند به طور مستقل مقیاس پذیر باشند. این معماری باعث می شود که سیستم به راحتی در صورت افزایش تعداد کاربران مقیاس پذیر شود.

– تأثیر: این راه حل مقیاس پذیری سیستم را بهبود می دهد و امکان گسترش سیستم بدون اختلال در بخش های دیگر فراهم می شود.

• راه حل مدیریتی: توسعه برنامه های مقیاس پذیر از ابتدا

- توضیح: تیم مدیریت باید از ابتدا مقیاس پذیری را به عنوان یک اولویت در طراحی سیستم در نظر بگیرد. این شامل انتخاب زیرساخت های مقیاس پذیر و استفاده از الگوریتم های مناسب است.
- تأثیر: این روش باعث می شود که سیستم در برابر بارهای بالا و افزایش تعداد کاربران مقاوم باشد و قابلیت گسترش آن به راحتی امکان پذیر شود.

بخش سوم

الف

در سیستم هایی مانند سیستم کنترل ترافیک هوایی که به دقت بالا و امنیت قوی نیاز دارند، همیشه تعارضاتی بین الزامات مختلف وجود دارد. برای مثال، در این سیستم، نیاز به امنیت بالا که شامل لایه های مختلف امنیتی و رمزنگاری داده ها است، باعث افزایش زمان پردازش داده ها می شود. این در حالی است که دقت و سرعت در انجام عملیات برای اطمینان از مدیریت بی وقفه و بدون تاخیر ترافیک هوایی بسیار حیاتی است.

برای مدیریت این "توافقات" باید اولویت های پروژه مشخص شود. ممکن است که برای حفظ امنیت داده ها، باید میزان پردازش بیشتری انجام گیرد که موجب تاخیر در پردازش ها شود. از طرف دیگر، این افزایش تاخیر ممکن است تأثیرات منفی بر عملکرد سیستم در محیط های پر ترافیک داشته باشد. به همین دلیل، نیاز به یک تصمیم گیری مناسب برای ایجاد تعادل بین این الزامات وجود دارد. برای مدیریت این چالش می توان از روش های زیر استفاده کرد:

- **محدودسازی پیچیدگی ها:** در ابتدای طراحی باید پیچیدگی های غیر ضروری در الگوریتم ها و لایه های امنیتی حذف شود تا از پیچیدگی های اضافی جلوگیری شود.
- **تخصیص منابع بهینه:** تخصیص مناسب منابع برای هر الزامات عملکردی و غیرعملکردی (مانند امنیت) می تواند کمک کند تا تعادل میان امنیت و سرعت پردازش برقرار شود.
- **ارزیابی و تست مداوم:** اجرای تست های مختلف برای ارزیابی اثرات متقابل بین الزامات مختلف و یافتن بهترین نقطه تعادل برای سیستم.

ب

در طول چرخه عمر پروژه، ممکن است الزامات جدیدی به سیستم افزوده شوند. برای مثال، با توجه به درخواست های جدید برای استفاده از سیستم در فرودگاه های بین المللی با ترافیک بالاتر، نیاز به مقیاس پذیری و بهبود سیستم خواهیم داشت. در این صورت، امکان بروز تغییرات زیاد در معماری سیستم، الگوریتم ها و لایه های امنیتی وجود دارد که می تواند اثرات منفی بر پایداری و عملکرد سیستم بگذارد. این افزایش الزامات به چالش هایی مانند تغییرات در ساختار داده ها، هزینه های اضافی، و لزوم اطمینان از کارکرد بی وقفه سیستم (که جزو الزامات حیاتی است) منجر می شود.

برای مدیریت این چالش می توان از روش ها و ابزارهای زیر استفاده کرد:

- **مدیریت تغییرات:** ایجاد یک فرآیند استاندارد برای پذیرش، ارزیابی و پیاده سازی تغییرات که شامل بررسی دقیق تأثیرات تغییرات جدید بر سیستم باشد.
- **استفاده از ابزارهای نظارتی و مدیریتی:** به کارگیری ابزارهای نظارتی برای پایش عملکرد سیستم و شناسایی مشکلات و نیازهای جدید.
- **مقیاس پذیری و انعطاف پذیری در طراحی:** استفاده از طراحی های مقیاس پذیر که بتوانند با افزایش بار و تغییرات مختلف سازگار شوند.

- تست و ارزیابی مداوم: اجرای تست های دوره ای برای ارزیابی قابلیت مقیاس پذیری، امنیت و پایداری سیستم در برابر تغییرات و افزایش بار ترافیکی.

پاسخ مسئله‌ی ۲.

الف

Impact-Mapping یک تکنیک برنامه‌ریزی در حوزه مهندسی نیازمندی‌ها است که به‌طور خاص برای تعیین و مستندسازی نحوه تحقق اهداف تجاری با استفاده از ویژگی‌ها و قابلیت‌های نرم‌افزاری طراحی شده است. هدف اصلی این تکنیک این است که درک روشنی از اهداف، رفتارها، و تأثیرات مورد نیاز برای دستیابی به این اهداف ایجاد کند.

یک Impact-Map معمولاً شامل چهار سطح مختلف است:

- **هدف:** هدف تجاری یا نیاز اصلی که باید محقق شود.
- **مخاطب:** افرادی یا سیستم‌هایی که بر هدف تأثیر می‌گذارند.
- **عملکردها:** تغییراتی که در نتیجه تعاملات با سیستم باید به وجود آید.
- **ویژگی‌ها:** قابلیت‌ها یا ویژگی‌هایی که باید پیاده‌سازی شوند تا این تأثیرات ایجاد شود.

مزایای Impact-Mapping عبارتند از:

- **تمرکز بر اهداف کسب‌وکار:** این روش کمک می‌کند تا تمرکز از ویژگی‌های فنی به اهداف تجاری انتقال یابد.
- **شفافیت در الزامات:** با این تکنیک می‌توان الزامات نرم‌افزاری را به‌طور واضح و بدون ابهام مستندسازی کرد.
- **تشخیص وابستگی‌ها:** Impact-Mapping به تیم‌ها کمک می‌کند تا وابستگی‌ها و تعاملات میان ویژگی‌ها و اهداف تجاری را شناسایی کنند.
- **اولویت‌بندی بهینه:** این تکنیک به اولویت‌بندی قابلیت‌ها و ویژگی‌ها بر اساس تأثیر آن‌ها بر اهداف تجاری کمک می‌کند.

ب

روش **BDD** یک روش توسعه است که بر اساس رفتار سیستم از دیدگاه کاربر یا مشتری تمرکز دارد. BDD بر روی توصیف سناریوهایی متمرکز است که نحوه تعامل سیستم با کاربران و سایر بازیگران را مشخص می‌کند. این روش با استفاده از زبان طبیعی برای نوشتن تست‌های رفتاری کاربرد دارد.

رابطه بین Impact-Map و BDD در این است که هر دو بر اساس اهداف تجاری و نیازهای کاربران تمرکز دارند. Impact-Map از طریق شناسایی اهداف و تأثیرات مختلف، به تعیین رفتارهای مورد نیاز برای دستیابی به اهداف کمک می‌کند. سپس، این رفتارها می‌توانند به سناریوهای BDD تبدیل شوند تا تست‌های رفتاری دقیق‌تری برای پیاده‌سازی ویژگی‌ها و قابلیت‌ها ایجاد کنند.

- **تطابق اهداف تجاری با سناریوهای BDD:** Impact-Map به شناسایی اهداف تجاری کمک می‌کند که می‌تواند به سناریوهای BDD تبدیل شود. این سناریوها نشان‌دهنده نحوه تعامل کاربران با سیستم برای رسیدن به این اهداف هستند.
- **ارتباط دقیق با رفتار سیستم:** Impact-Map رفتارهای مورد نیاز سیستم را بر اساس نیازهای تجاری مشخص می‌کند که با استفاده از BDD به زبان قابل فهم برای تمامی اعضای تیم مستند می‌شود.

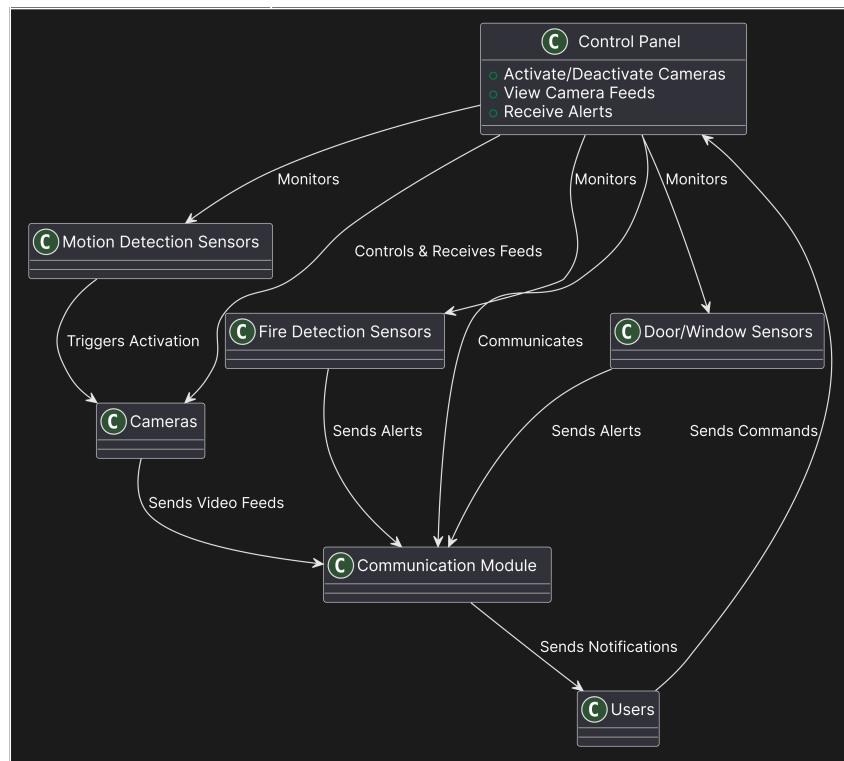
ج

برای ترکیب Impact-Map با روش BDD ، می توان مراحل زیر را دنبال کرد:

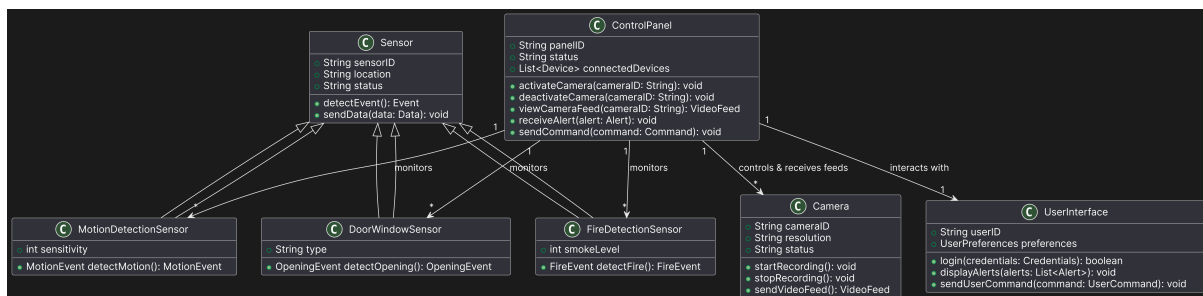
۱. **شناسایی اهداف تجاری:** ابتدا باید اهداف تجاری پروژه را مشخص کرد. این اهداف در Impact-Map تعریف می شوند.
۲. **شناسایی بازیگران و تأثیرات:** در این مرحله، بازیگران مختلف که می توانند بر اهداف تجاری تأثیر بگذارند، شناسایی می شوند و تأثیراتی که باید در نتیجه این تعاملات اتفاق بیفتد، مشخص می شود.
۳. **ترجمه تأثیرات به سناریوهای BDD:** تأثیرات شناسایی شده می توانند به سناریوهای BDD تبدیل شوند. این سناریوها باید به طور خاص رفتار سیستم را در مواجهه با هر بازیگر توصیف کنند.
۴. **اولویت بندی سناریوها:** از آنجا که Impact-Mapping به تعیین اولویت ها کمک می کند، سناریوهای BDD نیز می توانند بر اساس اولویت های تجاری تعیین شده در Impact-Map اولویت بندی شوند.
۵. **اجرای تست های رفتاری:** در نهایت، سناریوهای BDD به تست های رفتاری تبدیل می شوند که باید پیاده سازی شده و به طور مداوم آزمایش شوند.

پاسخ مسئله‌ی ۳.

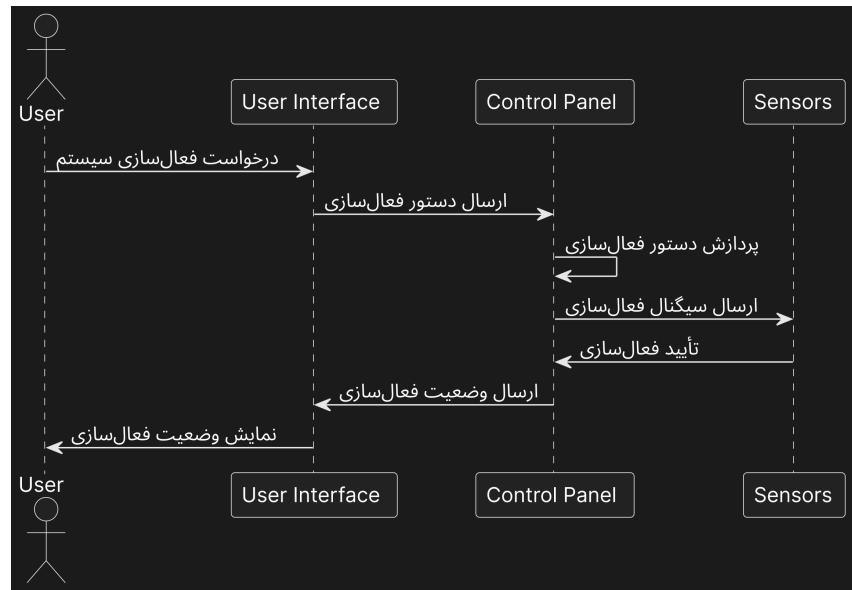
الف



ب.



ج



د

