

فاز دوم پروژه RoosterMe

طراحی UML Diagrams و معماری سیستم

درس مهندسی نرم افزار

ترم پاییز ۱۴۰۴

RoosterMe

سیستم آلرم چالشی مبتنی بر تعامل و انگیزه

اعضاي تيم

مریم رجبی ۴۰۱۴۶۳۱۳۷

مریم پورحسن ۴۰۱۴۶۳۱۲۷

دسامبر ۲۰۲۵

فهرست مطالب

	مقدمه	۱
۱	اهداف اصلی این فاز	۱.۱
۱	موارد ارائه شده در این فاز	۲.۱
۲	اهمیت فاز دوم	۳.۱
۲	UseCase Diagram	۲
۲	بازیگران سیستم (Actors)	۱.۲
۳	فهرست Use های اصلی	۲.۲
۳	توصیف کلی Case Use Diagram	۳.۲
۳	توضیح تفصیلی Use Caselar	۴.۲
۴	Case: Use ثبت نام / ورود به سیستم	۱.۴.۲
۴	Case: Use تنظیم آلام	۲.۴.۲
۴	Case: Use انتخاب چالش	۳.۴.۲
۴	Case: Use فعال شدن آلام	۴.۴.۲
۵	Case: Use انجام چالش	۵.۴.۲
۵	Case: Use مشاهده و مدیریت تاریخچه	۶.۴.۲
۵	Case: Use دریافت اعلان	۷.۴.۲
۵	Diagram Activity – فرآیند کامل اجرای آلام چالشی (RoosterMe)	۳
۵	معرفی دیاگرام فعالیت	۱.۳
۶	محدوده و نقش‌ها در دیاگرام	۲.۳
۶	توضیح کلی جریان فعالیت‌ها	۳.۳
۷	اهمیت دیاگرام فعالیت در سیستم RoosterMe	۴.۳
۷	جمع‌بندی	۵.۳
۷	System Architecture	۴
۷	توضیح معماری سیستم آلام چالشی	۱.۴
۷	سبک معماری و نمای کلی	۲.۴
۸	هدف طراحی معماری	۳.۴
۸	اجزای اصلی معماری سیستم	۴.۴
۸	اپلیکیشن موبایل Application) (Mobile	۱.۴.۴
۹	(CDN) Network Delivery Content	۲.۴.۴
۹	Balancer Load	۳.۴.۴
۹	سرورهای بک‌اند Servers) (Backend	۴.۴.۴
۹	Layer Cache	۵.۴.۴
۱۰	پایگاه داده (Database)	۶.۴.۴
۱۰	سیستم اعلان Service) (Notification	۷.۴.۴
۱۰	لایبندی معماری سیستم	۵.۴
۱۰	جریان کلی داده‌ها و فرآیندها	۶.۴
۱۱	نقاط کلیدی معماری	۱.۶.۴
۱۱	لایه‌ها و نقش‌ها	۲.۶.۴
۱۱	امنیت، پایداری و مقیاس‌پذیری	۷.۴
۱۲	جمع‌بندی معماری سیستم	۸.۴

۹.۴ تاریخچه نسخه های سند فاز دوم پروژه ۱۲

در فاز دوم پروژه RoosterMe، تمرکز اصلی بر طراحی و مدل‌سازی سطح بالا برای سیستم آنالرم چالشی است. هدف این مرحله، ارائه نمایی شفاف، دقیق و قابل درک از سیستم است تا توسعه‌دهندگان، طراحان و ذی‌نفعان بتوانند فرآیندها و تعاملات مختلف را به راحتی مشاهده و تحلیل کنند.

این فاز اهمیت بالایی دارد زیرا پایه و اساس طراحی دقیق ماذول‌ها، ارتباطات بین اجزا و معماری کلی سیستم را فراهم می‌کند و کمک می‌کند تصمیم‌گیری‌های فنی و مدیریتی بر مبنای تحلیل کامل و دقیق انجام شود.

۱.۱ اهداف اصلی این فاز

با طراحی UML و معماری سیستم، اهداف زیر دنبال می‌شوند:

- درک بهتر عملکرد سیستم: توسعه‌دهندگان و ذی‌نفعان می‌توانند جریان کارکرد سیستم و تعامل کاربران با امکانات آن را به‌وضوح مشاهده کنند. این موضوع باعث می‌شود پیچیدگی‌های سیستم از همان ابتدا شفاف و قابل مدیریت باشد.

- شفاف‌سازی تعامل بین اجزا و ماذول‌ها: ارتباطات بین اجزای مختلف سیستم مانند سرور، پایگاه داده، API‌ها و اپلیکیشن موبایل و سرویس اعلان، به صورت دیداری و قابل فهم ارائه می‌شود. این امر به کاهش ابهام و خطا در پیاده‌سازی کمک می‌کند.

- طراحی ماذول‌وار و قابل توسعه: دیاگرام‌ها و معماری ارائه شده کمک می‌کند ساختار سیستم به صورت ماذول‌وار باشد و تغییرات آینده یا افزودن قابلیت‌های جدید به راحتی انجام شود. این قابلیت برای توسعه و نگهداری طولانی‌مدت پروژه حیاتی است.

- ایجاد مبنای طراحی سطح پایین: دیاگرام‌ها و معماری سطح بالا، نقطه شروعی برای طراحی دقیق ماذول‌ها و فرآیندهای سطح پایین‌تر هستند و کمک می‌کنند تا تیم توسعه با دید کامل به پیاده‌سازی برسد.

۲.۱ موارد ارائه شده در این فاز

در این فاز، سه بخش اصلی طراحی و مدل‌سازی ارائه می‌شوند که هر کدام نقش مهمی در درک کامل سیستم دارند:

۱. Diagram: Case Use. این نمودار، رفتار سیستم و تعامل کاربران با امکانات مختلف را نمایش می‌دهد. به کمک این دیاگرام، می‌توان وظایف کاربران، نقش‌ها و عملکرد سیستم در موقعیت‌های مختلف را مشاهده کرد. این نمودار دید سطح بالا از عملکرد سیستم ارائه می‌دهد و مبنای طراحی‌های سطح پایین‌تر است.

۲. Diagram: Activity. نمودار فعالیت‌ها، جریان کارکرد اصلی سیستم را به صورت گام‌به‌گام نشان می‌دهد. از ایجاد آنالرم، انتخاب چالش، فعال شدن آنالرم، اجرای چالش تا ثبت نتایج و خاموش شدن آنالرم، تمامی مراحل با جزئیات و توالی دقیق نمایش داده می‌شوند.

۳. Architecture: System. معماری سطح بالا شامل سرور ابری، پایگاه داده، API‌ها و اپلیکیشن موبایل ارائه می‌شود. این معماری نشان می‌دهد که چگونه اجزا با یکدیگر ارتباط دارند و داده‌ها چگونه بین کاربران و سرور منتقل می‌شوند. همچنین، نقاط کلیدی برای مقیاس‌پذیری، امنیت و همگام‌سازی داده‌ها مشخص می‌شوند.

۳.۱ اهمیت فاز دوم

این فاز پایه و اساس توسعه و پیاده‌سازی سیستم RoosterMe را فراهم می‌کند. با تحلیل دقیق فرآیندها، نقش‌ها و اجزای سیستم، تیم توسعه می‌تواند:

- پیچیدگی‌ها و وابستگی‌ها را شناسایی کند،
- ریسک‌ها و چالش‌های احتمالی را پیش از پیاده‌سازی برطرف کند،
- طراحی سیستم را بهینه و منطبق بر اهداف بیزینسی و نیاز کاربران کند.

در نتیجه، فاز دوم نه تنها به ایجاد نمودارها و معما ری محدود نمی‌شود، بلکه نقش حیاتی در هدایت موفقیت‌آمیز فازهای بعدی توسعه و تضمین کیفیت سیستم دارد.

UseCase Diagram ۲

نمودار Use Case در سیستم RoosterMe با هدف نمایش تعامل کاربران و سایر اجزای خارجی با سیستم طراحی شده است. این نمودار نشان می‌دهد که هر بازیگر (Actor) چه قابلیت‌ها و وظایفی در ارتباط با سیستم دارد و سیستم چه خدماتی را در اختیار آن‌ها قرار می‌دهد.

نمایی سطح بالا از رفتار سیستم ارائه می‌دهد و تمرکز آن بر نیازمندی‌های عملکردی سیستم و تعاملات اصلی کاربران است. این نمودار مبنایی برای طراحی دقیق‌تر فرآیندها، دیاگرام‌های رفتاری و معما ری سیستم در مراحل بعدی توسعه محسوب می‌شود.

در نسخه اولیه سیستم، RoosterMe کاربر می‌تواند آلام ایجاد و مدیریت کند، برای خاموش کردن آلام چالش انجام دهد، روند پیشرفت (Streak) خود را مشاهده کند و اعلان‌های مربوط به آلام را دریافت نماید. در کنار کاربر، سیستم‌های جانبی مانند سرویس اعلان و سرور ابری نیز نقش پشتیبان در ارائه خدمات دارند.

۱.۲ بازیگران سیستم (Actors)

بازیگران اصلی سیستم RoosterMe و نقش هر یک به شرح زیر است:

- کاربر (User): بازیگر اصلی سیستم که از اپلیکیشن استفاده می‌کند. کاربر می‌تواند آلام تنظیم کند، نوع چالش را انتخاب نماید، چالش‌ها را انجام دهد، آلام را خاموش کند و تاریخچه و گزارش‌های مربوط به عملکرد خود را مشاهده نماید.
- سیستم پشتیبان (System Backend): مسئول پردازش منطق اصلی سیستم شامل مدیریت آلام‌ها، بررسی پاسخ چالش‌ها، ثبت، Streak، ذخیره داده‌ها و هماهنگی با سایر سرویس‌ها است.
- سرویس اعلان (Notification Service): وظیفه ارسال اعلان‌ها و هشدارها به کاربر را بر عهده دارد؛ از جمله اعلان نزدیک شدن به زمان آلام یا فعال شدن آن.
- سرور ابری (Cloud Server): Sync مسئول ذخیره‌سازی داده‌ها به صورت ابری و همگام‌سازی اطلاعات کاربر بین دستگاه‌ها و نسخه‌های مختلف اپلیکیشن است.
- مدیر سیستم (Admin) – در نسخه‌های آینده: این بازیگر در نسخه‌های بعدی سیستم برای نظارت بر عملکرد سرور، مدیریت داده‌ها و کنترل وضعیت کلی سیستم در نظر گرفته شده است و در نسخه فعلی نقش محدودی دارد.

۲.۲ فهرست Use های اصلی

Use های اصلی سیستم RoosterMe شامل موارد زیر هستند:

- ثبت‌نام / ورود به سیستم
- ایجاد آلام
- ویرایش آلام
- حذف آلام
- انتخاب چالش برای آلام
- فعال شدن آلام
- انجام چالش
- خاموش کردن آلام
- ثبت Streak
- مشاهده گزارش‌ها و تاریخچه
- دریافت اعلان
- همگام‌سازی داده‌ها

۳.۲ توصیف کلی Diagram Case Use

Diagram Case Use چرخه کامل تعامل کاربر با سیستم را نمایش می‌دهد. این چرخه از ثبت‌نام یا ورود کاربر آغاز شده و پس از تنظیم آلام و انتخاب چالش ادامه می‌یابد. در زمان مشخص، آلام فعال شده و کاربر برای خاموش کردن آن باید چالش تعیین شده را انجام دهد.

در صورت موفقیت در انجام چالش، آلام خاموش شده و نتیجه در سیستم ثبت می‌شود. همچنین، سیستم با ارسال اعلان‌ها و ثبت اطلاعات عملکرد کاربر، بازخورد مناسبی برای افزایش انگیزه و پایبندی به برنامه خواب فراهم می‌کند.

۴.۲ توضیح تفصیلی Use های Casel

در این بخش، Use های Casel مهم سیستم مطابق قالب استاندارد UML و به صورت متنی توضیح داده شده‌اند.

Case: Use ۱.۴.۲ ثبت نام / ورود به سیستم

کاربر Actor:

هدف: امکان دسترسی کاربر به قابلیت‌های سیستم و ذخیره اطلاعات شخصی.

پیش‌شرط: اپلیکیشن روی دستگاه کاربر نصب شده باشد.

سناریوی اصلی: کاربر اطلاعات ورود (شماره یا ایمیل) را وارد می‌کند. سیستم اطلاعات را بررسی کرده و در صورت معتبر بودن، ورود موفق انجام می‌شود.

سناریوی جایگزین: در صورت وارد کردن اطلاعات نادرست، پیام خطأ به کاربر نمایش داده می‌شود.

پس‌شرط: کاربر وارد سیستم شده و به امکانات اپلیکیشن دسترسی دارد.

Case: Use ۲.۴.۲ تنظیم آلام

کاربر Actor:

هدف: ایجاد یک آلام جدید برای بیدار شدن کاربر.

پیش‌شرط: کاربر وارد سیستم شده باشد.

سناریوی اصلی: کاربر وارد بخش تنظیم آلام می‌شود، ساعت و تکرار آلام را تعیین کرده و آن را ذخیره می‌کند.

سناریوی جایگزین: در صورت وارد کردن ساعت نامعتبر، پیام خطأ نمایش داده می‌شود.

پس‌شرط: آلام جدید در سیستم ذخیره می‌شود.

Case: Use ۳.۴.۲ انتخاب چالش

کاربر Actor:

هدف: انتخاب چالش مناسب برای خاموش کردن آلام.

پیش‌شرط: آلام قبلًا ایجاد شده باشد.

سناریوی اصلی: کاربر از لیست چالش‌ها یکی را انتخاب کرده و تنظیمات را ذخیره می‌کند.

سناریوی جایگزین: در صورت عدم انتخاب چالش، سیستم یک چالش پیش‌فرض اختصاص می‌دهد.

پس‌شرط: چالش به آلام متصل می‌شود.

Case: Use ۴.۴.۲ فعال شدن آلام

سیستم Actor:

هدف: فعال‌سازی آلام در زمان تعیین شده.

پیش‌شرط: آلام فعال وجود داشته باشد.

سناریوی اصلی: در زمان مشخص، سیستم آلام را فعال کرده و چالش مربوطه را نمایش می‌دهد.

پس‌شرط: چالش آماده اجرا بوده و آلام در حال پخش است.

Case: Use ۵.۴.۲ انجام چالش

کاربر Actor:

هدف: خاموش کردن آلام با حل موفق چالش.

پیششرط: آلام فعال شده باشد.

سناریوی اصلی: چالش به کاربر نمایش داده می‌شود. کاربر پاسخ را وارد می‌کند و سیستم صحت آن را بررسی می‌نماید.

سناریوی جایگزین: در صورت پاسخ نادرست، آلام همچنان فعال باقی می‌ماند.

پسشرط: در صورت موفقیت، آلام خاموش شده و نتیجه ثبت می‌شود.

Case: Use ۶.۴.۲ مشاهده و مدیریت تاریخچه

کاربر Actor:

هدف: مشاهده و بررسی سوابق آلام‌ها و عملکرد گذشته.

پیششرط: کاربر وارد سیستم شده باشد.

سناریوی اصلی: کاربر وارد بخش تاریخچه شده و اطلاعات مربوط به آلام‌ها را مشاهده می‌کند.

پسشرط: اطلاعات تاریخچه به صورت به روز نمایش داده می‌شود.

Case: Use ۷.۴.۲ دریافت اعلان

سرویس اعلان Actor:

هدف: اطلاع‌رسانی به کاربر در مورد وضعیت آلام.

پیششرط: وجود آلام فعال یا نزدیک شدن به زمان آلام.

سناریوی اصلی: سیستم زمان را بررسی کرده و اعلان مناسب را به کاربر ارسال می‌کند.

پسشرط: کاربر از وضعیت آلام مطلع می‌شود.

تمام دیاگرام‌ها را در پوشه Diagrams قرار داده ام.

۳ Diagram Activity – فرآیند کامل اجرای آلام چالشی (RoosterMe)

۱.۳ معرفی دیاگرام فعالیت

دیاگرام فعالیت (Activity) یکی از دیاگرام‌های رفتاری در UML است که برای نمایش جریان انجام فعالیت‌ها، ترتیب مراحل، نقاط تصمیم‌گیری و تعامل میان کاربر و سیستم مورد استفاده قرار می‌گیرد.

در پروژه RoosterMe، این دیاگرام برای نمایش فرآیند کامل اجرای یک آلام چالشی طراحی شده است. این فرآیند از لحظه‌ای که کاربر وارد برنامه می‌شود آغاز شده و تا زمان خاموش شدن موفق آلام ادامه می‌یابد. هدف اصلی این دیاگرام، ارائه نمایی شفاف از نحوه تعامل کاربر با سیستم و منطق داخلی سیستم در مدیریت و اجرای آلام چالشی است.

این دیاگرام کمک می‌کند تا رفتار سیستم در یک سناریوی واقعی به صورت مرحله‌به‌مرحله و قابل فهم نمایش داده شود.

۲.۳ محدوده و نقش‌ها در دیاگرام

در دیاگرام فعالیت RoosterMe، دو نقش اصلی در نظر گرفته شده است:

- User (کاربر): مسئول انجام فعالیت‌هایی مانند ورود به برنامه، ایجاد آلام، انتخاب چالش و پاسخ‌دهی به آن.
- System (سیستم): مسئول ذخیره‌سازی اطلاعات آلام، زمان‌بندی، فعال‌سازی آلام در زمان مقرر و بررسی صحت پاسخ چالش.

فعالیت‌ها در این دیاگرام به‌گونه‌ای طراحی شده‌اند که تعامل منطقی و پیوسته بین کاربر و سیستم را نشان دهند.

۳.۳ توضیح کلی جریان فعالیت‌ها

دیاگرام فعالیت، فرآیند اصلی عملکرد سیستم RoosterMe را به صورت یک جریان منظم و گام‌به‌گام نمایش می‌دهد. این فرآیند شامل مراحل زیر است:

۱. ورود کاربر به برنامه User (Login): فرآیند با ورود کاربر به اپلیکیشن آغاز می‌شود. کاربر باید وارد حساب کاربری خود شود تا بتواند از امکانات سیستم استفاده کند.
۲. ایجاد آلام جدید Create Alarm): پس از ورود موفق، کاربر اقدام به ایجاد یک آلام جدید کرده و زمان مورد نظر را مشخص می‌کند.
۳. انتخاب چالش Challenge): کاربر یکی از چالش‌های تعریف شده در سیستم را انتخاب می‌کند. این چالش شرط اصلی خاموش شدن آلام خواهد بود.
۴. ذخیره آلام در سیستم Save Alarm): سیستم اطلاعات مربوط به آلام، شامل زمان و نوع چالش، را ذخیره می‌کند.
۵. انتظار سیستم تا زمان آلام Alarm Until (Wait Time): سیستم وارد حالت انتظار شده و زمان تنظیم شده را پایش می‌کند.
۶. فعال‌سازی آلام Trigger Alarm): در زمان مقرر، آلام فعال شده و اعلان مربوطه برای کاربر نمایش داده می‌شود.
۷. اجرای چالش Start Challenge): پس از فعال شدن آلام، چالش انتخاب شده برای کاربر نمایش داده می‌شود و کاربر باید آن را انجام دهد.
۸. بررسی صحت پاسخ Answer): سیستم پاسخ یا عملکرد کاربر را بررسی می‌کند.
۹. تصمیم‌گیری Decision): در صورت صحیح بودن پاسخ، فرآیند به مرحله خاموش شدن آلام منتقل می‌شود. در غیر این صورت، کاربر باید چالش را مجددًا انجام دهد.

۱۰. خاموش شدن آلام (Turn Alarm Off): پس از انجام موفق چالش، آلام خاموش شده و فرآیند به پایان می‌رسد.

۴.۳ اهمیت دیاگرام فعالیت در سیستم RoosterMe

این دیاگرام نشان می‌دهد که در سیستم RoosterMe، خاموش شدن آلام تنها از طریق تعامل فعال کاربر و انجام موفق چالش امکان‌پذیر است. این موضوع باعث افزایش انگیزه کاربر برای بیدار شدن و تعامل مؤثر با سیستم می‌شود. همچنین، دیاگرام فعالیت منطقی کنترلی سیستم و نحوه مدیریت رفتار کاربر را به صورت شفاف نمایش می‌دهد و به درک بهتر عملکرد کلی سیستم کمک می‌کند.

۵.۳ جمع‌بندی

دیاگرام فعالیت ارائه شده، نمایی دقیق، منظم و قابل درک از فرآیند اجرای آلام چالشی در سیستم RoosterMe فراهم می‌کند. این دیاگرام ارتباط منطقی بین فعالیت‌های کاربر و وظایف سیستم را نمایش داده و مبنای مناسبی برای پیاده‌سازی و توسعه سیستم در فازهای بعدی پژوهش محسوب می‌شود.

تمام دیاگرام‌ها را در پوشه Diagrams قرار داده ام.

System Architecture ۴

۱.۴ توضیح معماری سیستم آلام چالشی

معماری سیستم RoosterMe نمایی جامع، سطح بالا و ساخت‌یافته از طراحی نرم‌افزار ارائه می‌دهد و نحوه تعامل اجزای مختلف سیستم را برای اجرای صحیح، پایدار و امن فرآیندهای آلام چالشی تشریح می‌کند. هدف از ارائه این معماری، ایجاد درکی شفاف از ساختار کلی سیستم، مسیر جریان داده‌ها، نقکیک مسئولیت‌ها و نحوه پیاده‌سازی اصول مهندسی نرم‌افزار در مقیاس‌پذیری، امنیت و توسعه‌پذیری است.

این معماری به عنوان مبنای اصلی طراحی و پیاده‌سازی سیستم عمل می‌کند و به توسعه‌دهندگان، طراحان و ذی‌نفعان کمک می‌کند تا پیش از ورود به جزئیات فنی، دیدی یکپارچه و دقیق از کل سیستم داشته باشند. همچنین، معماری سیستم نقش تعیین‌کننده‌ای در کیفیت نهایی نرم‌افزار، تجربه کاربری، قابلیت نگهداری بلندمدت و آمادگی سیستم برای توسعه‌های آینده ایفا می‌کند. معماری سیستم نقش حیاتی در موفقیت پژوهش دارد، زیرا پایه‌ای برای پیاده‌سازی صحیح، نگهداری بلندمدت، توسعه قابلیت‌های جدید و اطمینان از تجربه کاربری بهینه فراهم می‌کند.

۲.۴ سبک معماری و نمای کلی

معماری سیستم RoosterMe بر اساس یک Client–Server Layered Scalable CDN، with Architecture Caching and Balancer، Load Balancer، and CDN است. این سبک معماری، ترکیبی از معماری لایه‌ای و کلاینت–سرور بوده و با هدف پاسخ‌گویی به نیازهای عملکردی و غیرعملکردی سیستم انتخاب شده است.

در این معماری، اپلیکیشن موبایل به عنوان کلاینت، درخواست‌های کاربران را ارسال می‌کند و تمامی پردازش‌های اصلی در سمت سرور انجام می‌شود. برای افزایش پایداری، تحمل خطا و مقیاس‌پذیری، درخواست‌ها ابتدا وارد زیرساخت توزیع بار شده و سپس میان چندین سرور بگاند توزیع می‌شوند. وجود لایه Cache و CDN نیز باعث بُهْبود کارایی، کاهش تأخیر و کاهش بار پردازشی روی سرورهای اصلی می‌شود.

۳.۴ اهداف طراحی معماری

اهداف اصلی طراحی معماری سیستم RoosterMe عبارتند از:

- نمایش دید کلی و یکپارچه سیستم: ارائه تصویری جامع از ساختار نرم‌افزار که تمامی اجزا و ارتباطات بین آنها را به صورت شفاف نشان دهد.
- توضیح تعاملات و جریان داده‌ها: نشان دادن مسیر حرکت داده‌ها بین اپلیکیشن موبایل، سرور، پایگاه داده و سرویس اعلان.
- شفاف‌سازی مسئولیت‌ها: مشخص کردن وظایف هر بخش از سیستم و نحوه همکاری آنها برای اجرای صحیح فرآیندهای آلام چالشی.
- ایجاد مبنای طراحی و توسعه: فراهم کردن دیدی دقیق برای تیم توسعه به منظور پیاده‌سازی بهینه، مدیریت خطا، امنیت و مقیاس‌پذیری سیستم.
- اطمینان از پایداری و امنیت: نشان دادن نقاط کلیدی برای مدیریت خطا، جلوگیری از از دست رفتن داده‌ها و حفاظت از حریم خصوصی کاربران.
- پشتیبانی از توسعه آینده و افزودن قابلیت‌های جدید: طراحی معماری به گونه‌ای که امکان افزودن ویژگی‌های جدید یا تغییر فرآیندها بدون بازسازی کل سیستم فراهم باشد.

۴.۴ اجزای اصلی معماری سیستم

۱.۴.۴ اپلیکیشن موبایل (Mobile Application)

اپلیکیشن موبایل رابط اصلی تعامل کاربر با سیستم RoosterMe است و تمامی عملیات مرتبط با آلام‌ها و چالش‌ها از طریق آن انجام می‌شود. این لایه مسئول نمایش اطلاعات، دریافت ورودی کاربر و ارسال درخواست‌ها به زیرساخت سمت سرور است.

وظایف اصلی این بخش شامل موارد زیر است:

- ایجاد، ویرایش و حذف آلام‌های چالشی
- انتخاب و اجرای چالش‌های مرتبط با آلام
- نمایش وضعیت Streak و تاریخچه فعالیت‌ها
- دریافت اعلان‌ها و هشدارهای سیستم

• ارتباط امن با سرور از طریق API

(CDN) Network Delivery Content ۲.۴.۴

CDN به منظور بهبدود سرعت دسترسی کاربران به محتوای ایستا و کاهش تأخیر شبکه در معما ری سیستم در نظر گرفته شده است. محتوایی مانند تصاویر رابط کاربری، فایل‌های صوتی آلام و منابع ثابت از طریق CDN به کاربر تحویل داده می‌شود.

این مؤلفه باعث کاهش بار روی سرورهای بکاند شده و تجربه کاربری روانتری را به ویژه برای کاربران با موقعیت جغرافیایی متفاوت فراهم می‌کند.

Balancer Load ۳.۴.۴

نقطه ورود تمامی درخواست‌های کاربران به زیرساخت سمت سرور است. این مؤلفه درخواست‌های دریافتی از اپلیکیشن موبایل را به صورت هوشمند میان چندین سرور بکاند توزیع می‌کند. وجود Balancer Load باعث:

- جلوگیری از ایجاد گلوگاه پردازشی
- افزایش دسترسی‌پذیری سیستم
- افزایش تحمل خطأ در صورت از کار افتادن یک سرور
- امکان مقیاس‌پذیری افقی بدون تغییر در کلاینت

Servers) (Backend ۴.۴.۴

سرورهای بکاند هسته منطق کسب و کار سیستم RoosterMe را تشکیل می‌دهند. تمامی پردازش‌های مربوط به آلام‌ها، چالش‌ها، احراز هویت، مدیریت Streak و هماهنگی با سرویس اعلان در این لایه انجام می‌شود. در این معما ری، چندین نمونه از سرور بکاند به صورت هم‌زمان فعال هستند که همگی منطق یکسانی دارند و توسط Balancer Load مدیریت می‌شوند. این طراحی باعث افزایش پایداری و مقیاس‌پذیری سیستم می‌شود.

Layer Cache ۵.۴.۴

لایه Cache بین سرورهای بکاند و پایگاه داده قرار دارد و داده‌هایی که به طور مکرر مورد استفاده قرار می‌گیرند را به صورت موقت ذخیره می‌کند. این لایه باعث کاهش تعداد دسترسی مستقیم به پایگاه داده و افزایش سرعت پاسخ‌دهی سیستم می‌شود.

استفاده از Cache نقش مهمی در بهینه‌سازی عملکرد سیستم، کاهش زمان پاسخ و افزایش ظرفیت پردازش هم‌زمان کاربران دارد.

۶.۴.۴ پایگاه داده (Database)

پایگاه داده مسئول ذخیره‌سازی دائمی اطلاعات سیستم است. این اطلاعات شامل داده‌های کاربران، آلارم‌ها، چالش‌ها، نتایج فعالیت‌ها و تاریخچه Streak می‌باشد.
پایگاه داده به‌گونه‌ای طراحی شده است که:

- از یکپارچگی داده‌ها محافظت کند
- امنیت اطلاعات کاربران را تضمین نماید
- قابلیت مقیاس‌پذیری و نگهداری بلندمدت داشته باشد

۷.۴.۴ سیستم اعلان (Notification Service)

سیستم اعلان وظیفه ارسال هشدارها و یادآوری‌های مرتبط با آلارم‌ها و چالش‌ها را بر عهده دارد. این سرویس با سرورهای بک‌اند هماهنگ بوده و اعلان‌ها را در زمان مناسب به کاربر ارسال می‌کند.
این مؤلفه نقش کلیدی در موفقیت تجربه کاربری سیستم آلارم چالشی دارد.

۵.۴ لایه‌بندی معماری سیستم

معماری RoosterMe به صورت لایه‌ای طراحی شده است:

- لایه رابط کاربری (UI Layer)
- لایه تحويل محتوا (CDN Layer)
- لایه توزیع بار (Load Balancing Layer)
- لایه منطق کسب‌وکار (Business Logic Layer)
- لایه Cache
- لایه داده (Data Layer)
- لایه سرویس‌های کمکی (Notification Services)

این لایه‌بندی باعث شفافیت، کاهش وابستگی و سهولت توسعه سیستم می‌شود.

۶.۴ حریان کلی داده‌ها و فرآیندها

حریان اجرای عملیات در سیستم RoosterMe به صورت زیر است:

۱. ارسال درخواست از اپلیکیشن موبایل
۲. دریافت درخواست توسط Balancer Load

۳. هدایت درخواست به یکی از سرورهای بکاند

۴. بررسی داده‌ها در Cache

۵. بازیابی داده از پایگاه داده در صورت نیاز

۶. ارسال اعلان در زمان فعال شدن آلام

۷. ارائه پاسخ نهایی به کاربر

۱.۶.۴ نقاط کلیدی معماری

- هماهنگی بین اپلیکیشن و سرویس اعلان: ارسال اعلان‌ها باید دقیق و به موقع باشد.
- همگام‌سازی داده‌ها: (Sync) تغییرات آلام و نتایج چالش باید بلافصله بین اپلیکیشن و سرور همگام شوند.
- امنیت و حریم خصوصی: داده‌ها رمزگذاری شده و دسترسی غیرمجاز محدود می‌شود.
- مقیاس‌پذیری: سیستم باید قابلیت افزایش تعداد کاربران و سرویس‌ها بدون افت عملکرد داشته باشد.
- پایداری و مدیریت خطأ: خطاهای کنترل می‌شوند و از از دست رفتن داده‌ها جلوگیری می‌شود.
- قابلیت نگهداری و توسعه آینده: افزودن قابلیت‌های جدید یا تغییر فرآیندها بدون بازسازی کل سیستم ممکن باشد.

۲.۶.۴ لایه‌ها و نقش‌ها

- لایه رابط کاربری (UI): اپلیکیشن موبایل که تعامل مستقیم با کاربر را مدیریت می‌کند.
- لایه منطق کسب‌وکار (Business Logic): سرور ابری که پردازش درخواست‌ها، مدیریت آلام‌ها و چالش‌ها، مدیریت Streak و گزارش‌ها و هماهنگی با سرویس اعلان را انجام می‌دهد.
- لایه داده (Data Layer): پایگاه داده که مسئول ذخیره‌سازی، بازیابی و مدیریت اطلاعات سیستم است.
- لایه سرویس‌های کمکی (Auxiliary Services): شامل سیستم اعلان و دیگر سرویس‌های پشتیبان که عملکرد سیستم را یکپارچه می‌کنند.

۷.۴ امنیت، پایداری و مقیاس‌پذیری

معماری سیستم RoosterMe به‌گونه‌ای طراحی شده است که امنیت داده‌ها، پایداری سیستم و مقیاس‌پذیری در اولویت قرار دارند. استفاده از تفکیک لایه‌ها، توزیع بار، Cache و سرویس‌های مستقل، ریسک خرابی سیستم را کاهش داده و امکان توسعه آینده را فراهم می‌کند.

۸.۴ جمع‌بندی معماری سیستم

معماری سیستم RoosterMe یک ساختار مدرن، مقیاس‌پذیر و مهندسی شده را ارائه می‌دهد که تمامی نیازهای یک سیستم آلام چالشی را پوشش می‌دهد. این معماری علاوه بر پاسخ‌گویی به نیازهای فعلی، بستری مناسب برای رشد، توسعه و بهبود مستمر سیستم در آینده فراهم می‌سازد.

تمام دیاگرام‌ها را در پوشه Diagrams قرار داده‌ام.

۹.۴ تاریخچه نسخه‌های سند فاز دوم پروژه

نسخه	تاریخ	نویسنده	شرح تغییرات
۱	۱۴۰۴/۰۹/۰۳	مریم رجبی	تحلیل اولیه پروژه و تعریف دامنه سیستم آلام چالشی RoosterMe؛ شناسایی بازیگران اصلی، اهداف سیستم و نیازمندی‌های سطح بالا
۲	۱۴۰۴/۰۹/۰۶	مریم رجبی	طراحی و مستندسازی Diagram Case Use شامل شناسایی بازیگران، تعریف Use Case های اصلی، روابط include و extend و تشریح سناریوهای اصلی و جایگزین
۳	۱۴۰۴/۰۹/۱۱	مریم رجبی	طراحی و تکمیل Diagram Activity برای فرآیند کامل اجرای آلام چالشی، شامل جریان ورود کاربر، ایجاد آلام، انتخاب چالش، فعال‌سازی آلام، تصمیم‌گیری و خاموش شدن آلام
۴	۱۴۰۴/۰۹/۱۵	مریم رجبی	طراحی معماری اولیه سیستم شامل اپلیکیشن موبایل، سرور بکاند، پایگاه داده و سیستم اعلان و تشریح تعاملات و جریان داده‌ها
۵	۱۴۰۴/۰۹/۱۷	مریم رجبی	ارتقای معماری سیستم با افزودن Load Balancer و Cache به منظور افزایش مقیاس‌پذیری، بهبود عملکرد، پایداری و آمادگی برای توسعه در مقیاس بالا
۶	۱۴۰۴/۰۹/۲۰	مریم رجبی	تکمیل مستندات معماری سیستم شامل توضیح لایه‌ها، مسئولیت اجزا، ملاحظات امنیتی، مدیریت خطای و پشتیبانی از توسعه آینده

جدول ۱ : سوابق تغییرات، طراحی و توسعه سیستم RoosterMe