

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

۱	۱ مقدمه	
۱	۱.۱ اهداف اصلی این فاز	۱.۱
۱	۲.۱ موارد ارائه شده در این فاز	۲.۱
۲	۳.۱ اهمیت فاز دوم	۳.۱
UseCase Diagram		
۲	۱.۲ بازیگران سیستم (Actors)	۱.۲
۳	۲.۲ فهرست UseCase های اصلی	۲.۲
۳	۳.۲ توصیف کلی UseCase Diagram	۳.۲
۴	۴.۲ توضیح تفصیلی UseCase ها	۴.۲
۴	۱.۴.۲ ثبت نام / ورود به سیستم	۱.۴.۲
۴	۲.۴.۲ تنظیم آلام	۲.۴.۲
۴	۳.۴.۲ انتخاب چالش	۳.۴.۲
۴	۴.۴.۲ فعال شدن آلام	۴.۴.۲
۵	۵.۴.۲ انجام چالش	۵.۴.۲
۵	۶.۴.۲ مشاهده و مدیریت تاریخچه	۶.۴.۲
۵	۷.۴.۲ دریافت اعلان	۷.۴.۲
Activity Diagram		
۷	۷ فرآیند کامل اجرای آلام چالشی (RoosterMe)	
۷	۱.۳ معرفی دیاگرام فعالیت	۱.۳
۷	۲.۳ محدوده و نقش‌ها در دیاگرام	۲.۳
۷	۳.۳ توضیح کلی جریان فعالیت‌ها	۳.۳
۸	۴.۳ اهمیت دیاگرام فعالیت در سیستم RoosterMe	۴.۳
Sequence Diagram		
۱۰	۱.۴ Challenge-Based Alarm Dismissal	۱.۴
۱۱	۲.۴ User Registration & Authentication Flow	۲.۴
۱۳	۳.۴ ثبت موفقیت کاربر	۳.۴
۱۳	۴.۴ ارسال اعلان و یادآوری	۴.۴
Data Flow Diagram		
Class Diagram		
System Architecture		
۱۶	۱.۷ توضیح معماری سیستم آلام چالشی	۱.۷
۱۶	۲.۷ سبک معماری و نمای کلی	۲.۷
۱۷	۳.۷ اهداف طراحی معماری	۳.۷
۱۷	۴.۷ اجزای اصلی معماری سیستم	۴.۷
۱۷	۱.۴.۷ اپلیکیشن موبایل (Mobile Application)	۱.۴.۷
۱۸	۲.۴.۷ Content Delivery Network (CDN)	۲.۴.۷
۱۸	۳.۴.۷ Load Balancer	۳.۴.۷
۱۸	۴.۴.۷ سرورهای بک‌اند (Backend Servers)	۴.۴.۷
۱۹	۵.۴.۷ Cache Layer	۵.۴.۷

۱۹	پایگاه داده (Database)	۶.۴.۷
۱۹	سیستم اعلان (Notification Service)	۷.۴.۷
۱۹	لایه‌بندی معماری سیستم	۵.۷
۲۰	جزیان کلی داده‌ها و فرآیندها	۶.۷
۲۰	نقاط کلیدی معماری	۱.۶.۷
۲۰	لایه‌ها و نقش‌ها	۲.۶.۷
۲۱	امنیت، پایداری و مقیاس‌پذیری	۷.۷
۲۱	جمع‌بندی معماری سیستم	۸.۷
۲۳	تاریخچه نسخه‌های سند فاز دوم پروژه	۸

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 مازول‌ها، 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 مازول‌ها: ارتباطات بین اجزای مختلف سیستم مانند سرور، پایگاه داده، API‌ها، 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 مازولار و قابل توسعه: دیاگرام‌ها و معماری ارائه شده کمک می‌کند ساختار سیستم به صورت مازولار 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 مازول‌ها 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. **Use Case Diagram:** 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. **Activity Diagram:** 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. **System Architecture:** معماری سطح بالا شامل سرور ابری، پایگاه داده، API‌ها و اپلیکیشن موبایل ارائه می‌شود. این معماری نشان می‌دهد که چگونه اجزا با یکی‌گر 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 معما‌ری محدود نمی‌شود، بلکه نقش حیاتی در هدایت موفقیت‌آمیز فازهای بعدی توسعه و تضمین کیفیت سیستم دارد.

UseCase Diagram ۲

نمودار UseCase 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.

UseCase Diagram 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 معما‌ری 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 (Backend System) : 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 Sync Server) : 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 UseCase های اصلی

UseCase 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 UseCase Diagram

UseCase Diagram سیستم RoosterMe 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 UseCase ها

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

۱.۴.۲ UseCase : ثبت‌نام / ورود به سیستم

Actor : کاربر

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.

۲.۴.۲ UseCase : تنظیم آلام

Actor : کاربر

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.

۳.۴.۲ UseCase : انتخاب چالش

Actor : کاربر

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.

۴.۴.۲ UseCase : فعال شدن آلام

Actor : سیستم

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.

۵.۴.۲ UseCase : انجام چالش

Actor : کاربر

هدف: خاموش کردن آلام با حل موفق چالش.

پیششرط: آلام فعال شده باشد.

سناریوی اصلی: چالش به کاربر نمایش داده می شود. کاربر پاسخ را وارد می کند و سیستم صحت آن را بررسی می نماید.

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

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

۶.۴.۲ UseCase : مشاهده و مدیریت تاریخچه

Actor : کاربر

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 می کند.

پسشرط: اطلاعات تاریخچه به صورت به روز نمایش داده می شود.

۷.۴.۲ UseCase : دریافت اعلان

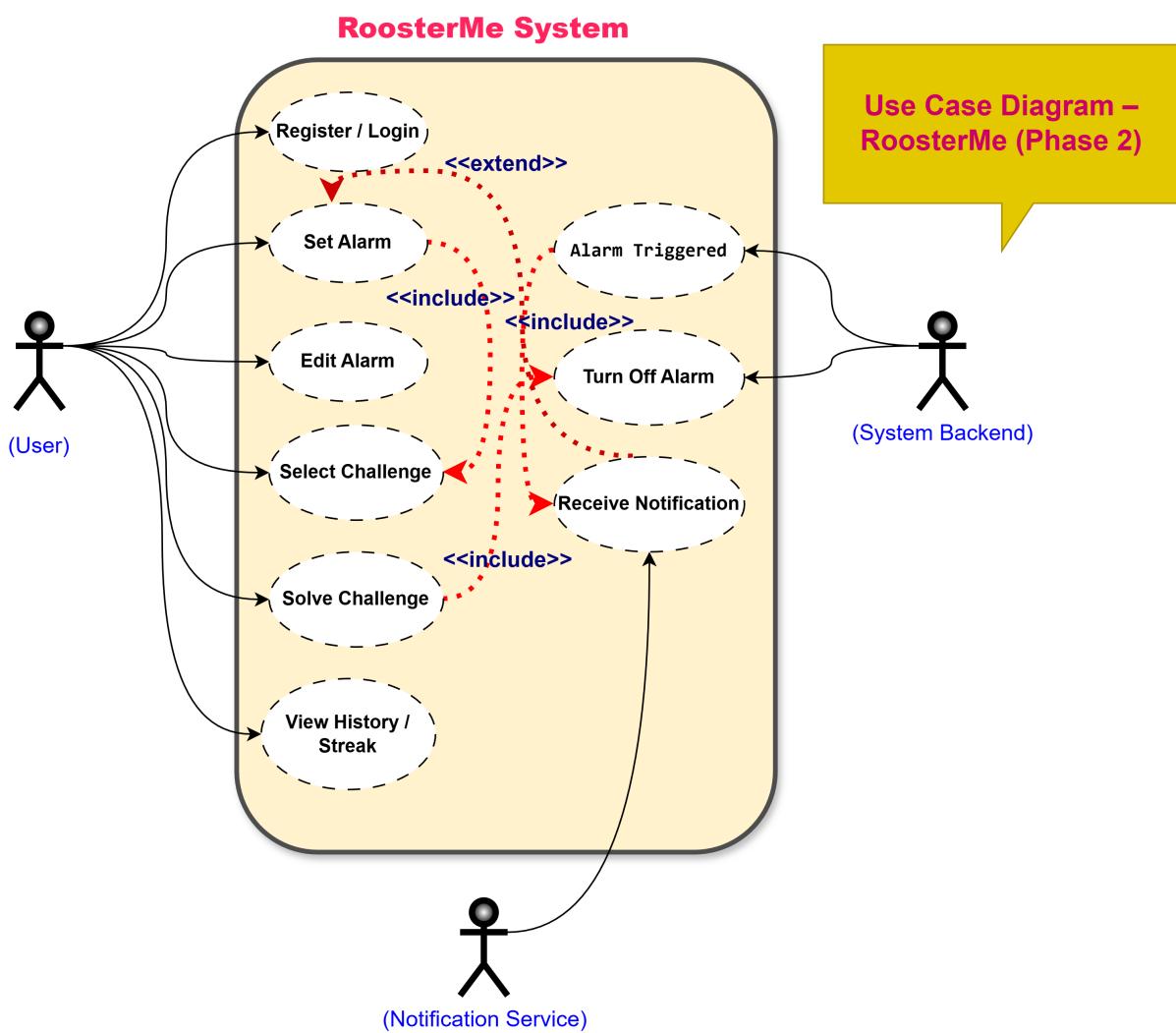
Actor : سرویس اعلان

هدف: اطلاع رسانی 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 می کند.

پسشرط: کاربر از وضعیت آلام مطلع می شود.



شكل ١ : RoosterMe UseCase Diagram

تمام دیاگرام‌ها در پوشه Diagrams آورده شده است.

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

۱.۳ معرفی دیاگرام فعالیت

دیاگرام فعالیت (Activity Diagram) 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.
- RoosterMe System: 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

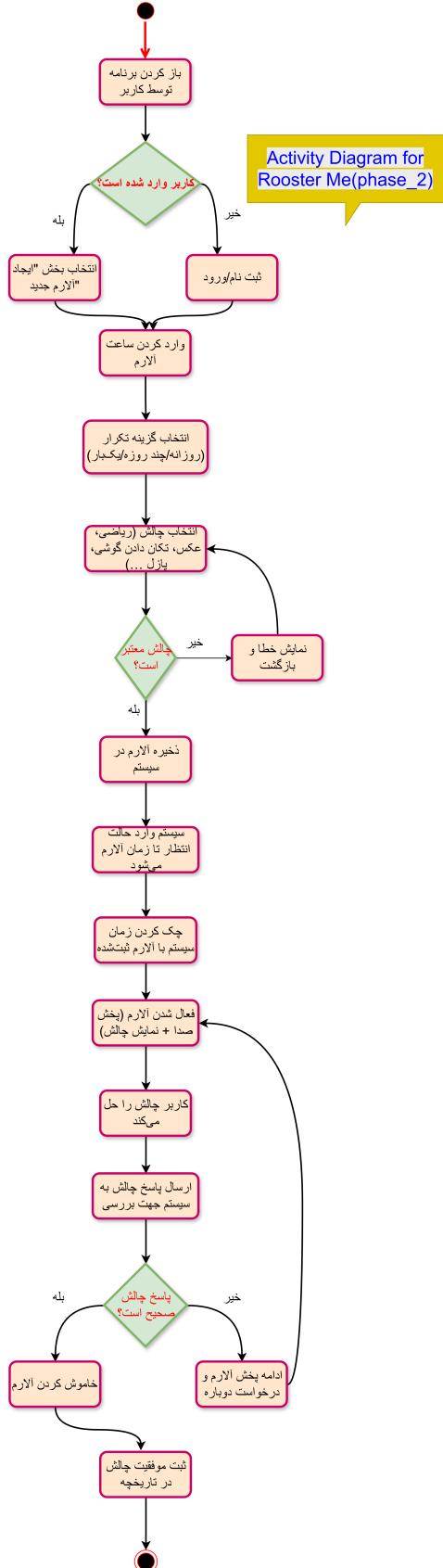
دیاگرام فعالیت، فرآیند اصلی عملکرد سیستم RoosterMe را به صورت یک جریان منظم و گام‌به‌گام نمایش می‌دهد. این فرآیند شامل مراحل زیر است:

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 (Select 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 آلرم (Wait Until Alarm Time) : سیستم وارد حالت انتظار شده و زمان تنظیم شده را پایش می کند.
۶. فعال سازی آلرم (Trigger Alarm) : در زمان مقرر، آلرم فعال شده و اعلان مربوطه برای کاربر نمایش داده می شود.
۷. اجرای چالش (Start Challenge) : پس از فعال شدن آلرم، چالش انتخاب شده برای کاربر نمایش داده می شود و کاربر باید آن را انجام دهد.
۸. بررسی صحت پاسخ (Validate Answer) : سیستم پاسخ یا عملکرد کاربر را بررسی می کند.
۹. تصمیم گیری (Decision) : در صورت صحیح بودن پاسخ، فرآیند به مرحله خاموش شدن آلرم منتقل می شود. در غیر این صورت، کاربر باید چالش را مجدداً انجام دهد.
۱۰. خاموش شدن آلرم (Turn Off Alarm) : پس از انجام موفق چالش، آلرم خاموش شده و فرآیند به پایان می رسد.

۴.۳ اهمیت دیاگرام فعالیت در سیستم RoosterMe

این دیاگرام نشان می دهد که در سیستم RoosterMe، خاموش شدن آلرم تنها از طریق تعامل فعال کاربر و انجام موفق چالش امکان پذیر است. این موضوع باعث افزایش انگیزه کاربر برای بیدار شدن و تعامل مؤثر با سیستم می شود. همچنین، دیاگرام فعالیت منطق کنترلی سیستم و نحوه مدیریت رفتار کاربر را به صورت شفاف نمایش می دهد و به درک بهتر عملکرد کلی سیستم کمک می کند.



شكل ۲: RoosterMe سیستم Activity Diagram

تمام دیاگرام‌ها در پوشه Diagrams آورده شده است.

Sequence Diagram ۴

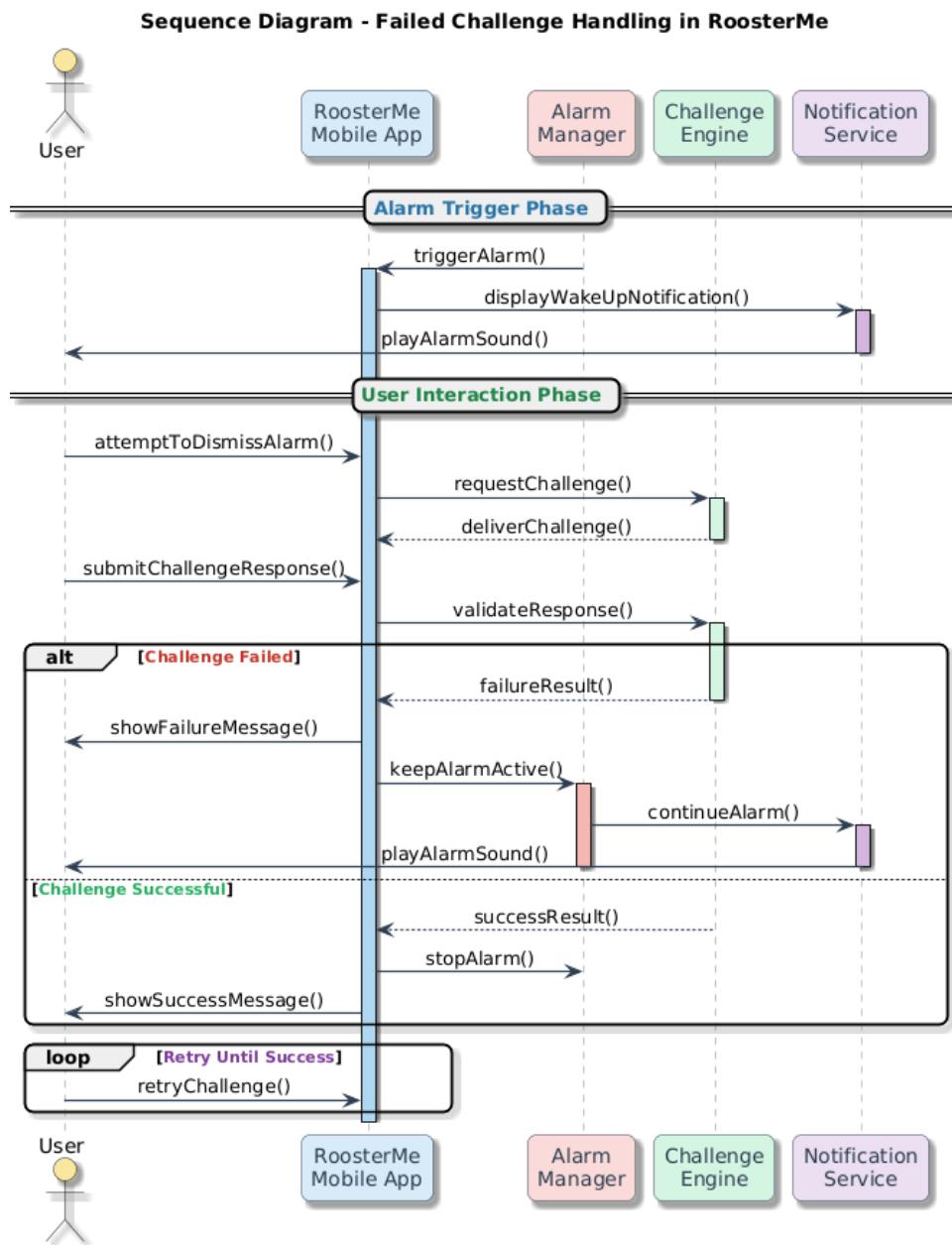
در این بخش، توالی تعامل بین اجزای مختلف سیستم RoosterMe در سناریوهای اصلی نمایش داده می‌شود.

Challenge-Based Alarm Dismissal ۱.۴

این نمودار توالی، فرآیند خاموش‌سازی آلام مبتنی بر چالش در سامانه RoosterMe را نمایش می‌دهد. در این سناریو، پس از فعال‌سازی آلام، کاربر تنها در صورتی قادر به غیرفعال کردن آن خواهد بود که چالش ارائه شده توسط سیستم را با موفقیت پشت سر بگذارد. هدف اصلی این مکانیزم، افزایش میزان بیدار شدن واقعی کاربر و جلوگیری از خاموش کردن ساده آلام است. در ابتدای فرآیند، سرویس مدیریت آلام، رویداد فعال‌سازی آلام را به اپلیکیشن ارسال می‌کند. سپس

اپلیکیشن از طریق سرویس اعلان و صدا، اعلان بیداریابش و صدای آلام را برای کاربر پخش می‌نماید. با تلاش کاربر برای خاموش کردن آلام، اپلیکیشن درخواست تولید چالش را به موتور ارزیابی چالش ارسال کرده و چالش مناسب به کاربر نمایش داده می‌شود. پس از ارسال پاسخ چالش توسط کاربر، موتور ارزیابی نتیجه را بررسی می‌کند. در صورتی که

پاسخ ناموفق باشد، پیام خطابه کاربر نمایش داده شده و وضعیت آلام همچنان فعال باقی می‌ماند و صدای آلام ادامه پیدا می‌کند. این فرآیند تا زمانی که کاربر بتواند چالش را با موفقیت حل کند، در قالب یک حلقه تکرار می‌شود. در صورت موفقیت کاربر در حل چالش، نتیجه تأیید شده و فرمان غیرفعال‌سازی آلام صادر می‌شود. در نهایت، پیام موفقیت به کاربر نمایش داده شده و فرآیند بیداریابش به پایان می‌رسد. این نمودار، تعامل میان کاربر و اجزای اصلی سیستم شامل اپلیکیشن موبایل، سرویس مدیریت آلام، موتور ارزیابی چالش و سرویس اعلان را به صورت شفاف نشان می‌دهد.

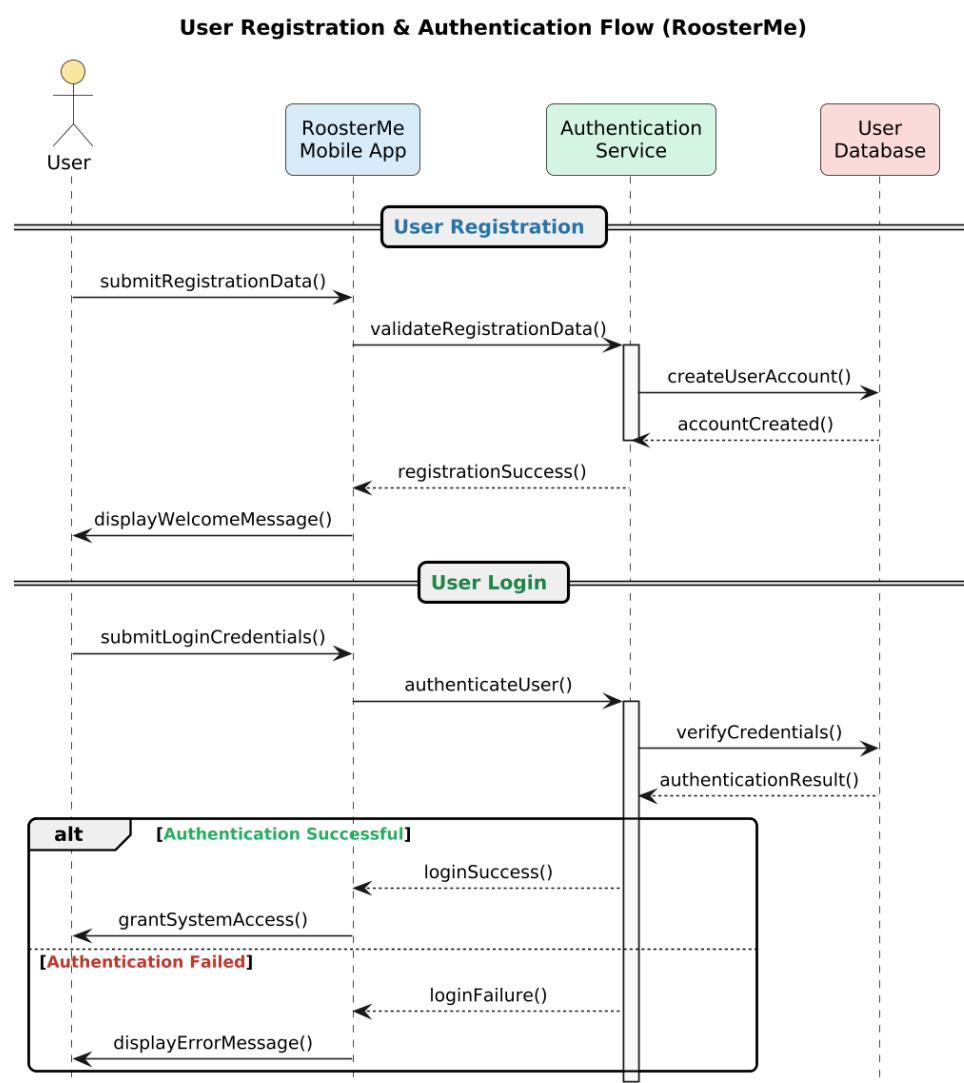


شكل ۳: Challenge-Based Alarm Dismissal in RoosterMe

User Registration & Authentication Flow ۲.۴

این نمودار توالی، فرآیند ثبت‌نام و احراز هویت کاربران در سامانه RoosterMe را نمایش می‌دهد. هدف این فرآیند، ایجاد دسترسی امن و کنترل شده برای کاربران و مدیریت صحیح اطلاعات هویتی آنها در سیستم است. در بخش ثبت‌نام، کاربر اطلاعات اولیه خود را از طریق اپلیکیشن موبایل ارسال می‌کند. اپلیکیشن این داده‌ها را برای بررسی صحت و اعتبار به سرویس احراز هویت ارسال می‌نماید. پس از اعتبارسنجی اطلاعات، سرویس احراز هویت درخواست ایجاد حساب کاربری جدید را به پایگاه داده کاربران ارسال می‌کند. با ایجاد موفق حساب کاربری، نتیجه به اپلیکیشن بازگردانده شده و

پیام خوشآمدگویی به کاربر نمایش داده می‌شود. در بخش ورود به سیستم، کاربر اطلاعات ورود خود را به اپلیکیشن ارسال می‌کند. اپلیکیشن درخواست احراز هویت را به سرویس مربوطه منتقل کرده و این سرویس با بررسی اطلاعات در پایگاه داده کاربران، نتیجه احراز هویت را مشخص می‌نماید. در صورت موفقیت آمیز بودن فرآیند احراز هویت، دسترسی کاربر به قابلیت‌های سیستم فراهم می‌شود. در غیر این صورت، پیام خطای مناسب به کاربر نمایش داده شده و از ورود غیرمجاز جلوگیری می‌شود. این نمودار تعامل بین اجزای اصلی شامل کاربر، اپلیکیشن موبایل، سرویس احراز هویت و پایگاه داده کاربران را به صورت شفاف نمایش داده و نقش هر مؤلفه را در تأمین امنیت و مدیریت دسترسی سیستم مشخص می‌کند.



شكل ۴: User Registration and Authentication Flow in RoosterMe

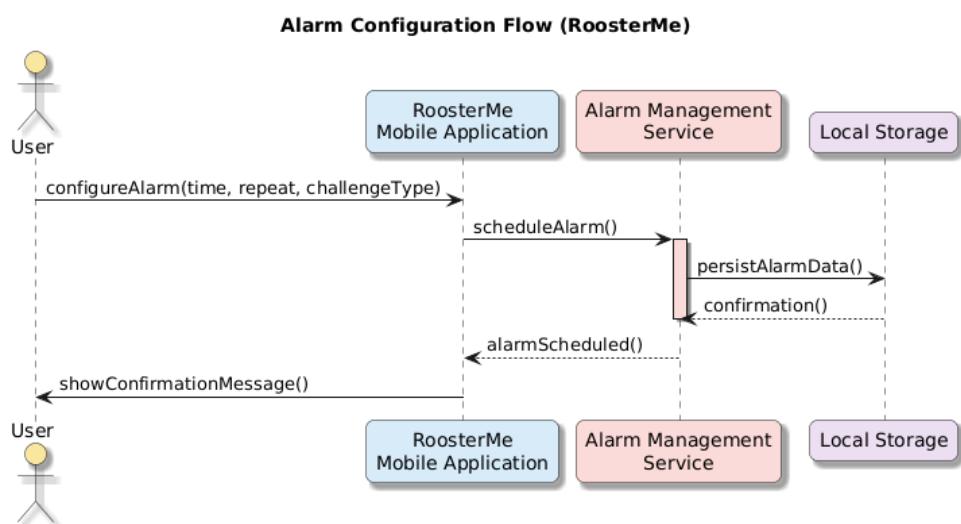
۳.۴ ثبت موفقیت کاربر

این نمودار توالی، فرآیند پیکربندی و ثبت موفق آلام توسط کاربر در سامانه RoosterMe را نمایش می‌دهد. هدف این فرآیند، فراهم کردن امکان تنظیم آلام شخصی‌سازی شده و ذخیره‌سازی پایدار اطلاعات آن برای استفاده در زمان فعال‌سازی آلام است. در این سناریو، کاربر از طریق اپلیکیشن موبایل، مشخصات آلام شامل زمان بیداری‌باش، الگوی تکرار و نوع

چالش موردنظر را تنظیم می‌کند. اپلیکیشن پس از دریافت این اطلاعات، درخواست زمان‌بندی آلام را به سرویس مدیریت آلام ارسال می‌نماید. سرویس مدیریت آلام مسئول برنامه‌ریزی دقیق آلام و هماهنگی با زیرساخت سیستم برای فعال‌سازی آن در زمان مقرر است. پس از زمان‌بندی موفق آلام، اطلاعات مربوط به آلام در فضای ذخیره‌سازی

محلی سیستم ثبت می‌شود تا پایداری داده‌ها و بازیابی آن‌ها در صورت بسته شدن یا راهاندازی مجدد اپلیکیشن تضمین شود. پس از ذخیره موفق اطلاعات، تأییدیه به اپلیکیشن ارسال شده و پیام موفقیت‌آمیز بودن عملیات به کاربر نمایش داده می‌شود. این نمودار تعامل میان کاربر، اپلیکیشن موبایل، سرویس مدیریت آلام و فضای ذخیره‌سازی محلی را به صورت

شفاف نمایش می‌دهد و نشان می‌دهد که چگونه تنظیمات کاربر به صورت ایمن و پایدار در سیستم ثبت می‌شوند.



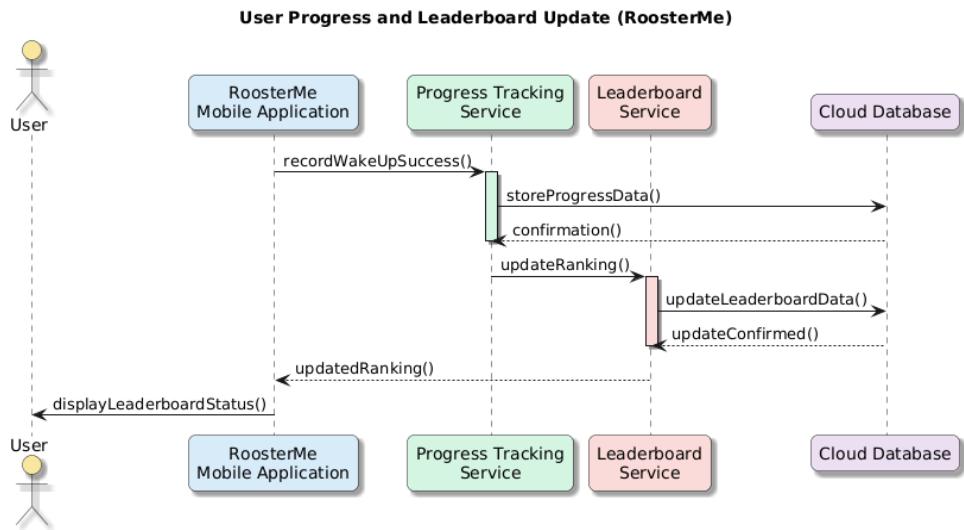
شكل ۵: Alarm Configuration and Successful Registration Flow in RoosterMe

۴.۴ ارسال اعلان و یادآوری

این نمودار توالی، فرآیند ارسال اعلان‌ها، ثبت پیشرفت کاربر و بروزرسانی رتبه‌بندی در سامانه RoosterMe را نمایش می‌دهد. هدف اصلی این فرآیند، اطلاع‌رسانی به موقع به کاربر، ثبت موفقیت‌های روزانه و ایجاد انگیزه از طریق نمایش پیشرفت و جایگاه کاربر در سیستم است.

در این سناریو، پس از ثبت موفق بیدار شدن کاربر، اپلیکیشن موبایل رویداد موفقیت را به سرویس ثبت پیشرفت ارسال می‌کند. این سرویس مسئول ذخیره‌سازی داده‌های مربوط به عملکرد کاربر در پایگاه داده ابری است تا اطلاعات پیشرفت به صورت پایدار و قابل بازیابی نگهداری شوند.

پس از ذخیره موفق داده‌ها، سرویس ثبت پیشرفت درخواست به روزرسانی رتبه‌بندی را به سرویس Leaderboard ارسال می‌کند. سرویس رتبه‌بندی با محاسبه مجدد جایگاه کاربر نسبت به سایر کاربران، اطلاعات جدید را در پایگاه داده ابری ثبت می‌نماید. پس از تکمیل این فرآیند، رتبه به روز شده به اپلیکیشن بازگردانده می‌شود. در نهایت، اپلیکیشن وضعیت جدید کاربر را شامل میزان پیشرفت و جایگاه او در جدول رتبه‌بندی به کاربر نمایش می‌دهد. این مکانیزم نقش مهمی در افزایش انگیزه، تعامل و مشارکت مستمر کاربران ایفا می‌کند و مستقیماً با اهداف بیزینسی سیستم در حوزه‌ی Engagement و Retention هم‌راستا است.



شکل ۶: User Progress Tracking and Leaderboard Update in RoosterMe

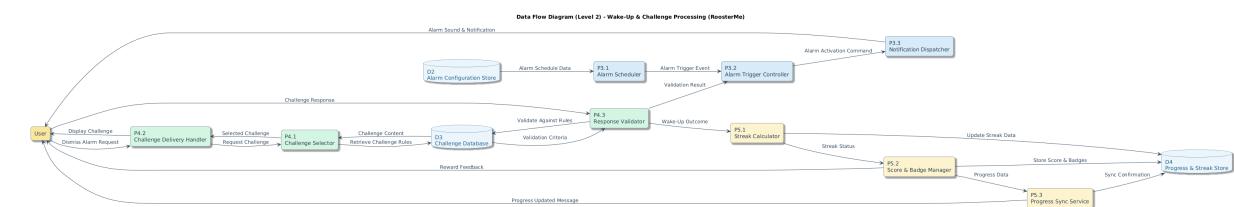
Data Flow Diagram ۵

در این بخش، نمودار جریان داده (Data Flow Diagram) سامانه RoosterMe ارائه شده است که نحوه تبادل داده میان کاربر، پردازش‌های اصلی سیستم و مخازن داده را نمایش می‌دهد. این نمودار با تمرکز بر فرآیند بیداری‌اش، اجرای چالش و ثبت پیشرفت کاربر طراحی شده و دید جامعی از جریان داده‌ها در سطح عملیاتی سیستم فراهم می‌کند. در این نمودار، کاربر به عنوان موجودیت خارجی اصلی، با سیستم تعامل دارد. فرآیند از مرحله زمان‌بندی آلام آغاز

می‌شود؛ جایی که داده‌های مربوط به تنظیمات آلام از مخزن پیکربندی آلام دریافت شده و توسط ماژول زمان‌بندی آلام پردازش می‌گردد. پس از فعال شدن آلام، رویداد بیداری‌اش به ماژول کنترل آلام ارسال شده و از طریق سرویس اعلان، صدای آلام و پیام یادآوری به کاربر نمایش داده می‌شود.

پس از درخواست خاموش کردن آلام توسط کاربر، سیستم وارد مرحله اجرای چالش می‌شود. در این مرحله، چالش مناسب از پایگاه داده چالش‌ها انتخاب شده و از طریق ماژول تحويل چالش به کاربر نمایش داده می‌شود. پاسخ کاربر به چالش، به ماژول اعتبارسنجی ارسال شده و با استفاده از قوانین و معیارهای موجود در پایگاه داده چالش‌ها بررسی می‌گردد. نتیجه این اعتبارسنجی، وضعیت نهایی بیداری‌اش کاربر را مشخص می‌کند.

در صورت موفقیت کاربر، داده‌های مربوط به نتیجه بیدارباش به ماژول محاسبه Streak ارسال می‌شود. سپس اطلاعات امتیاز، نشان‌ها و وضعیت پیشرفت کاربر توسط ماژول مدیریت امتیاز و نشان پردازش شده و در مخزن پیشرفت و ذخیره می‌گردد. در نهایت، سرویس همگام‌سازی پیشرفت وظیفه ثبت نهایی و همگام‌سازی داده‌ها را بر عهده داشته و پیام بهروزرسانی پیشرفت به کاربر ارسال می‌شود. این نمودار جریان داده نشان می‌دهد که چگونه داده‌ها به صورت ساختاریافته و مرحله‌ای در سیستم RoosterMe جریان پیدا می‌کنند و تضمین می‌کند که فرآیندهای بیدارباش، چالش، اعتبارسنجی و ثبت پیشرفت به صورت منسجم، قابل ردیابی و قابل توسعه پیاده‌سازی شده‌اند.

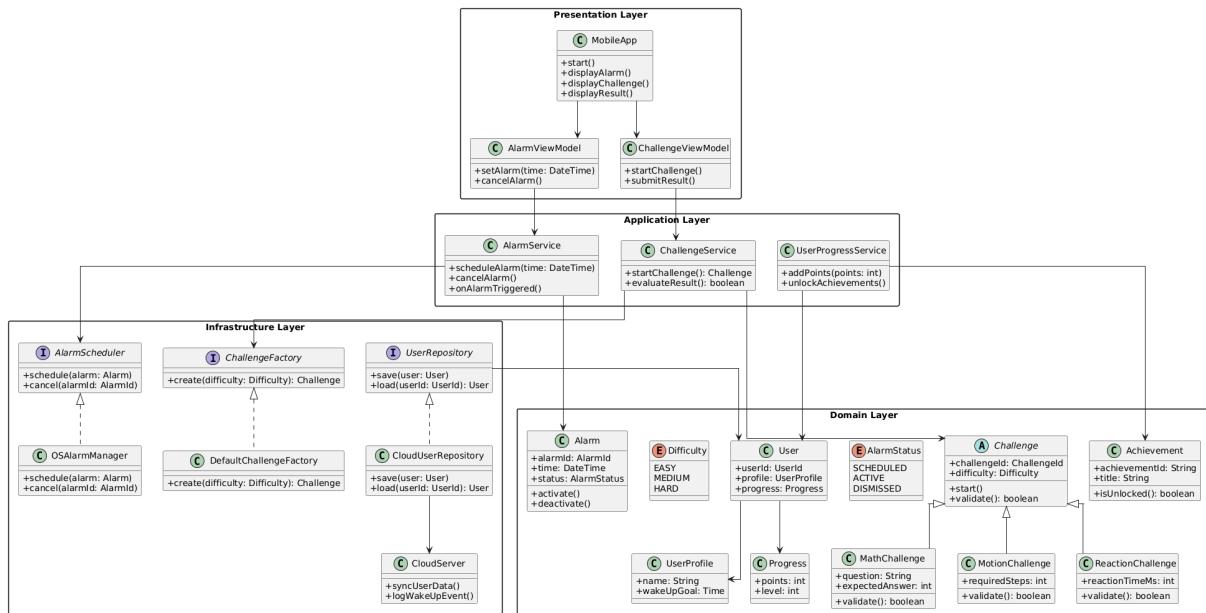


شکل ۷: Data Flow Diagram (Level 2) – Wake-Up and Challenge Processing in RoosterMe

Class Diagram ۶

این بخش ساختار ایستا و کلاس‌های اصلی سیستم RoosterMe را نمایش می‌دهد. این نمودار، ساختار ایستای سیستم RoosterMe را نمایش می‌دهد و شامل کلاس‌های اصلی، ویژگی‌ها (Attributes) و رفتارها (Methods) هر کلاس است.

در این نمودار، کلاس User نمایانگر کاربر نهایی سیستم بوده و اطلاعات مربوط به حساب کاربری و تنظیمات شخصی وی را نگهداری می‌کند. کلاس Alarm مسئول مدیریت آلام‌ها شامل زمان‌بندی، وضعیت فعال یا غیرفعال بودن و ارتباط با چالش‌های مربوطه است. کلاس Challenge منطق مربوط به چالش‌های بیدارباش را پیاده‌سازی می‌کند و بررسی موفقیت یا شکست کاربر در انجام چالش را بر عهده دارد. همچنین کلاس Alarm Manager به عنوان هسته مرکزی، هماهنگی بین آلام‌ها، چالش‌ها و اعلان‌ها را انجام می‌دهد. در نهایت، کلاس Server یا CloudService مسئول ذخیره‌سازی اطلاعات، ثبت عملکرد کاربران و همگام‌سازی داده‌ها با سرور مرکزی است. روابط بین کلاس‌ها به گونه‌ای طراحی شده‌اند که اصل تقسیم مسئولیت‌ها (Separation of Concerns) رعایت شده و توسعه‌پذیری سیستم در نسخه‌های آینده تضمین شود.



شکل ۸: Class Diagram سیستم RoosterMe

تمام دیاگرامها در پوشه **Diagrams** آورده شده است.

System Architecture ۷

۱.۷ توضیح معماری سیستم آلام چالشی

معماری سیستم **RoosterMe** نمایی جامع، سطح بالا و ساختیافته از طراحی نرم افزار 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، تفکیک مسئولیت‌ها و نحوه پیاده‌سازی اصول مهندسی نرم افزار در مقیاس پذیری، امنیت و توسعه پذیری است.

این معماری به عنوان مبنای اصلی طراحی و پیاده‌سازی سیستم عمل می‌کند و به توسعه‌دهنگان، 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 نرم افزار، تجربه کاربری، قابلیت نگهداری بلندمدت و آمادگی سیستم برای توسعه‌های آینده ایفا می‌کند. معماری سیستم نقش حیاتی در موفقیت پروژه دارد، زیرا پایه‌ای برای پیاده‌سازی صحیح، نگهداری بلندمدت، توسعه قابلیت‌های جدید و اطمینان از تجربه کاربری بهینه فراهم می‌کند.

۲.۷ سبک معماری و نمای کلی

معماری سیستم **RoosterMe** بر اساس یک Scalable Layered Client–Server Architecture with CDN, Load Balancer, and Caching طراحی شده است. این سبک معماری، ترکیبی از معماری لایه‌ای و کلاینت–서ور 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 بکاند توزیع می‌شوند. وجود لایه Cache و CDN نیز باعث بهبود کارایی، کاهش تأخیر و کاهش بار پردازشی روی سرورهای اصلی می‌شود.

۳.۷ 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.

۴.۷ اجزاء اصلی معماری سیستم

۱.۴.۷ 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 آلارم
- نمایش وضعیت Streak و تاریخچه فعالیت‌ها
- دریافت اعلان‌ها و هشدارهای سیستم
- ارتباط امن با سرور از طریق API

Content Delivery Network (CDN) ۴.۴.۷

CDN بهمنظور بهبود 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.

Load Balancer ۴.۴.۷

Load Balancer 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 Load Balancer 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

سرورهای بکاند (Backend Servers) ۴.۴.۷

سرورهای بکاند هسته منطق کسب‌وکار 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 Load Balancer مدیریت می‌شوند. این طراحی باعث افزایش پایداری و مقیاس‌پذیری سیستم می‌شود.

Cache Layer ۵.۴.۷

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.

۶.۴.۷ 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

۷.۴.۷ 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.

۵.۷ لایه‌بندی معماری سیستم

معماری RoosterMe به صورت لایه‌ای طراحی شده است:

- لایه رابط کاربری (UI Layer)
- لایه تحويل محتوا (CDN Layer)
- لایه توزيع 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.

۶.۷ جریان کلی دادهها و فرآیندها

جریان اجرای عملیات در سیستم RoosterMe به صورت زیر است:

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

۱.۶.۷ 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 بلافارسله 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 دست رفتن داده‌ها جلوگیری می‌شود.
- قابلیت نگهداری و توسعه آینده: افزودن قابلیت‌های جدید یا تغییر فرآیندها بدون بازسازی کل سیستم ممکن باشد.

۲.۶.۷ لایه‌ها و نقش‌ها

- لایه رابط کاربری (UI Layer) : اپلیکیشن موبایل که تعامل مستقیم با کاربر را مدیریت می‌کند.
- لایه منطق کسب‌وکار (Business Logic Layer) : 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.

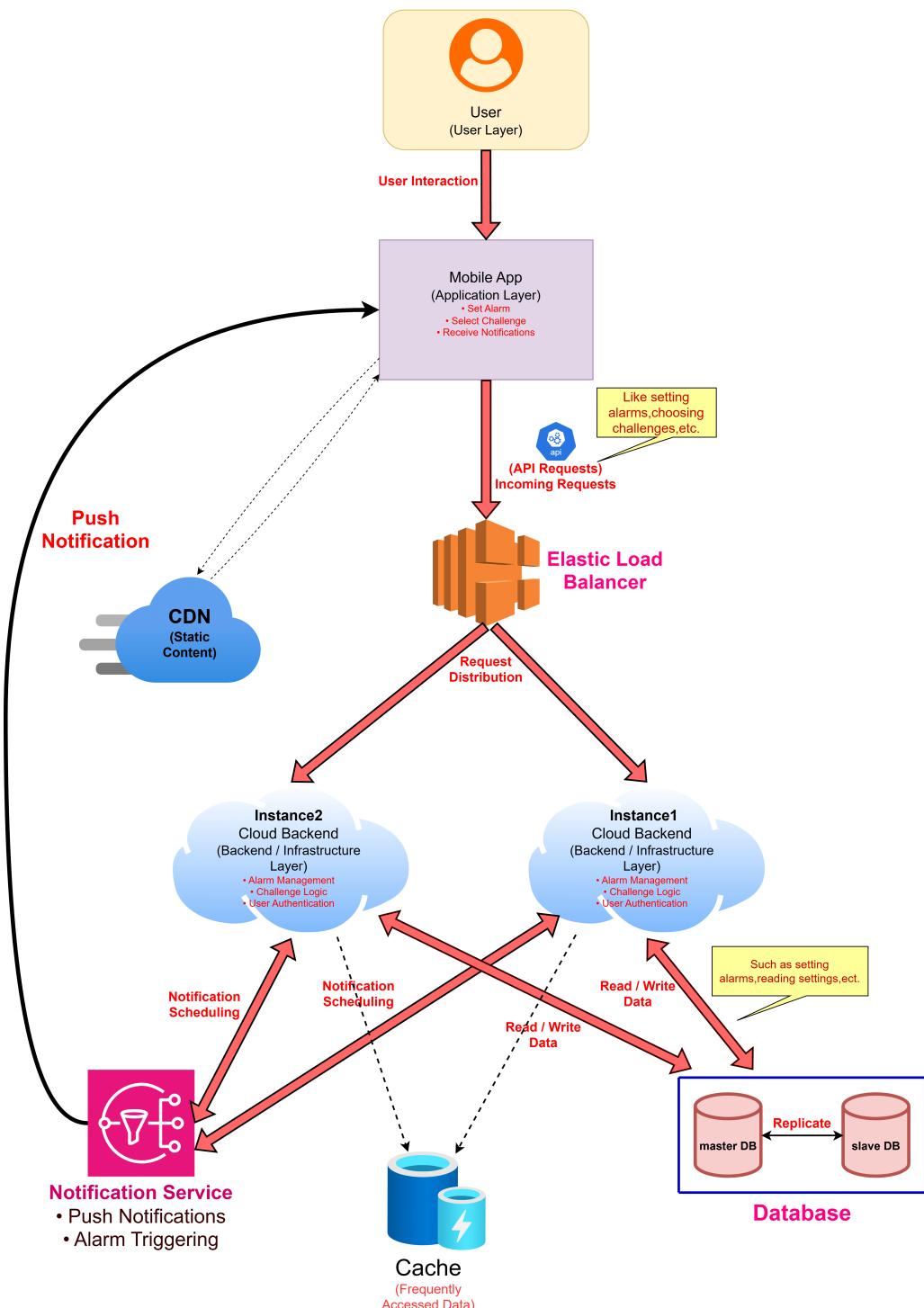
۷.۷ امنیت، پایداری و مقیاس‌پذیری

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.

۸.۷ 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.

Scalable Layered Client-Server Architecture with caching and Service-Oriented Components with Load Balancer Support (RoosterMe)



شكل ٩ System Architecture Diagram : سیستم RoosterMe

تمام دیاگرام‌ها در پوشه Diagrams آورده شده است.

۸ 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؛ شناسایی بازیگران اصلی، اهداف سیستم و نیازمندی‌های سطح بالا
۲	۱۴۰۴/۰۹/۰۶	مریم رجبی	طرابی و مستندسازی UseCase Diagram شامل شناسایی بازیگران، تعریف‌های UseCase اصلی، روابط extend و include و تشریح سناریوهای اصلی و جایگزین
۳	۱۴۰۴/۰۹/۱۱	مریم رجبی	طرابی و تکمیل Activity Diagram 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	طرابی 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، CDN 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
۷	۱۴۰۴/۰۹/۲۲	مریم پورحسن	طرابی و مستندسازی Sequence Diagram‌های سیستم شامل فرآیند خاموش‌سازی آلام مبتنی بر چالش، ثبت‌نام و احرار هويت کاربر، پیکربندی و ثبت آلام، و ارسال اعلان و بهروزرسانی پیشرفت کاربر
۸	۱۴۰۴/۰۹/۲۳	مریم پورحسن	طرابی و تشریح Data Flow Diagram (DFD – Level 2) با تمرکز بر جریان داده‌ها در فرآیند بیداری‌اش، اجرای چالش، اعتبارسنجی پاسخ، محاسبه Streak و همگام‌سازی پیشرفت کاربر
۹	۱۴۰۴/۰۹/۲۴	مریم پورحسن	طرابی و مستندسازی Class Diagram سیستم User , Alarm کلاس‌های اصلی RoosterMe CloudSer- , Challenge , Alarm Manager vice و تشریح روابط و مسئولیت‌های هر کلاس

جدول ۱: سوابق تغییرات، طرابی و توسعه سیستم RoosterMe