

# فاز دوم پروژه RoosterMe

طراحی UML Diagrams و معماری سیستم

درس مهندسی نرم افزار

ترم پاییز ۱۴۰۴

---

## RoosterMe

سیستم آلارم چالشی مبتنی بر تعامل و انگیزه

---

اعضای تیم

مریم رجبی ۴۰۱۴۶۳۱۳۷

مریم پورحسن ۴۰۱۴۶۳۱۲۷

دسامبر ۲۰۲۵

## فهرست مطالب

۱	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	<b>UseCase Diagram</b>	۲
۲	بازیگران سیستم (Actors)	۱.۲
۳	فهرست UseCase های اصلی	۲.۲
۳	توصیف کلی UseCase Diagram	۳.۲
۴	توضیح تفصیلی UseCase ها	۴.۲
۴	UseCase : ثبت نام / ورود به سیستم	۱.۴.۲
۴	UseCase : تنظیم آلارم	۲.۴.۲
۴	UseCase : انتخاب چالش	۳.۴.۲
۴	UseCase : فعال شدن آلارم	۴.۴.۲
۵	UseCase : انجام چالش	۵.۴.۲
۵	UseCase : مشاهده و مدیریت تاریخچه	۶.۴.۲
۵	UseCase : دریافت اعلان	۷.۴.۲
۷	<b>Activity Diagram – فرآیند کامل اجرای آلارم چالشی (RoosterMe)</b>	۳
۷	معرفی دیاگرام فعالیت	۱.۳
۷	محدوده و نقش ها در دیاگرام	۲.۳
۷	توضیح کلی جریان فعالیت ها	۳.۳
۸	اهمیت دیاگرام فعالیت در سیستم RoosterMe	۴.۳
۱۰	<b>Sequence Diagram</b>	۴
۱۰	Challenge-Based Alarm Dismissal	۱.۴
۱۱	User Registration & Authentication Flow	۲.۴
۱۳	ثبت موفقیت کاربر	۳.۴
۱۳	ارسال اعلان و یادآوری	۴.۴
۱۴	<b>Data Flow Diagram</b>	۵
۱۵	<b>Class Diagram</b>	۶
۱۶	<b>System Architecture</b>	۷
۱۶	توضیح معماری سیستم آلارم چالشی	۱.۷
۱۷	سبک معماری و نمای کلی	۲.۷
۱۷	اهداف طراحی معماری	۳.۷
۱۷	اجزای اصلی معماری سیستم	۴.۷
۱۷	اپلیکیشن موبایل (Mobile Application)	۱.۴.۷
۱۸	Content Delivery Network (CDN)	۲.۴.۷
۱۸	Load Balancer	۳.۴.۷
۱۸	سرورهای بک اند (Backend Servers)	۴.۴.۷
۱۹	Cache Layer	۵.۴.۷

۱۹	..... پایگاه داده (Database)	۶.۴.۷
۱۹	..... سیستم اعلان (Notification Service)	۷.۴.۷
۱۹	..... لایه بندی معماری سیستم	۵.۷
۲۰	..... جریان کلی داده ها و فرآیندها	۶.۷
۲۰	..... نقاط کلیدی معماری	۱.۶.۷
۲۰	..... لایه ها و نقش ها	۲.۶.۷
۲۱	..... امنیت، پایداری و مقیاس پذیری	۷.۷
۲۱	..... جمع بندی معماری سیستم	۸.۷
۲۳	تاریخچه نسخه های سند فاز دوم پروژه	۸

## ۱ مقدمه

در فاز دوم پروژه RoosterMe، تمرکز اصلی بر طراحی و مدل سازی سطح بالا برای سیستم آلام چالشی است. هدف این مرحله، ارائه نمایی شفاف، دقیق و قابل درک از سیستم است تا توسعه دهندگان، طراحان و ذی نفعان بتوانند فرآیندها و تعاملات مختلف را به راحتی مشاهده و تحلیل کنند. این فاز اهمیت بالایی دارد زیرا پایه و اساس طراحی دقیق ماژول ها، ارتباطات بین اجزا و معماری کلی سیستم را فراهم می کند و کمک می کند تصمیم گیری های 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 می شوند:

- درک بهتر عملکرد سیستم: توسعه دهندگان و ذی نفعان می توانند جریان کارکرد سیستم و تعامل کاربران با امکانات آن را به وضوح مشاهده کنند. این موضوع باعث می شود پیچیدگی های سیستم از همان ابتدا شفاف و قابل مدیریت باشد.
- شفاف سازی تعامل بین اجزا و ماژول ها: ارتباطات بین اجزای مختلف سیستم مانند سرور، پایگاه داده، API ها، اپلیکیشن موبایل و سرویس اعلان، به صورت دیداری و قابل فهم ارائه می شود. این امر به کاهش ابهام و خطا در پیاده سازی کمک می کند.
- طراحی ماژولار و قابل توسعه: دیاگرام ها و معماری ارائه شده کمک می کند ساختار سیستم به صورت ماژولار باشد و تغییرات آینده یا افزودن قابلیت های جدید به راحتی انجام شود. این قابلیت برای توسعه و نگهداری طولانی مدت پروژه حیاتی است.
- ایجاد مبنای طراحی سطح پایین: دیاگرام ها و معماری سطح بالا، نقطه شروعی برای طراحی دقیق ماژول ها و فرآیندهای سطح پایین تر هستند و کمک می کنند تا تیم توسعه با دید کامل به پیاده سازی برسد.

### ۲.۱ موارد ارائه شده در این فاز

در این فاز، سه بخش اصلی طراحی و مدل سازی ارائه می شوند 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 می دهد. به کمک این دیاگرام، می توان وظایف کاربران، نقش ها و عملکرد سیستم در موقعیت های مختلف را مشاهده کرد. این نمودار دید سطح بالا از عملکرد سیستم ارائه می دهد و مبنای طراحی های سطح پایین تر است.
۲. **Activity Diagram:** نمودار فعالیت ها، جریان کارکرد اصلی سیستم را به صورت گام به گام نشان می دهد. از ایجاد آلام، انتخاب چالش، فعال شدن آلام، اجرای چالش تا ثبت نتایج و خاموش شدن آلام، تمامی مراحل با جزئیات و توالی دقیق نمایش داده می شوند.
۳. **System Architecture:** معماری سطح بالا شامل سرور ابری، پایگاه داده، API ها و اپلیکیشن موبایل ارائه می شود. این معماری نشان می دهد که چگونه اجزا با یکدیگر ارتباط دارند و داده ها چگونه بین کاربران و سرور منتقل می شوند. همچنین، نقاط کلیدی برای مقیاس پذیری، امنیت و همگام سازی داده ها مشخص می شوند.

## 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.

## ۲ UseCase Diagram

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 آلارم ایجاد و مدیریت کند، برای خاموش کردن آلارم چالش انجام دهد، روند پیشرفت (Streak) خود را مشاهده کند و اعلان‌های مربوط به آلارم 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 آلارم تنظیم کند، نوع چالش را انتخاب نماید، چالش‌ها را انجام دهد، آلارم 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 آلارم‌ها، 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 آلارم 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:

- ثبت نام / ورود به سیستم
- ایجاد آلارم
- ویرایش آلارم
- حذف آلارم
- انتخاب چالش برای آلارم
- فعال شدن آلارم
- انجام چالش
- خاموش کردن آلارم
- ثبت Streak
- مشاهده گزارش ها و تاریخچه
- دریافت اعلان
- همگام سازی داده ها

## ۳.۲ توصیف کلی UseCase Diagram

UseCase Diagram سیستم RoosterMe چرخه کامل تعامل کاربر با سیستم را نمایش می دهد. این چرخه از ثبت نام یا ورود کاربر آغاز شده و پس از تنظیم آلارم و انتخاب چالش ادامه می یابد. در زمان مشخص، آلارم فعال شده و کاربر برای خاموش کردن آن باید چالش تعیین شده را انجام دهد.

در صورت موفقیت در انجام چالش، آلارم خاموش شده و نتیجه در سیستم ثبت می شود. همچنین، سیستم با ارسال اعلان ها و ثبت اطلاعات عملکرد کاربر، بازخورد مناسبی برای افزایش انگیزه و پایبندی به برنامه خواب فراهم می کند.

## 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 : ثبت نام / ورود به سیستم

#### Actor : کاربر

هدف: امکان دسترسی کاربر به قابلیت‌های سیستم و ذخیره اطلاعات شخصی.  
پیش شرط: 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 خطا به کاربر نمایش داده می‌شود.  
پس شرط: 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.  
پیش شرط: 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 خطا نمایش داده می‌شود.  
پس شرط: آلام جدید در سیستم ذخیره می‌شود.

### ۳.۴.۲ UseCase : انتخاب چالش

#### Actor : کاربر

هدف: انتخاب چالش مناسب برای خاموش کردن آلام.  
پیش شرط: آلام قبلاً 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 پیش فرض اختصاص می‌دهد.  
پس شرط: چالش به آلام متصل می‌شود.

### ۴.۴.۲ UseCase : فعال شدن آلام

#### Actor : سیستم

هدف: فعال سازی آلام در زمان تعیین شده.  
پیش شرط: آلام فعال وجود داشته باشد.  
سناریوی اصلی: در زمان مشخص، سیستم آلام را فعال کرده و چالش مربوطه را نمایش می‌دهد.  
پس شرط: 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.

پیش شرط: 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.

پس شرط: 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 آلام ها و عملکرد گذشته.

پیش شرط: کاربر وارد سیستم شده باشد.

سناریوی اصلی: کاربر وارد بخش تاریخچه شده و اطلاعات مربوط به آلام ها را مشاهده می کند.

پس شرط: اطلاعات تاریخچه به صورت به روز نمایش داده می شود.

#### ۷.۴.۲ UseCase : دریافت اعلان

##### Actor : سرویس اعلان

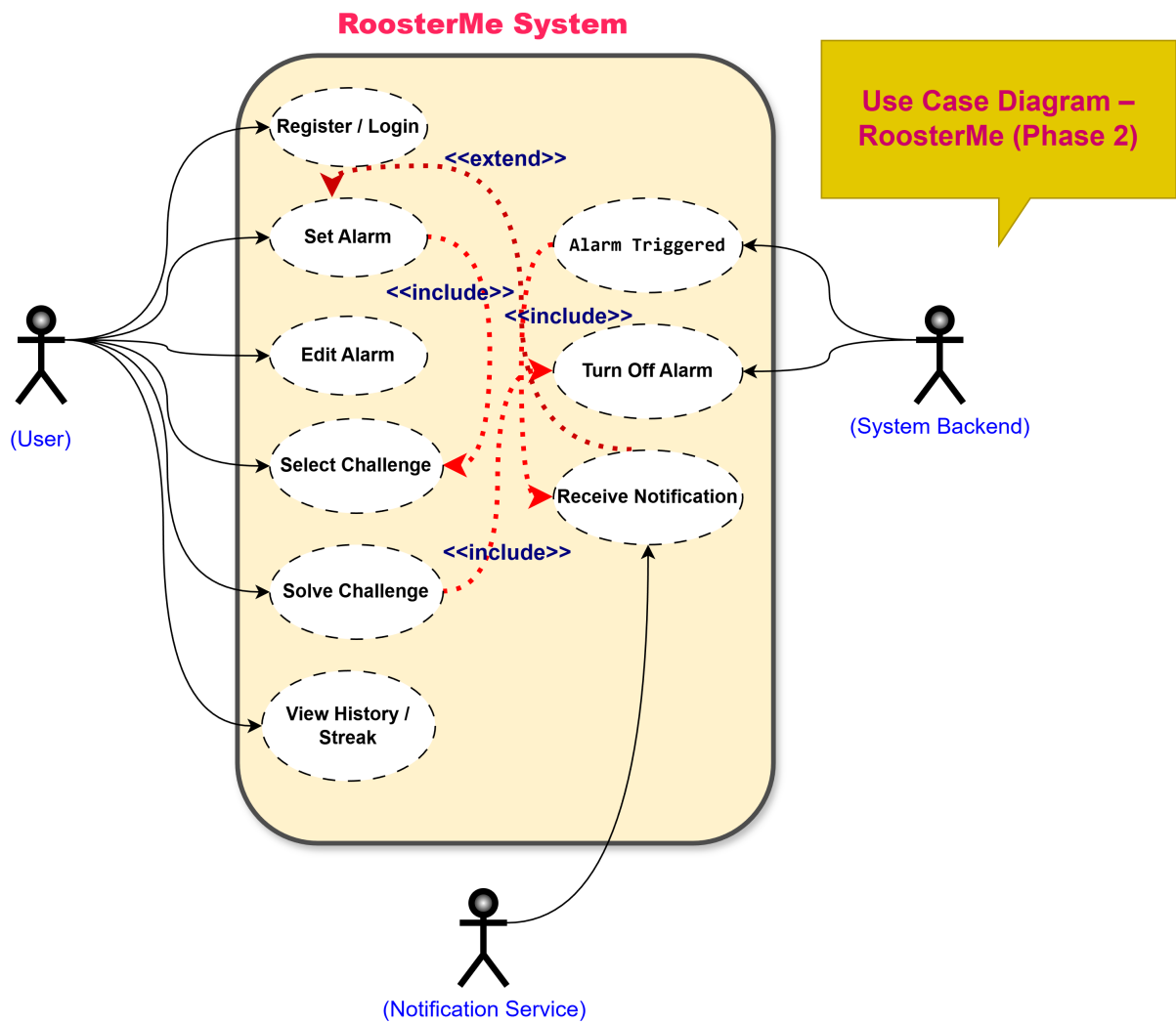
هدف: اطلاع رسانی به کاربر در مورد وضعیت آلام.

پیش شرط: 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.

پس شرط: کاربر از وضعیت آلام مطلع می شود.





شكل ١: UseCase Diagram سیستم RoosterMe

تمام دیاگرام‌ها در پوشه Diagrams آورده شده است .

## ۳ Activity Diagram – فرآیند کامل اجرای آلارم چالشی (RoosterMe)

### 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 آلارم چالشی طراحی شده است. این فرآیند از لحظه‌ای که کاربر وارد برنامه می‌شود آغاز شده و تا زمان خاموش شدن موفق آلارم ادامه می‌یابد. هدف اصلی این دیاگرام، ارائه‌ی شفاف از نحوه تعامل کاربر با سیستم و منطق داخلی سیستم در مدیریت و اجرای آلارم چالشی است. این دیاگرام کمک می‌کند تا رفتار سیستم در یک سناریوی واقعی به صورت مرحله به مرحله 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 آلارم، انتخاب چالش و پاسخ‌دهی به آن.
  - (RoosterMe System): مسئول ذخیره‌سازی اطلاعات آلارم، زمان‌بندی، فعال‌سازی آلارم در زمان مقرر و بررسی صحت پاسخ چالش.
- فعالیت‌ها در این دیاگرام به گونه‌ای طراحی شده‌اند که تعامل منطقی و پیوسته بین کاربر و سیستم را نشان دهند.

### ۳.۳ توضیح کلی جریان فعالیت‌ها

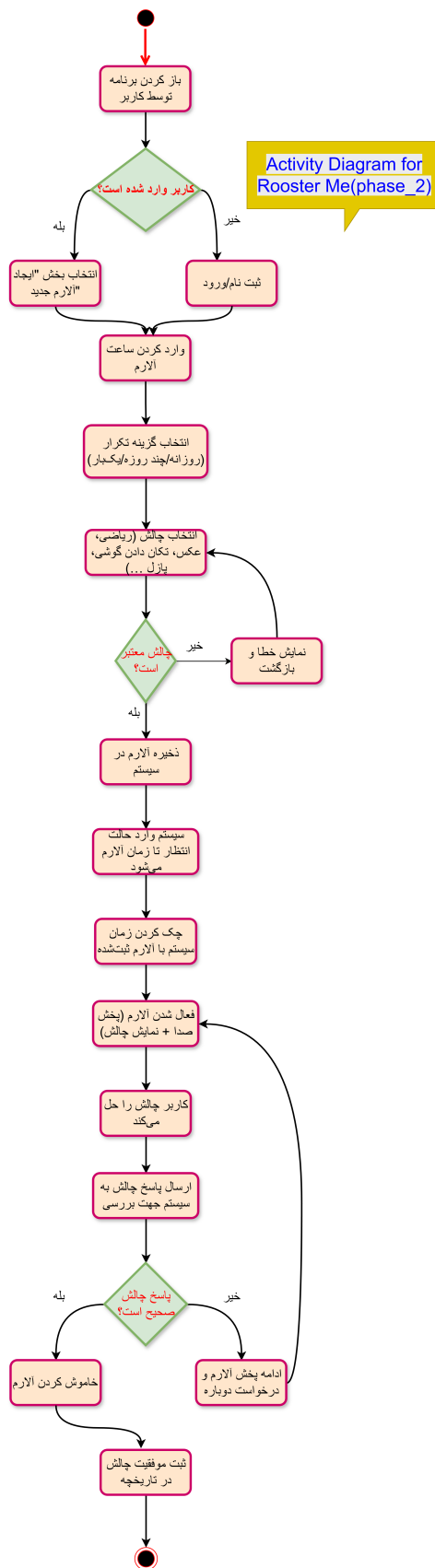
دیاگرام فعالیت، فرآیند اصلی عملکرد سیستم RoosterMe را به صورت یک جریان منظم و گام به گام نمایش می‌دهد. این فرآیند شامل مراحل زیر است:

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 آلارم جدید (Create Alarm): پس از ورود موفق، کاربر اقدام به ایجاد یک آلارم 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 آلارم خواهد بود.
۴. ذخیره آلارم در سیستم (Save Alarm): سیستم اطلاعات مربوط به آلارم، شامل زمان و نوع چالش، را ذخیره می‌کند.

۵. انتظار سیستم تا زمان آلارم (Wait Until Alarm Time) : سیستم وارد حالت انتظار شده و زمان تنظیم شده را پایش می‌کند.
۶. فعال‌سازی آلارم (Trigger Alarm) : در زمان مقرر، آلارم فعال شده و اعلان مربوطه برای کاربر نمایش داده می‌شود.
۷. اجرای چالش (Start Challenge) : پس از فعال شدن آلارم، چالش انتخاب شده برای کاربر نمایش داده می‌شود و کاربر باید آن را انجام دهد.
۸. بررسی صحت پاسخ (Validate Answer) : سیستم پاسخ یا عملکرد کاربر را بررسی می‌کند.
۹. تصمیم‌گیری (Decision) : در صورت صحیح بودن پاسخ، فرآیند به مرحله خاموش شدن آلارم منتقل می‌شود. در غیر این صورت، کاربر باید چالش را مجدداً انجام دهد.
۱۰. خاموش شدن آلارم (Turn Off Alarm) : پس از انجام موفق چالش، آلارم 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 آلارم 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 منطق کنترلی سیستم و نحوه مدیریت رفتار کاربر را به صورت شفاف نمایش می‌دهد و به درک بهتر عملکرد کلی سیستم کمک می‌کند.



شکل ۲: Activity Diagram سیستم RoosterMe

تمام دیاگرام‌ها در پوشه Diagrams آورده شده است .

## ۴ Sequence Diagram

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

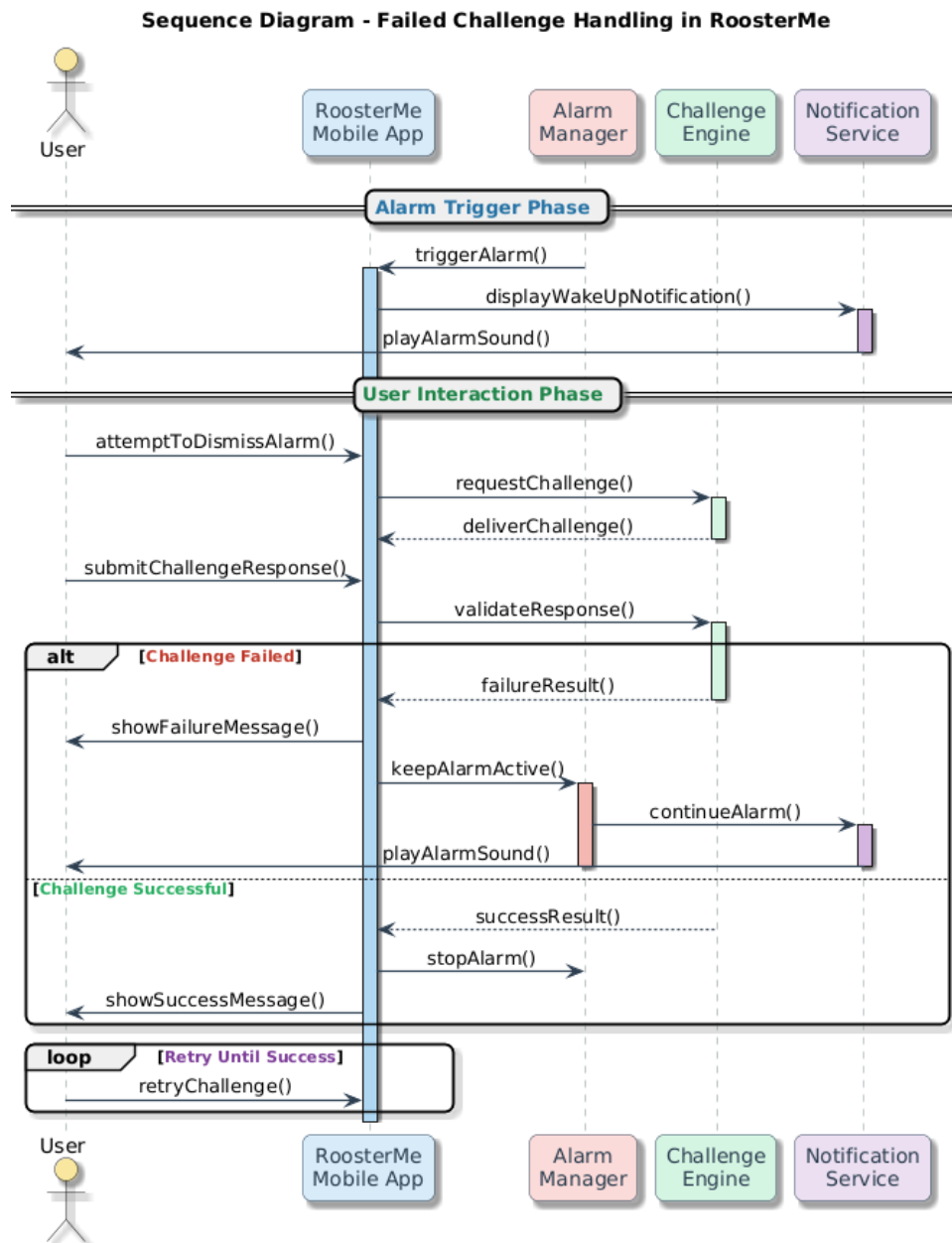
### ۱.۴ Challenge-Based Alarm Dismissal

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

اپلیکیشن از طریق سرویس اعلان و صدا، اعلان بیدارباش و صدای آلارم را برای کاربر پخش می‌نماید. با تلاش کاربر برای خاموش کردن آلارم، 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 خطا به کاربر نمایش داده شده و وضعیت آلارم همچنان فعال باقی می‌ماند و صدای آلارم 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 آلارم صادر می‌شود. در نهایت، پیام موفقیت به کاربر نمایش داده شده و فرآیند بیدارباش به پایان می‌رسد. این نمودار، تعامل میان کاربر و اجزای اصلی سیستم شامل اپلیکیشن موبایل، سرویس مدیریت آلارم، موتور ارزیابی چالش و سرویس اعلان را به صورت شفاف نشان می‌دهد.



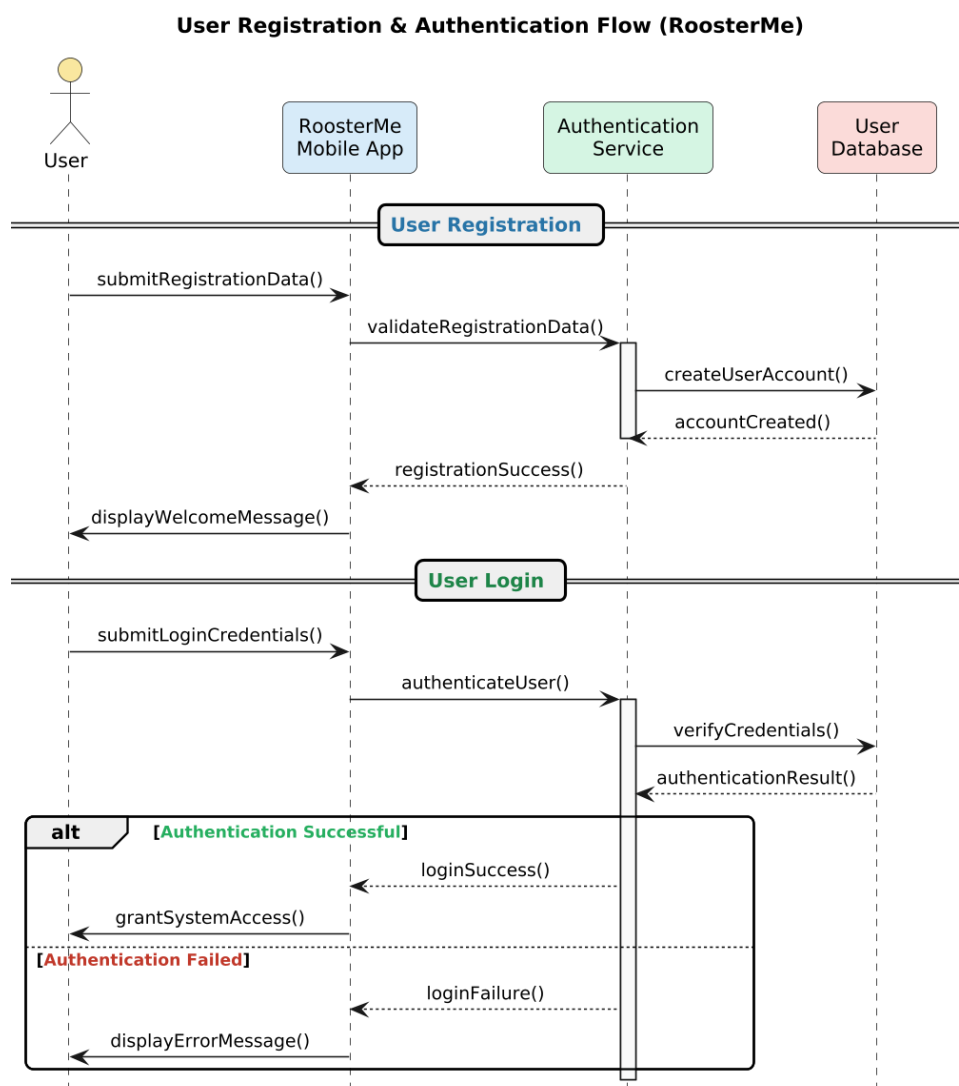
شکل ۳: Challenge-Based Alarm Dismissal in RoosterMe

## ۲.۴ User Registration & Authentication Flow

این نمودار توالی، فرآیند ثبت نام و احراز هویت کاربران در سامانه RoosterMe را نمایش می دهد. هدف این فرآیند، ایجاد دسترسی امن و کنترل شده برای کاربران و مدیریت صحیح اطلاعات هویتی آن ها در سیستم است. در بخش ثبت نام،

کاربر اطلاعات اولیه خود را از طریق اپلیکیشن موبایل ارسال می کند. اپلیکیشن این داده ها را برای بررسی صحت و اعتبار به سرویس احراز هویت ارسال می نماید. پس از اعتبارسنجی اطلاعات، سرویس احراز هویت درخواست ایجاد حساب کاربری جدید را به پایگاه داده کاربران ارسال می کند. با ایجاد موفق حساب کاربری، نتیجه به اپلیکیشن بازگردانده شده و

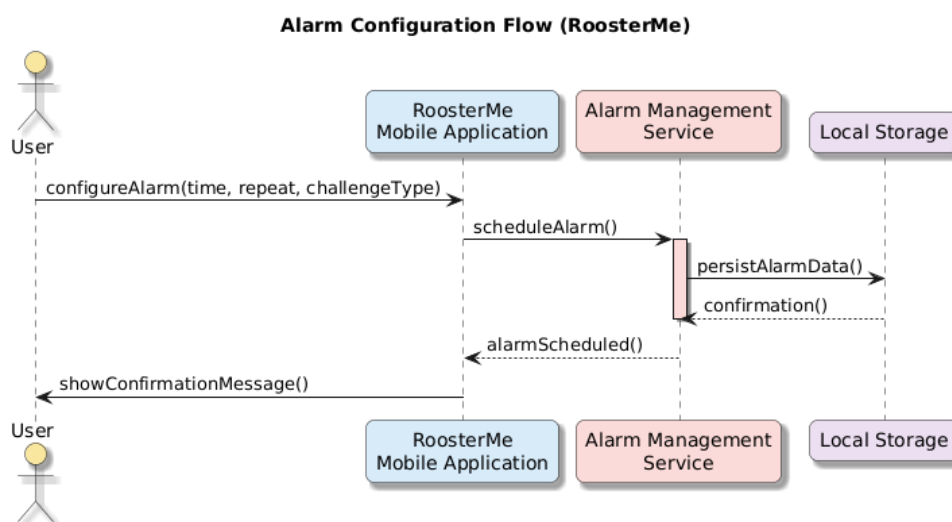
پیام خوش‌آمدگویی به کاربر نمایش داده می‌شود. در بخش ورود به سیستم، کاربر اطلاعات ورود خود را به اپلیکیشن ارسال می‌کند. اپلیکیشن درخواست احراز هویت را به سرویس مربوطه منتقل کرده و این سرویس با بررسی اطلاعات در پایگاه داده کاربران، نتیجه احراز هویت را مشخص می‌نماید. در صورت موفقیت‌آمیز بودن فرآیند احراز 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.



شکل 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 بیداریش، 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.



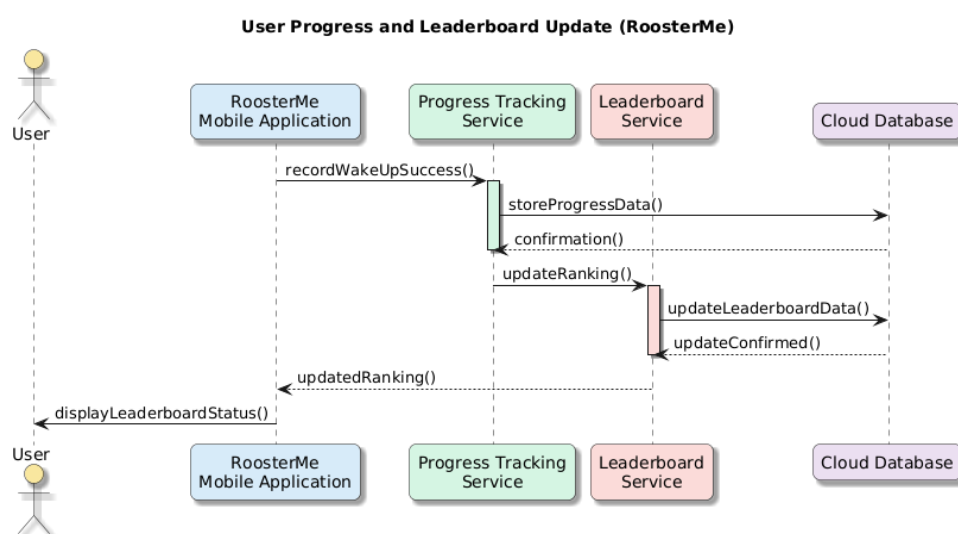
شکل ۵: Alarm Configuration and Successful Registration Flow in RoosterMe

### ۴.۴ ارسال اعلان و یادآوری

این نمودار توالی، فرآیند ارسال اعلان‌ها، ثبت پیشرفت کاربر و به‌روزرسانی 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 به‌روزرسانی 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 به‌روزشده 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

## ۵ Data Flow Diagram

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 بیدارباش، اجرای چالش و ثبت پیشرفت کاربر طراحی شده و دید جامعی از جریان داده‌ها در سطح عملیاتی سیستم فراهم می‌کند. در این نمودار، کاربر به‌عنوان موجودیت خارجی اصلی، با سیستم تعامل دارد. فرآیند از مرحله زمان‌بندی آلارم آغاز

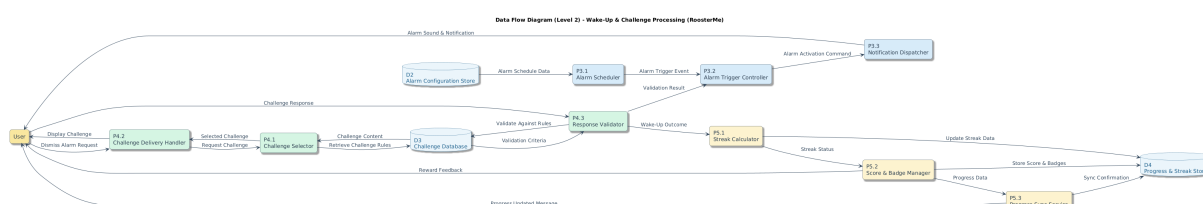
می‌شود؛ جایی که داده‌های مربوط به تنظیمات آلارم از مخزن پیکربندی آلارم دریافت شده و توسط ماژول زمان‌بندی آلارم پردازش می‌گردد. پس از فعال شدن آلارم، رویداد بیدارباش به ماژول کنترل آلارم ارسال شده و از طریق سرویس اعلان، صدای آلارم و پیام یادآوری به کاربر نمایش داده می‌شود.

پس از درخواست خاموش کردن آلارم توسط کاربر، سیستم وارد مرحله اجرای چالش می‌شود. در این مرحله، چالش

مناسب از پایگاه داده چالش‌ها انتخاب شده و از طریق ماژول تحویل 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 بیدارباش کاربر را مشخص می‌کند.

در صورت موفقیت کاربر، داده‌های مربوط به نتیجه بیدارباش به مازول محاسبه Streak ارسال می‌شود. سپس اطلاعات امتیاز، نشان‌ها و وضعیت پیشرفت کاربر توسط مازول مدیریت امتیاز و نشان پردازش شده و در مخزن پیشرفت و Streak 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 به‌صورت ساختاریافته و مرحله‌ای در سیستم RoosterMe جریان پیدا می‌کنند و تضمین می‌کند که فرآیندهای بیدارباش، چالش، اعتبارسنجی و ثبت پیشرفت به‌صورت منسجم، قابل ردیابی و قابل توسعه پیاده‌سازی شده‌اند.



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

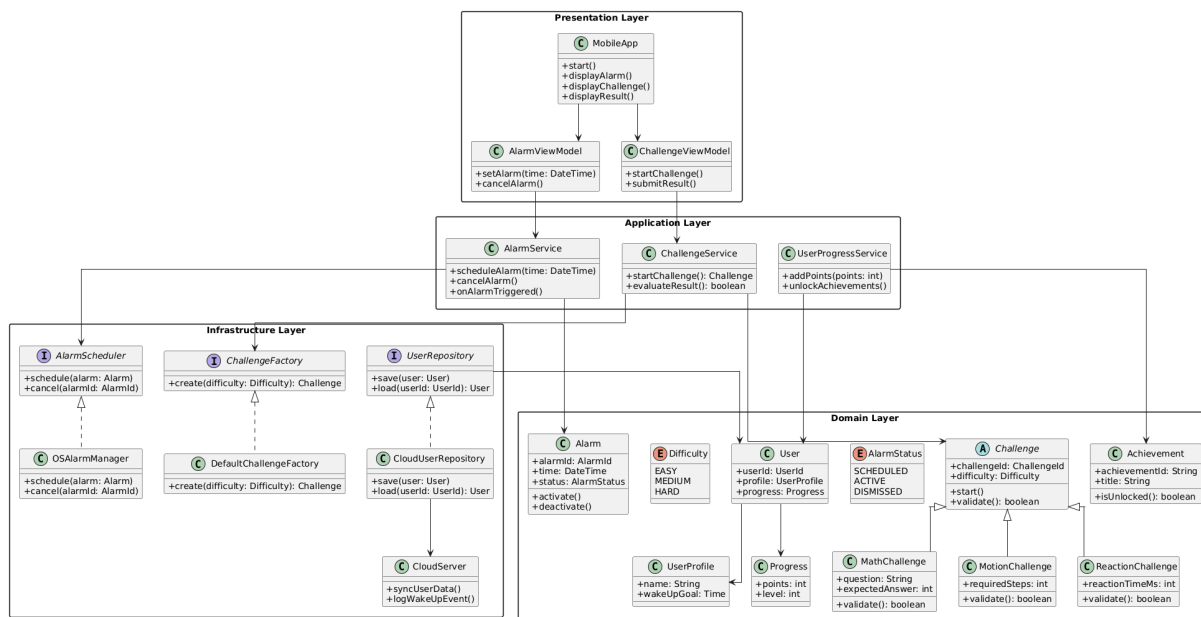
## ۶ Class Diagram

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 آلارم‌ها 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 به‌عنوان هسته مرکزی، هماهنگی بین آلارم‌ها، 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 به‌گونه‌ای طراحی شده‌اند که اصل تفکیک 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.

## ۷ System Architecture

### ۱.۷ توضیح معماری سیستم آلارم چالشی

معماری سیستم RoosterMe نمایی جامع، سطح بالا و ساخت یافته از طراحی نرم افزار ارائه می دهد و نحوه تعامل اجزای مختلف سیستم را برای اجرای صحیح، پایدار و امن فرآیندهای آلارم چالشی تشریح می کند. هدف از ارائه این معماری، ایجاد درکی شفاف از ساختار کلی سیستم، مسیر جریان داده ها، تفکیک مسئولیت ها و نحوه پیاده سازی اصول مهندسی نرم افزار در مقیاس پذیری، امنیت و توسعه پذیری است.

این معماری به عنوان مبنای اصلی طراحی و پیاده سازی سیستم عمل می کند و به توسعه دهندگان، طراحان و ذی نفعان کمک می کند تا پیش از ورود به جزئیات فنی، دیدی یکپارچه و دقیق از کل سیستم داشته باشند. همچنین، معماری سیستم نقش تعیین کننده ای در کیفیت نهایی نرم افزار، تجربه کاربری، قابلیت نگهداری بلندمدت و آمادگی سیستم برای توسعه های آینده ایفا می کند. معماری سیستم نقش حیاتی در موفقیت پروژه دارد، زیرا پایه ای برای پیاده سازی صحیح، نگهداری بلندمدت، توسعه قابلیت های جدید و اطمینان از تجربه کاربری بهینه فراهم می کند.

## 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-سرور بوده و با هدف پاسخ‌گویی به نیازهای عملکردی و غیرعملکردی سیستم انتخاب شده است. در این معماری، اپلیکیشن موبایل به‌عنوان 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 به‌صورت 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 آلارم 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 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 اجزای 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 آلارم‌ها 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 آلارم‌های 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

## ۲.۴.۷ Content Delivery Network (CDN)

CDN به منظور بهبود سرعت دسترسی کاربران به محتوای ایستا و کاهش تأخیر شبکه در معماری 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 به‌ویژه برای کاربران با موقعیت جغرافیایی متفاوت فراهم می‌کند.

## ۳.۴.۷ Load Balancer

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 دسترسی پذیری سیستم
- افزایش تحمل خطا در صورت از کار افتادن یک سرور
- امکان مقیاس‌پذیری افقی بدون تغییر در کلاینت

## ۴.۴.۷ سرورهای بک‌اند (Backend Servers)

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 معماری، 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.

## ۵.۴.۷ Cache Layer

لایه Cache بین سرورهای بک‌اند و پایگاه داده قرار دارد و داده‌هایی که به‌طور مکرر مورد استفاده قرار می‌گیرند را به‌صورت 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 به‌صورت لایه‌ای طراحی شده است:

- لایه رابط کاربری (UI Layer)
  - لایه تحویل 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.

## ۶.۷ 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 آلارم
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 آلارم و نتایج چالش باید بلافاصله 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 خطا: خطاها 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.

### ۲.۶.۷ 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 آلارم‌ها 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

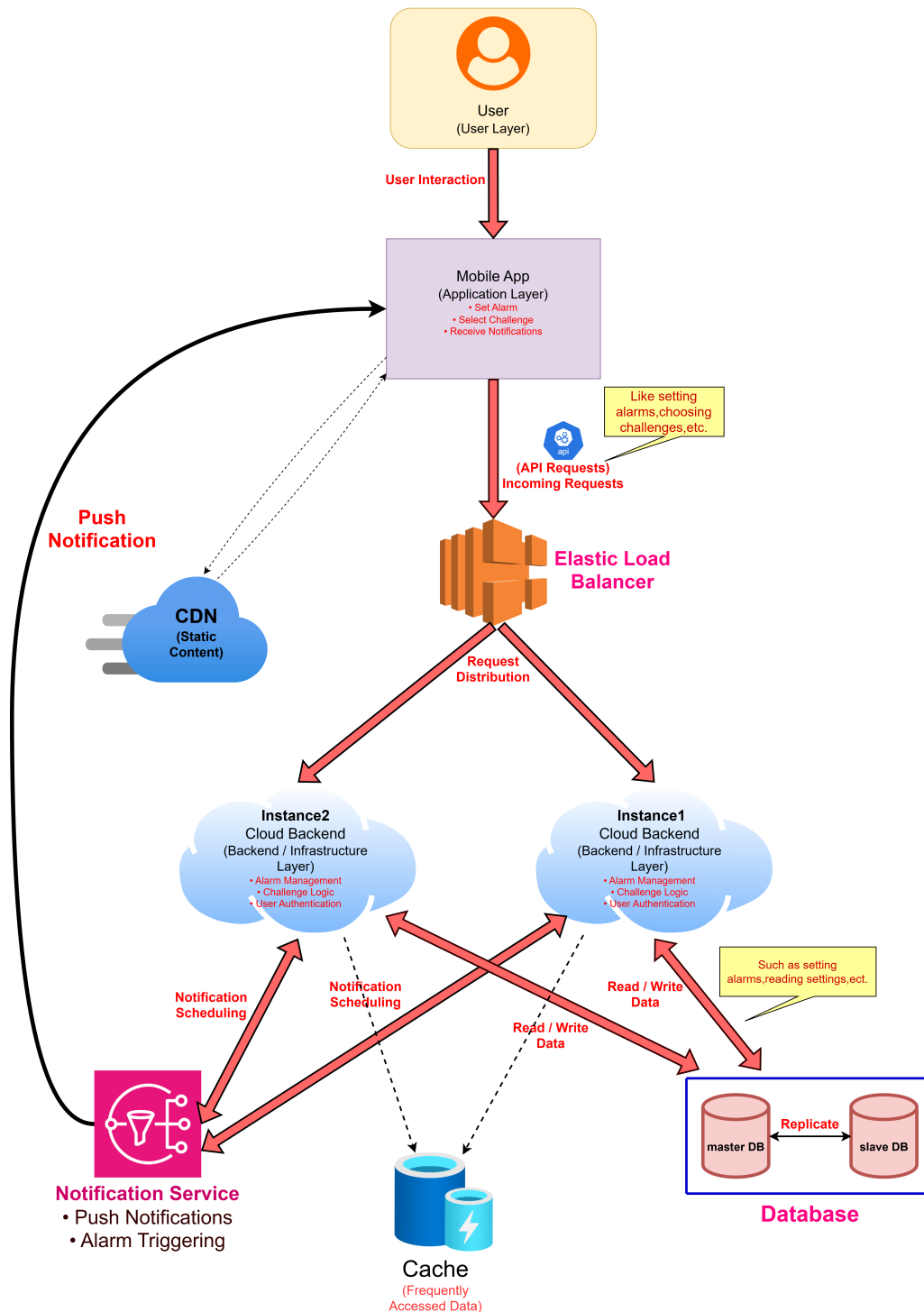
معماری سیستم RoosterMe به‌گونه‌ای طراحی شده است که امنیت داده‌ها، پایداری سیستم و مقیاس‌پذیری در اولویت قرار دارند. استفاده از تفکیک لایه‌ها، توزیع بار، Cache و سرویس‌های مستقل، ریسک خرابی سیستم را کاهش داده و امکان توسعه آینده را فراهم می‌کند.

## ۸.۷ جمع‌بندی معماری سیستم

معماری سیستم RoosterMe یک ساختار مدرن، مقیاس‌پذیر و مهندسی‌شده را ارائه می‌دهد که تمامی نیازهای یک سیستم آلارم 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)



شکل ۹: RoosterMe سیستم System Architecture Diagram

تمام دیاگرام‌ها در پوشه Diagrams آورده شده است.

## ۸ تاریخچه نسخه های سند فاز دوم پروژه

شرح تغییرات	نویسنده	تاریخ	نسخه
تحلیل اولیه پروژه و تعریف دامنه سیستم آلارم چالشی؛ شناسایی بازیگران اصلی، اهداف سیستم و نیازمندی‌های سطح بالا	مریم رجبی	۱۴۰۴/۰۹/۰۳	۱
طراحی و مستندسازی UseCase Diagram شامل شناسایی بازیگران، تعریف های UseCase اصلی، روابط include و extend و تشریح سناریوهای اصلی و جایگزین	مریم رجبی	۱۴۰۴/۰۹/۰۶	۲
طراحی و تکمیل Activity Diagram برای فرآیند کامل اجرای آلارم چالشی، شامل جریان ورود کاربر، ایجاد آلارم، انتخاب چالش، فعال‌سازی آلارم، تصمیم‌گیری و خاموش شدن آلارم	مریم رجبی	۱۴۰۴/۰۹/۱۱	۳
طراحی معماری اولیه سیستم شامل اپلیکیشن موبایل، سرور بک‌اند، پایگاه داده و سیستم اعلان و تشریح تعاملات و جریان داده‌ها	مریم رجبی	۱۴۰۴/۰۹/۱۵	۴
ارتقای معماری سیستم با افزودن Load Balancer، CDN و Cache به منظور افزایش مقیاس‌پذیری، بهبود عملکرد، پایداری و آمادگی برای توسعه در مقیاس بالا	مریم رجبی	۱۴۰۴/۰۹/۱۷	۵
تکمیل مستندات معماری سیستم شامل توضیح لایه‌ها، مسئولیت اجزا، ملاحظات امنیتی، مدیریت خطا و پشتیبانی از توسعه آینده	مریم رجبی	۱۴۰۴/۰۹/۲۰	۶
طراحی و مستندسازی Sequence Diagram های سیستم شامل فرآیند خاموش‌سازی آلارم مبتنی بر چالش، ثبت‌نام و احراز هویت کاربر، پیکربندی و ثبت آلارم، و ارسال اعلان و به‌روزرسانی پیشرفت کاربر	مریم پورحسن	۱۴۰۴/۰۹/۲۲	۷
طراحی 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	مریم پورحسن	۱۴۰۴/۰۹/۲۳	۸
طراحی و مستندسازی Class Diagram سیستم RoosterMe شامل کلاس‌های اصلی User، Alarm، CloudSer-، Challenge، Alarm Manager و vice و تشریح روابط و مسئولیت‌های هر کلاس	مریم پورحسن	۱۴۰۴/۰۹/۲۴	۹

جدول ۱: سوابق تغییرات، طراحی و توسعه سیستم RoosterMe