



مرکز تحقیقات فضایی

عنوان اختصاری پروژه: اینترنت اشیاء کشاورزی

کد پروژه: CNS۹۹۹۷-۰۱

کد فعالیت: CNS۹۹۹۷-۰۱-۰۷

طراحی نرم افزار - نیازمندی های کارکردی

کد سند: MSRI-CNS۹۹۹۷-۰۱/R-۰۷/۱۲

شماره ویرایش: ۱/۱


طبقه بندی: عادی

تاریخ: ۹۷/۰۴/۱۶

تعداد کل صفحات: ۶۰ صفحه

(با احتساب برگ روی جلد)

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

کد سند: MSRI-CNS۹۹۹۷-۰۱/R-۰۷/۱۲ ویرایش: ۱/۱ طبقه بندی: عادی	<b>کتابخانه های طراحی نرم افزار - نیازمندی های کارکردی</b>	 پژوهشگاه فضایی ایران
---	--	--

## شناسنامه سند

### ۱- مشخصات پروژه

عنوان کامل پروژه	انجام مطالعات امکانسنجی و پیاده سازی ایده اینترنت اشیا در حوزه کشاورزی			مدیر پروژه	-
کد پروژه	CNS۹۹۹۷-۰۱	پژوهشکده (حوزه) مجری	مرکز تحقیقات فضایی	تاریخ شروع پروژه	۹۵/۰۲/۰۱
کد فعالیت	CNS۹۹۹۷-۰۱-۰۷			تاریخ خاتمه پروژه	۹۵/۰۹/۰۱


### ۲- مشخصات سند

عنوان سند	تحلیل طراحی نرم افزار - نیازمندی های کارکردی			تعداد صفحات	
طبقه بندی سند	عادی	کد سند	MSRI-CNS۹۹۹۷-۰۱/R-۰۷/۱۲	کل سند	۶۰
ویرایش	۱/۱	تاریخ ویرایش	۹۷/۰۴/۱۶	پیوست ها	۵۳

### ۳- جدول تهیه، تایید و تصویب در پژوهشکده (حوزه) مجری

سمت*	نام و نام خانوادگی	امضا	تاریخ
تهیه کننده (گان)	رئیس بخش طراحی و پیاده سازی نرم افزار		
تایید کننده (گان)	رئیس اداره برنامه ریزی		
تصویب کننده	رئیس مرکز		

\* برای مواردی که مجری، حوزه دیگری غیر از پژوهشکده است، مثل مراکز یا گروه های پژوهشی مستقل و ... از سمت های معادل بر اساس نظر رئیس حوزه استفاده شود.

کد سند: MSRI-CNS9997-01/R-07/12 ویرایش: ۱/۱ طبقه‌بندی: عادی	<b>کتابخانه‌های طراحی نرم‌افزار - نیازمندی‌های کارکردی</b>	 <b>پژوهشگاه فضایی ایران</b>
---	--	---

### شناسنامه سند (ادامه)

#### ۴- جدول تایید و تصویب در پژوهشگاه

تاریخ	امضا	نام و نام خانوادگی	سمت	
			مدیران مرکز طراحی و توسعه سامانه‌های فضایی یا مدیر پژوهش و فناوری (برحسب مورد)	تاییدکننده(گان)
			سایر افراد (مانند معاون تضمین کیفیت، بهره‌بردار و ... براساس قرارداد یا نظر تصویب‌کننده و مدیریت کنترل پروژه)	
			رییس مرکز طراحی و توسعه سامانه‌های فضایی یا معاون پژوهش و فناوری (برحسب مورد)	تصویب‌کننده

#### ۵- جدول توزیع نسخ (گیرندگان)

توزیع	عنوان واحد	توزیع	عنوان واحد
	مدیریت راهبرد و طراحی مأموریت		ریاست پژوهشگاه فضایی ایران
	مدیریت مهندسی سامانه‌های فضایی		دفتر ریاست، روابط عمومی و امور بین الملل
	مدیریت آزمون و عملیات میدان		مدیریت حراست
	معاونت اجرایی		اداره امور حقوقی
	مدیریت توسعه منابع انسانی		مدیریت نظارت و ارزیابی و پاسخگویی به شکایات
	مدیریت پشتیبانی		مدیریت بازرگانی خارجی
	مدیریت امور مالی	●	مدیریت طرح و برنامه
	اداره تشکیلات و بهبود روش‌ها		معاونت پژوهش و فناوری
	سازمان فضایی ایران		مدیریت آموزش و تحصیلات تکمیلی
	پژوهشکده سامانه‌های حمل‌ونقل فضایی	✓	مدیریت پژوهش و فناوری
	پژوهشکده سامانه‌های ماهواره		مدیریت دانش
	پژوهشکده مکانیک		معاونت تضمین کیفیت و ایمنی
	پژوهشکده مواد و انرژی		مدیریت مهندسی تضمین کیفیت
	پژوهشکده رانشگرهای فضایی		مدیریت کالیبراسیون و استاندارد
✓	مرکز تحقیقات فضایی		مدیریت ایمنی و محیط زیست
			مرکز طراحی و توسعه سامانه‌های فضایی
			سایر گیرندگان:

\*توزیع نسخ بر اساس علامت‌های زیر انجام می‌شود:

✓: سند برای این واحدها ارسال می‌شود.

●: سند برای این واحدها ارسال نمی‌شود و صرفاً اطلاع‌رسانی می‌شود.

#### ۶- تایید مرکز اسناد

مدیریت دانش (مرکز اسناد) پژوهشگاه فضایی ایران
نام و نام خانوادگی: .....
تاریخ: .....
مهر و امضا

مدیریت دانش (مرکز اسناد) پژوهشکده مجری
نام و نام خانوادگی: .....
تاریخ: .....
مهر و امضا

## شناسنامه سند (ادامه)

## ۷- جدول مشخصات و شرح وظایف دست اندرکاران تدوین سند\*

ردیف	نام و نام خانوادگی	آخرین مدرک تحصیلی	رشته تحصیلی	مرتبه علمی**	محل کار	شرح وظایف	درصد مشارکت
۱	دانشگاه امیرکبیر				دانشگاه امیرکبیر		۱۰۰
۲							
۳							
۴							
۵							
۶							
۷							
۸							
۹							
جمع							۱۰۰

\*منظور کلیه افرادی است که در انجام فعالیت های مرتبط با این سند نقش اصلی داشته اند.

\*\*برای اعضای هیات علمی از عناوین مربوط (استاد، دانشیار، استادیار، مربی) و برای دیگر پژوهشگران از عنوان کارشناس استفاده شود.

## ۸- دیگر همکاران تدوین سند\*

ردیف	نام و نام خانوادگی	آخرین مدرک تحصیلی	رشته تحصیلی	مرتبه علمی	محل کار	نقش

\* منظور کسانی است که ضمن مطالعه سند، نظرات قابل توجهی را در خصوص سند ارائه کرده اند. ویراستاران ادبی نیز در این جدول ذکر می شوند.



## چکیده

گزارش حاضر حاوی سند دریافتی از دانشگاه صنعتی امیرکبیر (پیمانکار پژوهشگاه فضایی ایران در قرارداد "طراحی و پیاده سازی پلتفرم اینترنت اشیا") با عنوان "طراحی نرم‌افزار - نیازمندی‌های کارکردی" می باشد.

## واژه‌های کلیدی:

اینترنت اشیا، پلتفرم، کشاورزی هوشمند



دانشگاه صنعتی امیرکبیر  
(پلی تکنیک تهران)

## پروژه طراحی و پیاده سازی پلتفرم اینترنت اشیا



### عنوان:

## طراحی نرم افزار - نیازمندی های کارکردی

### ارائه دهنده:

کارگروه پلتفرم، گروه پژوهشی اینترنت اشیا، دانشگاه صنعتی امیرکبیر

### کد سند:

ISRC-AUT-۹۷۰۱۲۱۰۰

### تاریخ انتشار:

۱۳۹۷/۰۱/۲۱

### حق مالکیت سند

این سند در مالکیت کارگروه پلتفرم، گروه پژوهشی اینترنت اشیا، دانشگاه صنعتی امیرکبیر به نشانی تهران، خیابان حافظ، دانشگاه صنعتی امیرکبیر، دانشکده مهندسی کامپیوتر و فناوری اطلاعات بوده و شامل اطلاعات محرمانه و تجاری است. مالکیت این سند را نمی توان بدون کسب اجازه کتبی از آزمایشگاه اینترنت اشیا به شخص حقیقی یا حقوقی دیگری انتقال داد. هیچ کدام از اقلام این سند را نمی توان بدون اجازه کتبی از آزمایشگاه اینترنت اشیا مورد استفاده قرار داد، مجدداً استفاده نمود، یا منتشر کرد.



## اطلاعات سند

نام پروژه:	پروژه طراحی و پیاده سازی پلتفرم اینترنت اشیا
عنوان سند:	طراحی نرم افزار-نیازمندی های کارکردی
کد سند:	ISRC-AUT-۹۷۰۱۲۱،۰
نگارش:	۱/۰
نام تهیه کنندگان:	تیم فنی
تاریخ تهیه:	۱۳۹۷/۰۱/۲۱
نام بازبینی کننده:	مدیر پروژه
تاریخ آخرین بازبینی:	۱۳۹۷/۰۱/۲۴
نام تأیید کننده:	
تاریخ تأیید:	
وضعیت:	نسخه اولیه
نوع طبقه بندی سند:	محرمانه

### چکیده:

این سند سنگ بنای طراحی پروژه پلتفرم اینترنت اشیا می باشد. بخش اصلی آن، معماری کلان پلتفرم اینترنت اشیا را پوشش می دهد. معماری ارائه شده براساس موارد استخراج شده در فاز تحلیل نیازمندی ها طراحی گردیده است. در طراحی این پلتفرم از معماری میکروسرویس استفاده شده است که طبق بررسی های انجام شده، یکی از معماری های مطلوب جهت توسعه پلتفرم های اینترنت اشیا می باشد. علاوه بر معماری کلان پلتفرم، طراحی بخش های مختلف پلتفرم در این سند بررسی می گردد و ارتباطات بین اجزای پلتفرم مشخص می گردد.

دو مورد در خصوص این سند لازم به ذکر است:

- این سند، مستندسازی سامانه توسعه داده شده نیست. بنابراین به هیچ وجه به جزئیات پیاده سازی پلتفرم نپرداخته است. بلکه هدف اصلی این مستند، تشریح اجزای اصلی پلتفرم و ارتباطات آنها با یکدیگر است.
- این نسخه از سند، طراحی نهایی پلتفرم نیست. با توجه به نظرات کارفرمای محترم، همچنین تست های عملیاتی سامانه، امکان تغییرات جزئی در این معماری وجود دارد.

### واژه های کلیدی:

پلتفرم اینترنت اشیا، طراحی معماری، میکروسرویس

نوع طبقه بندی سند: محرمانه	کد سند: ۹۷۰۱۲۱،۰-ISRC-AUT	تاریخ: ۱۳۹۷/۰۱/۲۱	صفحه: ۳ از ۵۳
تمامی اطلاعات موجود در این سند متعلق به کارگروه پلتفرم، گروه پژوهشی اینترنت اشیا، دانشگاه صنعتی امیرکبیر بوده و حقوق قانونی آن محفوظ است.			

## فهرست مطالب

چکیده	۳
۱- مقدمه	۹
۲- معماری کلان پلتفرم	۱۰
۱-۲- مقدمه	۱۰
۲-۲- معماری میکروسرویس	۱۰
۳-۲- معماری پلتفرم مبتنی بر میکروسرویس	۱۱
۳- مولفه ها و فرآیندهای پلتفرم اشیا	۱۴
۱-۳- مقدمه	۱۴
۲-۳- مولفه API-Server	۱۴
۳-۳- مولفه UI Server و Web UI	۱۵
۴-۳- مولفه Project Manager	۱۷
۵-۳- مولفه Up-link	۱۸
۶-۳- مولفه Down-link	۱۹
۷-۳- مولفه Data Manager	۲۰
۸-۳- مولفه Orchestrator	۲۱
۴- پروتکل های ارتباطی با اشیا	۲۳
۱-۴- مقدمه	۲۳
۲-۴- پروتکل LoRaWAN	۲۳
۳-۴- پروتکل LAN	۲۶
۴-۴- واحدهای مرتبط به انتخاب پروتکل ارتباطی با اشیا	۲۷
۵- پروتکل ارتباطی با برنامه های کاربردی	۲۸
۱-۵- مقدمه	۲۸
۲-۵- کلیات پروتکل	۲۸
۳-۵- جزییات API مربوط به برنامه های کاربردی	۲۸
۶- پروتکل ارتباطی با ابزارهای تحلیل داده	۳۰

۳۰	۶-۱- مقدمه
۳۰	۶-۲- اتصال به ابزارهای داده
۳۱	۷- حسابرسی
۳۱	۷-۱- مقدمه
۳۱	۷-۲- سرویس های ارائه شده در این بخش
۳۲	۷-۳- فرایند خرید و پرداخت
۳۳	۸- پایگاه داده
۳۳	۸-۱- مقدمه
۳۳	۸-۲- خوشه بندی در پایگاه داده Mongo
۳۴	۸-۳- تکثیر پایگاه داده
۳۶	۸-۴- معماری پیشنهادی
۳۷	۸-۵- انواع پایگاه داده های پلتفرم
۳۷	۸-۵-۱- پایگاه داده های خام
۳۷	۸-۵-۲- پایگاه داده های پارس شده
۳۷	۸-۵-۳- پایگاه داده های پلتفرم
۳۸	۸-۶- سایر موارد مرتبط به پایگاه داده ها
۳۹	۹- طراحی فرآیند مدیریت اشیا
۴۰	۱۰- طراحی و mockup پنل کاربر
۵۱	۱۱- طراحی و mockup پنل مدیریت

## فهرست اشکال

شکل ۱-۲ معماری میکروسرویس	۱۱
شکل ۲-۲ معماری سیستم پیشنهادی	۱۲
شکل ۱-۳ فرآیند ثبت نام کاربر	۱۶
شکل ۲-۳ فرآیند تعریف شی	۱۶
شکل ۳-۳ سناریوی تعریف پروژه	۱۸
شکل ۴-۳ سناریوی دریافت داده	۱۹
شکل ۵-۳ سناریوی ارسال داده	۲۰
شکل ۶-۳ سناریوی نمایش داده	۲۱
شکل ۱-۴ معماری مدل ارتباطی مبتنی بر LoRaServer	۲۴
شکل ۲-۴ ارتباط اشیا با پلتفرم از طریق Loraserver.io	۲۵
شکل ۳-۴ معماری گذرگاه LAN جهت ارتباط با اشیا	۲۶
شکل ۱-۵ سناریوی ثبت نام کاربر از طریق اپلیکیشن اندروید	۲۹
شکل ۱-۸ معماری خوشه ای	۳۴
شکل ۲-۸ معماری تکثیر پایگاه داده	۳۵
شکل ۳-۸ معماری تکثیر با استفاده از خوشه بندی	۳۶
شکل ۴-۸ معماری خوشه بندی با استفاده از تکثیر	۳۷
شکل ۱-۱۰ صفحه ورود کاربران	۴۰
شکل ۲-۱۰ صفحه ثبت نام کاربران	۴۰
شکل ۳-۱۰ صفحه ویرایش اطلاعات حساب کاربری	۴۱
شکل ۴-۱۰ داشبورد سامانه	۴۱
شکل ۵-۱۰ صفحه افزودن افزونه جدید به داشبورد	۴۲
شکل ۶-۱۰ نمایش لیست پروژه ها	۴۲
شکل ۷-۱۰ صفحه تعریف پروژه جدید	۴۳
شکل ۸-۱۰ صفحه مدیریت پروژه (۱)	۴۳
شکل ۹-۱۰ صفحه مدیریت پروژه (۲)	۴۳
شکل ۱۰-۱۰ صفحه مدیریت پروژه (۳)	۴۴
شکل ۱۱-۱۰ صفحه مدیریت پروژه (۴)	۴۴
شکل ۱۲-۱۰ صفحه افزودن شی جدید	۴۵
شکل ۱۳-۱۰ صفحه افزودن دسته ای اشیا	۴۵
شکل ۱۴-۱۰ صفحه فعال سازی اشیا	۴۶
شکل ۱۵-۱۰ صفحه ارسال کدک	۴۶

نوع طبقه بندی سند: محرمانه	کد سند: ISRC-AUT-۹۷۰۱۲۱،۰	تاریخ: ۱۳۹۷/۰۱/۲۱	صفحه: ۵۳ از ۵۳
تمامی اطلاعات موجود در این سند متعلق به کارگروه پلتفرم، گروه پژوهشی اینترنت اشیا، دانشگاه صنعتی امیرکبیر بوده و حقوق قانونی آن محفوظ است.			

شکل ۱۰-۱۶. صفحه ارسال داده به شی.....	۴۷
شکل ۱۰-۱۷. صفحه نمایش پروژه (۱).....	۴۷
شکل ۱۰-۱۸. صفحه نمایش پروژه (۲).....	۴۸
شکل ۱۰-۱۹. صفحه لیست اشیا.....	۴۸
شکل ۱۰-۲۰. صفحه لیست گذرگاه ها.....	۴۸
شکل ۱۰-۲۱. صفحه افزودن گذرگاه.....	۴۹
شکل ۱۰-۲۲. صفحه لیست پروفایل اشیا.....	۴۹
شکل ۱۰-۲۳. صفحه ساخت پروفایل اشیا.....	۵۰
شکل ۱۱-۱. صفحه لیست کاربران سامانه.....	۵۱
شکل ۱۱-۲. صفحه اطلاعات کاربر (۱).....	۵۲
شکل ۱۱-۳. صفحه اطلاعات کاربر (۲).....	۵۲
شکل ۱۱-۴. صفحه لیست بسته های موجود در سامانه.....	۵۲
شکل ۱۱-۵. صفحه افزودن بسته جدید.....	۵۳
شکل ۱۱-۶. صفحه مدیریت درگاه های بانکی.....	۵۳

نوع طبقه بندی سند: محرمانه	کد سند: ۹۷۰۱۲۱،۰-ISRC-AUT	تاریخ: ۱۳۹۷/۰۱/۲۱	صفحه: ۷ از ۵۳
تمامی اطلاعات موجود در این سند متعلق به کارگروه پلتفرم، گروه پژوهشی اینترنت اشیا، دانشگاه صنعتی امیرکبیر بوده و حقوق قانونی آن محفوظ است.			

## فهرست جداول:

جدول ۱-۲. واحدها به همراه شرح وظایف ..... ۱۳

نوع طبقه بندی سند: محرمانه	کد سند: ۹۷۰۱۲۱،۰-ISRC-AUT	تاریخ: ۱۳۹۷/۰۱/۲۱	صفحه: ۸ از ۵۳
تمامی اطلاعات موجود در این سند متعلق به کارگروه پلتفرم، گروه پژوهشی اینترنت اشیا، دانشگاه صنعتی امیرکبیر بوده و حقوق قانونی آن محفوظ است.			

### ۱- مقدمه

فاز طراحی یکی از فازهای اصلی در توسعه نرم افزار است. اسناد طراحی در حقیقت سنگ بنای مراحل توسعه، تست و ارزیابی سیستم می باشند. در فصل دوم این مستند معماری کلان پلتفرم به همراه کلیت مولفه های آن مورد بررسی قرار می گیرد. فصل سوم اجزای داخلی پلتفرم به همراه فرآیندها یا سناریوهای آن را ارائه می دهد. در ادامه در فصل چهارم پروتکل های ارتباطی با اشیا و نحوه ارتباط با Gateway ها تشریح می گردد. فصل پنجم نحوه ارتباط برنامه های کاربردی با پلتفرم را شرح می دهد. سپس در فصل ششم پروتکل ها و API های طراحی شده جهت ارتباط با ابزارهای تحلیل داده مورد بررسی قرار می گیرد. فصل هفتم جزئیات بیشتر از نحوه حسابرسی و شارژینگ را مطرح می کند. در فصل هشتم، طراحی پایگاه داده پلتفرم و انواع معماری های موجود را بررسی و روش پیشنهادی ارائه می گردد. فصل نهم به مدیریت اشیا اشاره ای دارد. در نهایت فصل های آخر به طراحی صفحات کاربری و مدیریت پلتفرم اختصاص پیدا کرده است.

نوع طبقه بندی سند: محرمانه	کد سند: ۹۷۰۱۲۱،۰-ISRC-AUT	تاریخ: ۱۳۹۷/۰۱/۲۱	صفحه: ۹ از ۵۳
تمامی اطلاعات موجود در این سند متعلق به کارگروه پلتفرم، گروه پژوهشی اینترنت اشیا، دانشگاه صنعتی امیرکبیر بوده و حقوق قانونی آن محفوظ است.			



## ۲- معماری کلان پلتفرم

### ۲-۱- مقدمه

یکی از فاکتورهایی که نقش کلیدی در موفقیت یک محصول نرم افزاری دارد، معماری آن است. طبیعتاً طراحی پلتفرم هم از این قضیه مستثنی نیست و تامین همه نیازمندی ها بخصوص نیازمندی های غیرکارکردی شامل دسترس پذیری و مقیاس پذیری در گرو استفاده از یک معماری مناسب برای سامانه است. معماری سامانه، اجزای داخلی سامانه و نحوه تعاملات آن ها را در راستای برآورد انتظارات کارفرما مشخص می کند. همان طور که در پروپوزال پروژه نیز شده است معماری ارائه شده بر اساس میکروسرویس می باشد. در ابتدا معماری میکروسرویس و مزایای آن تشریح شده است. سپس معماری پلتفرم پیشنهادی با جزئیات کامل ارائه و مورد بررسی قرار می گیرد.

### ۲-۲- معماری میکروسرویس

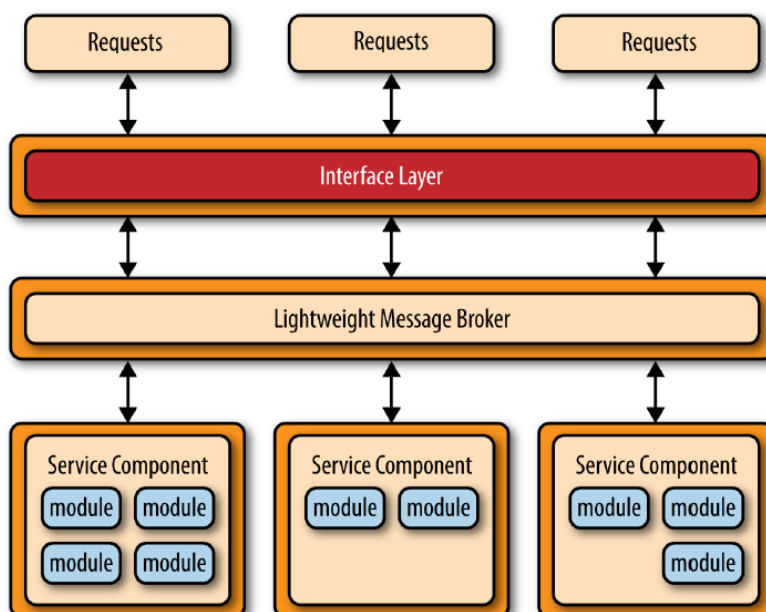
در حوزه معماری سیستم های نرم افزاری، الگوهای معماری متعدد و دسته بندی های متفاوتی وجود دارد. مهمترین معماری هایی که در حال حاضر استفاده می شوند شامل معماری های لایه ای<sup>۱</sup> (مانند MVC<sup>۲</sup>) Layered، مبتنی بر رویداد<sup>۳</sup>، میکرو کرنل، میکروسرویس و Space-Based می باشد. معماری میکروسرویس برخلاف روش های سنتی سامانه های یکپارچه، یک سیستم توزیع شده متشکل از چندین سرویس است که هر سرویس به صورت مجزا پیاده سازی شده و سرویس مربوطه را از طریق API در اختیار سایر سرویس ها و مشتریان قرار می دهد. این معماری در حقیقت سرعت و کارایی بالایی را در توسعه و تغییر سیستم فراهم می کند و معماری کلاسیک مبتنی بر سرویس را بهبود می دهد. شمای کلی این معماری در شکل ۱-۲ نشان داده شده است.

<sup>۱</sup> Layered (n-tier)

<sup>۲</sup> Model View Controller

<sup>۳</sup> Event Driven

نوع طبقه بندی سند: محرمانه	کد سند: ISRC-AUT-۹۷۰۱۲۱،۰	تاریخ: ۱۳۹۷/۰۱/۲۱	صفحه: ۱۰ از ۵۳
تمامی اطلاعات موجود در این سند متعلق به کارگروه پلتفرم، گروه پژوهشی اینترنت اشیا، دانشگاه صنعتی امیرکبیر بوده و حقوق قانونی آن محفوظ است.			



شکل ۱-۲ معماری میکروسرویس

در معماری میکروسرویس، کارکردهای سامانه در قالب سرویس هایی پیاده سازی می شوند که هر سرویس یک واحد مستقل است و همه موارد مورد نیاز جهت اجرای خود را دارد. علاوه بر آن می تواند در هر کجای شبکه استقرار پیدا کند. برای توزیع درخواست های داده شده به سامانه و همچنین ارتباطات بین سرویس ها عموماً از یک Message Broker در این معماری استفاده می شود. استفاده از این معماری برای پیاده سازی پلتفرم بومی عام منظوره پیشنهادی دارای چندین مزیت است. اول اینکه اکثر نیازمندی های کارکردی سامانه می توانند به صورت سرویس های مجزا در قالب این معماری پیاده سازی شوند که فرایند نگهداری و قابلیت اطمینان آن را راحت تر می کند چرا که به راحتی می توان تشخیص داد کدام واحد دچار خطا شده است. این معماری به دلیل توزیع شدگی به صورت ذاتی مقیاس پذیر بوده و امکان اضافه کردن قابلیت های غیرکارکردی به این معماری وجود دارد. همچنین در طراحی هایی که کامپوننت های مختلف وجود دارند و یا توسعه آن ها برون سپاری می گردد امکان توسعه آن ها به زبان های مختلف و در تیم های جداگانه به راحتی از طریق API های استاندارد فراهم می گردد.

## ۲-۳ - معماری پلتفرم مبتنی بر میکروسرویس

شمای کلی معماری ارائه شده برای پلتفرم اینترنت اشیا در شکل ۲-۲ نشان داده شده است. در این شکل اجزای مختلف پلتفرم، ارتباطات داخلی و ارتباطات خارجی آن مشخص شده است.



جدول ۱-۲. واحدها به همراه شرح وظایف

نام واحد	وظیفه	زبان / تکنولوژی
Data Analysis	این واحد برای اتصال به ابزارهای تحلیل داده مورد استفاده قرار می گیرد.	Java/Apache Spark
Web UI	این واحد واسطه های کاربری گرافیکی <sup>۱</sup> وب سایت پلتفرم را پیاده می کند (front-end).	HTML/CSS/JS
UI server	این واحد وظیفه پیاده سازی توابع واسطه های کاربری و ارتباط با API Server جهت فراخوانی واسطه های سمت سرور را بر عهده دارد (front-end).	ReactJS
API-Server	واسطه ها و توابع back-end سمت سرور توسط این واحد پیاده سازی می گردد. واسطه های فراهم شده هم توسط وب و هم توسط موبایل قابل استفاده و فراخوانی است.	Laravel (PHP Framework)
Project Manager	وظیفه مدیریت پروژه های ایجاد شده توسط کاربر را به عهده دارد و به هر کاربر یک container اختصاص می دهد.	Go
Data Manager	این واحد وظیفه پیاده سازی توابع مربوط به پرس و جوی های بر روی داده ها را بر عهده دارد.	Go
Containers	مجموعه ای از Container ها را شامل می گردد که هر Container وظیفه مدیریت سناریوها و کدک های آن کاربر را بر عهده دارد.	Python3/Go
Down-Link	این واحد وظیفه انتقال اطلاعات از پلتفرم به اشیاء را بر عهده دارد.	Go
Up-Link	این واحد وظیفه انتقال اطلاعات از اشیاء به پلتفرم را به عهده دارد.	Go
Connectivity	این واحد وظیفه مدیریت اتصال به gateway های پروتکل های ارتباطی با اشیاء را به عهده دارد.	Go
LoRaWAN Gateway	Gateway ارتباط با اشیاء را از طریق LoRa Server فراهم می کند.	Go
LAN Gateway	Gateway ارتباط با اشیاء را از طریق LAN Server فراهم می کند.	Python3
Orchestrator	این واحد وظیفه مدیریت تمامی واحدها را بر عهده دارد.	Kubernetes/ Rancher
Database	این واحد وظیفه مدیریت داده های پلتفرم را به عهده دارد.	MongoDB
Redis	به عنوان یک محل ذخیره سازی داده به صورت in memory، نقش یک واسطه را بین پایگاه داده و واحد Up-Link ایفا می کند.	Redis
Android App	یک برنامه کاربردی موبایل است که امکان اتصال مستقیم و استفاده از توابع ارائه شده توسط API Server را دارد.	--

در ادامه این مستند عملکرد واحدها به تفصیل مورد بحث و بررسی قرار خواهد گرفت.

<sup>۱</sup> GUI

### ۳- مولفه ها و فرآیندهای پلتفرم اشیا

#### ۳-۱- مقدمه

در این بخش ابتدا مولفه های درون پلتفرم اینترنت اشیا تشریح می گردد. پس از معرفی سرویس ها، فرآیندهای پایه ای که در راستای مدیریت اشیا و پلتفرم تعریف شده است مورد بررسی قرار می گیرد.

#### ۳-۲- مولفه API-Server

این مولفه به نوعی در قلب پلتفرم قرار دارد و وظیفه پیاده سازی توابع Backend وب را برعهده دارد. تمام توابع مرتبط به سمت کاربر و مدیریت پلتفرم در این واحد پیاده سازی شده اند. این مولفه از یک سمت با واحدهای Project Manager, Data Manager, Gateway Manager و در برخی موارد با Gateway های ارتباطی با اشیا مانند LoRaWAN در ارتباط است. در سمت دیگر نیز با واسطه های گرافیکی کاربران نهایی و مدیر پلتفرم ( Front-End ) و برنامه های کاربردی (مانند کاربرد سمت اندروید) در ارتباط است. این مولفه در این ارتباط، سرویس های لازم را فراهم می کند و در صورت لزوم برخی از سرویس ها را دریافت می کند. توابع ارائه شده توسط این بخش به دو دسته اصلی زیر تقسیم شده اند:

۱- توابع کاربران نهایی که برخی از مهمترین آنها عبارتند از

- احراز هویت شامل ورود به سیستم، ثبت نام و ویرایش کاربران
- مدیریت پروژه ها شامل ساخت پروژه، و به روزرسانی اطلاعات پروژه
- مدیریت اشیاء شامل ساخت شی، حذف شی، دریافت و ارسال داده، نوع اینترنتیسی ارتباطی (LAN و LoRaWAN)
- مدیریت پروفایل اشیاء شامل ساخت پروفایل، و بازیابی پروفایل
- مدیریت کدک شامل ساخت کدک، و بازیابی کدک
- مدیریت سناریوها شامل ساخت سناریو، به روزرسانی سناریو، بازیابی سناریو، و صحت سنجی سناریو
- مدیریت Gateway ها
- حسابرسی و پرداخت کاربران
- داشبورد کاربر
- ...

۲- توابع مدیریت پلتفرم که برخی از مهمترین آنها عبارتند از

- مدیریت کاربران و اطلاعات آنها

نوع طبقه بندی سند: محرمانه	کد سند: ۹۷۰۱۲۱۰۰-ISRC-AUT	تاریخ: ۱۳۹۷/۰۱/۲۱	صفحه: ۱۴ از ۵۳
تمامی اطلاعات موجود در این سند متعلق به کارگروه پلتفرم، گروه پژوهشی اینترنت اشیا، دانشگاه صنعتی امیرکبیر بوده و حقوق قانونی آن محفوظ است.			

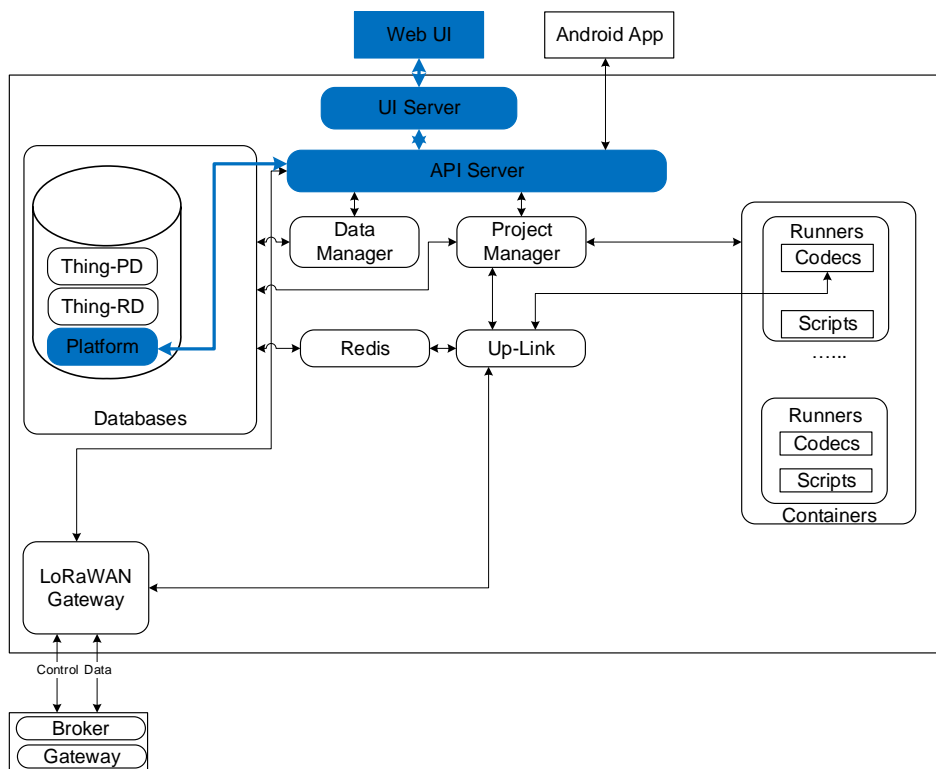
- مدیریت اجازه دسترسی ها و نقش ها
- مدیریت درگاه های ارتباطی
- مدیریت کانتینرهای (Containers) پروژه
- مدیریت درگاه های ارتباطی و وضعیت آن ها
- گزارش گیری
- مدیریت پرداخت
- ...

همه توابع ذکر شده، قابلیت ارائه سرویس به واسطه های گرافیکی تحت وب (Web UI , UI Server) و کاربردهای سمت اندروید (android App) را دارند.

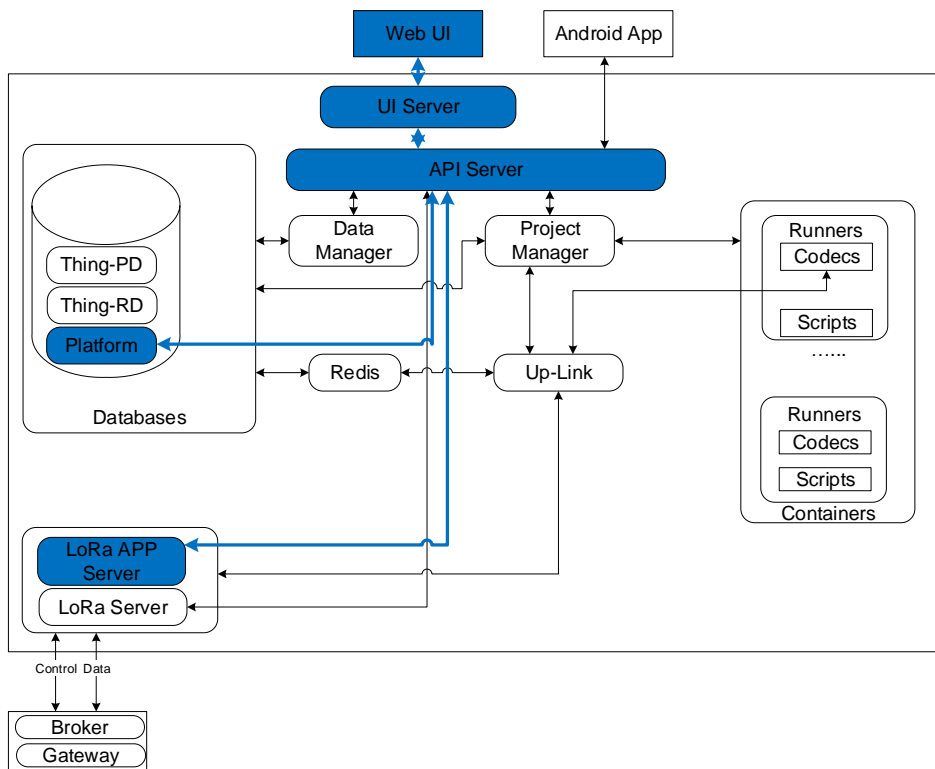
### ۳-۳- مولفه Web UI و UI Server

این مولفه ها همان طور که قبلا گفته شد بخش Front-end پلتفرم را تشکیل می دهند. که با توجه به جدا بودن Back-end، در نتیجه API های فراهم شده برای این بخش توسط سایر برنامه های کاربردی نیز قابل استفاده است. واحد Web UI ظاهر گرافیکی سایت و UI Server نیز توابع مربوط به واسطه گرافیکی سایت و فراخوانی API های مربوط به Back-end را برعهده دارد.

برای نمونه دو فرآیند اصلی ثبت نام کاربر و تعریف شی در شکل ۳-۱ و شکل ۳-۲ نمایش داده شده است. در شکل ها واحدهای درگیر در فرآیند نیز با رنگ آبی نشان داده شده اند. در سناریوی ثبت نام کاربر (شکل ۳-۱) ابتدا کاربر از واسطه کاربری خود گزینه ثبت نام را انتخاب می کند. پس از انتخاب کاربر توابع مورد نیاز برای ثبت اطلاعات از API-Server فراخوانی می شود و اطلاعات کاربر در پایگاه داده مربوط به کاربران ثبت می گردد. در سناریوی تعریف شی یا Activation از نوع اشیا Lora (شکل ۳-۲)، کاربر از طریق واسطه کاربری گزینه مرتبط به فعال سازی شی را انتخاب می کند. پس از انتخاب کاربر، توابع مرتبط از بخش API-Server فراخوانی می شود و اطلاعات شی مورد نظر پس از طی روال تایید آن در سمت LoRaWAN Gateway، در پایگاه داده ثبت می گردد. نیازمندی ها و پروسه ثبت نام اشیا به دو روش OTAA و ABP قبلا در سند تحلیل نیازمندی های کارکردی اینترنت اشیا - سرویس های پایه (IoT-RA-BS-v1,16) شرح داده شده است.



شکل ۳-۱. فرآیند ثبت نام کاربر



شکل ۳-۲. فرآیند تعریف شی

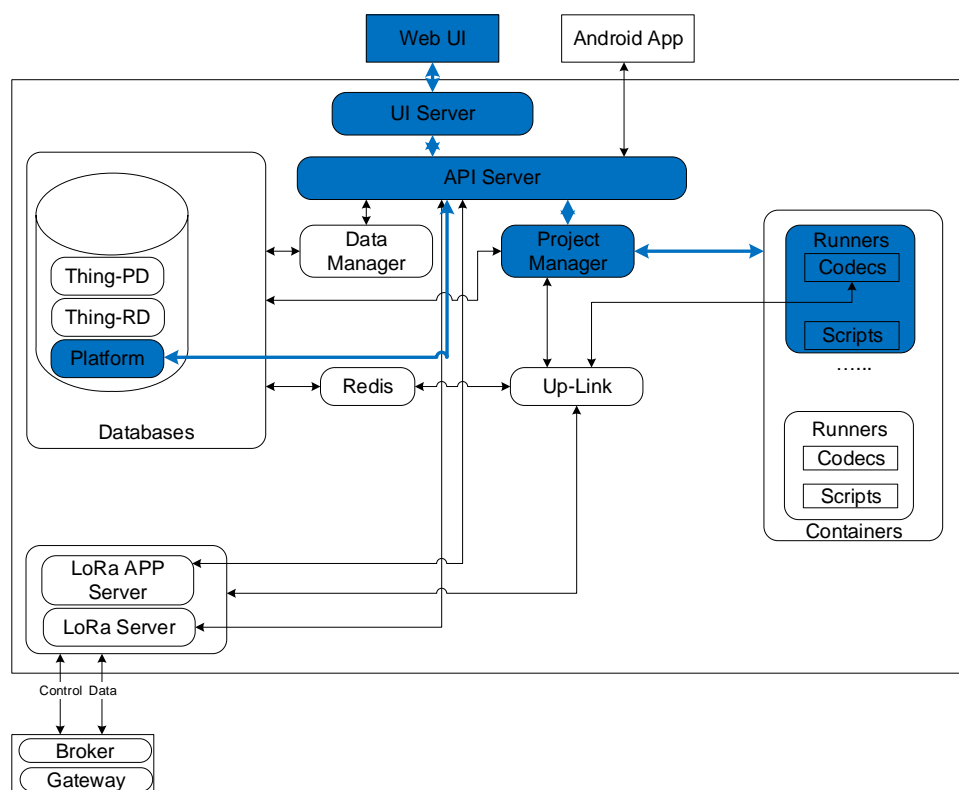
### ۳-۴- مؤلفه Project Manager

این واحد وظیفه مدیریت پروژه های تعریف شده کاربران را دارد که شامل ایجاد پروژه، فعال یا غیر فعال سازی پروژه می باشد. در

شکل ۳-۳ سناریوی تعریف پروژه نشان داده شده است و واحدهای مرتبط با رنگ آبی مشخص شده اند. در این سناریو کاربر از طریق منوی تعریف پروژه که در واسط کاربری فراهم شده است یک پروژه ایجاد می کند. پس از دریافت درخواست کاربر از طریق واسط کاربری، توابع مرتبط در سمت سرور فراخوانی می شود که این امر باعث فعال شدن واحدهای Project Manager، Runner و Database می شود و در نهایت پروژه تعریف می گردد. واحد Runners مجموعه ای از کانتینرها می باشد و در حقیقت سناریو و کدک های کاربران را مدیریت می کند و به ازای هر کاربر یک کانتینتر تعریف می گردد که کدهای کاربر در آن اجرا می گردد. در این صورت خرابی کدک یا سناریو یک کاربر تاثیری در کدک ها و سناریوهای سایر کاربران ندارد. با توجه به اینکه هر کاربر یک Container برای خود دارد در نتیجه حجم مورد استفاده آن به ازای هر کاربر، بر اساس تعداد کاربران و اندازه دیتاست در مراحل پایانی پروژه بررسی شده و ارائه می گردد. البته وضعیت این Runnerها نیز در پلتفرم مدیریتی در اختیار مدیر پلتفرم قرار خواهد گرفت. اطلاعات ایجاد شده نیز در پایگاه داده Platform، جهت استفاده های بعدی ذخیره می گردد.

نوع طبقه بندی سند: محرمانه	کد سند: ۹۷۰۱۲۱،۰-ISRC-AUT	تاریخ: ۱۳۹۷/۰۱/۲۱	صفحه: ۱۷ از ۵۳
تمامی اطلاعات موجود در این سند متعلق به کارگروه پلتفرم، گروه پژوهشی اینترنت اشیا، دانشگاه صنعتی امیرکبیر بوده و حقوق قانونی آن محفوظ است.			

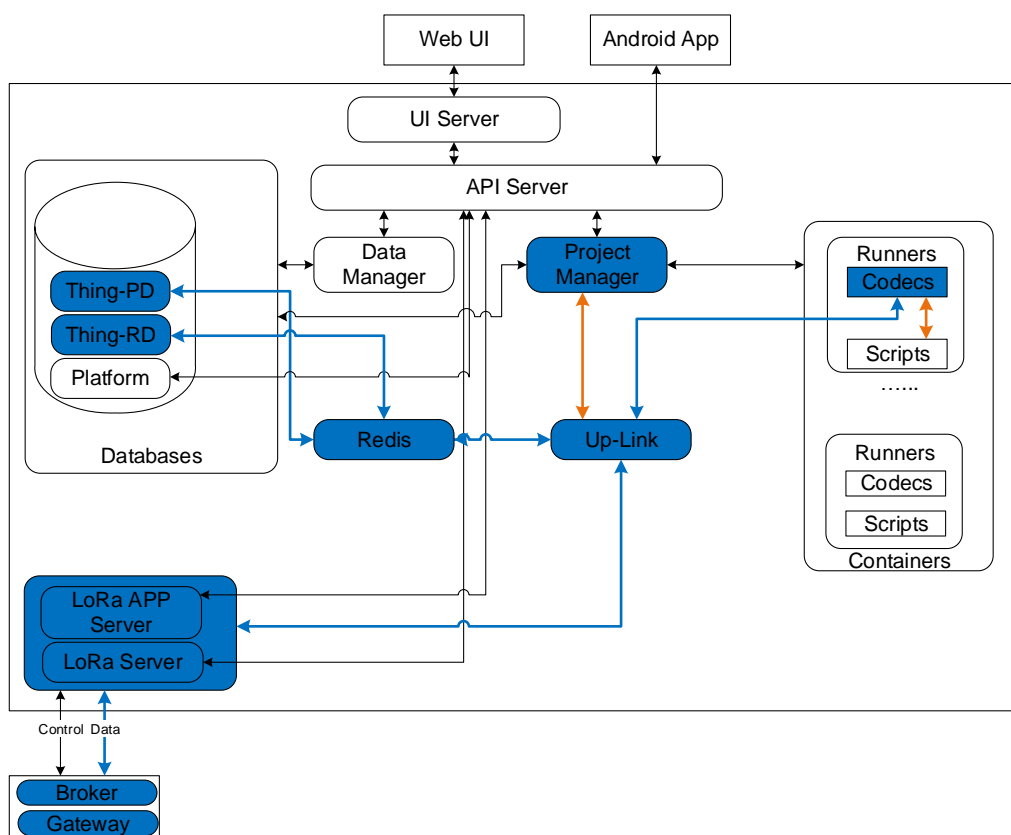




شکل ۳-۳. سناریوی تعریف پروژه

### ۳-۵ - مولفه Up-link

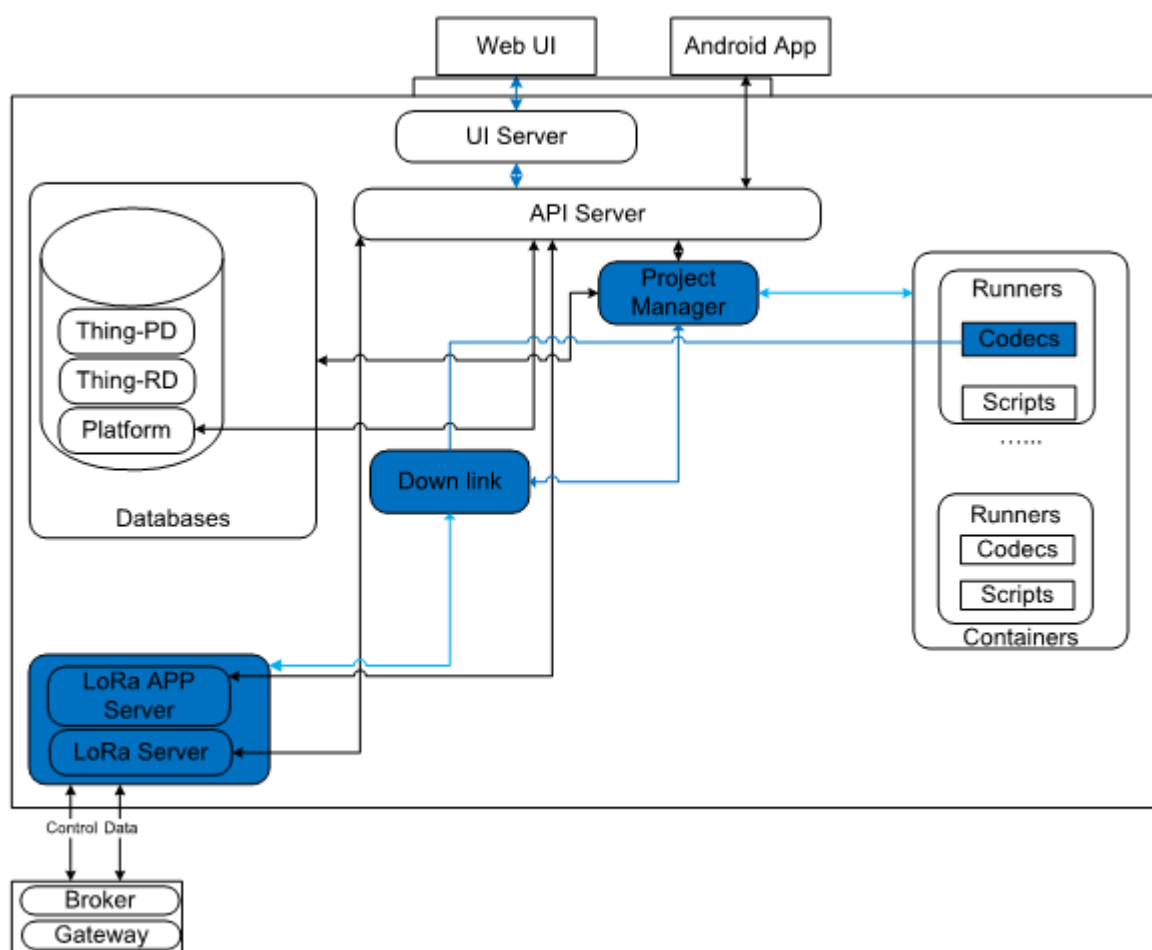
این مولفه وظیفه انتقال اطلاعات از شی به پلتفرم را به عهده دارد. در این بخش داده ها به دو صورت پارس شده و خام در پایگاه داده قرار می گیرند. در شکل ۳-۴ سناریوی دریافت داده نشان داده شده است و واحدهای مرتبط با رنگ آبی نشان داده شده اند. اطلاعات خام دریافتی از سمت LoRaGateway از طریق uplink به سمت پایگاه داده ارسال می گردد. برای افزایش سرعت نیز از Redis به عنوان یک منبع ذخیره سازی ساختار داده in-memory استفاده شده است که توانایی پاسخ گویی به تعدادی زیادی از داده های دریافت شده در کسری از میلی ثانیه را داراست در نتیجه امکان Lost داده های دریافتی از بین می رود. برای داده های پارس شده نیز، از کدکی که برای اشیا از سمت کاربران تعریف شده است استفاده می گردد و اطلاعات اشیا در ابتدا پردازش و سپس جهت ذخیره سازی به پایگاه داده ارسال می گردد. در این حالت، واحد Up-Link اطلاعات مربوط به کدک کاربران را از Runner یا کانتینر مربوطه دریافت می کند. اینکه کدام Runner، کدک مربوطه را داراست از طریق Project Manager مشخص می شود. سپس اطلاعات درون شی را استخراج می کند و در پایگاه داده ذخیره می کند.



شکل ۳-۴. سناریوی دریافت داده

### ۳-۶- مولفه Down-link

این بخش وظیفه انتقال داده از پلتفرم به اشیاء را به عهده دارد. روال آن در شکل ۳-۵ آمده است. واحد Down-link پس از دریافت اطلاعات از سمت کاربر، از طریق Project Manager، کدک های مرتبط شامل encoding مربوطه به آن شی را پیدا کرده، داده مد نظر را با استفاده از encoder مربوطه کد کرد و تبدیل به یک bit stream می کند که می تواند به شی مربوطه ارسال گردد. با توجه به شناسه شی، داده شده، اطلاعات از طریق Downlink به سمت LoRa Gateway ارسال می گردد.

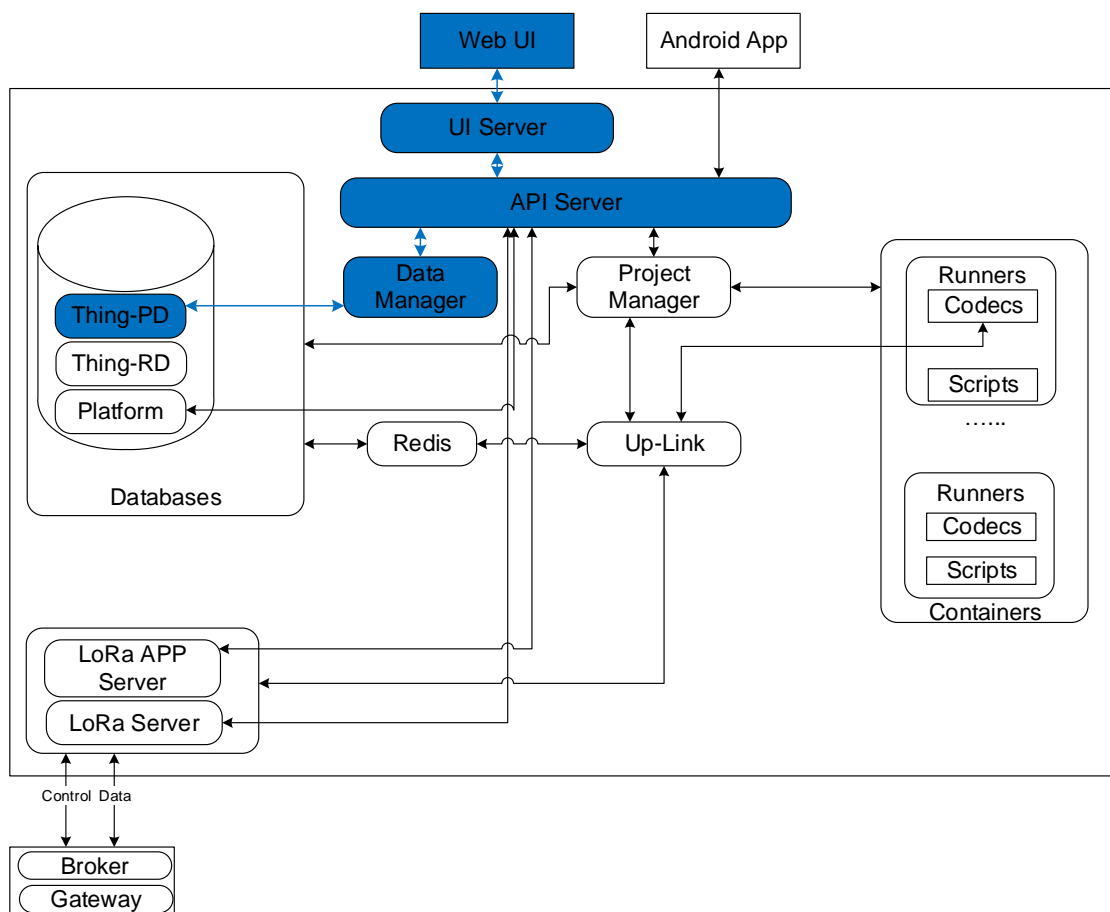


شکل ۳-۵. سناریوی ارسال داده

### ۳-۷- مولفه Data Manager

این واحد وظیفه پیاده سازی توابع مرتبط به عملیات پس و جوی (query) کاربران بر روی داده های پایگاه داده پارس شده را بر عهده دارد. با توجه به اینکه درخواست های متعددی از واسط های گرافیکی کاربر برای نمایش داده به صورت نمودار و جدولی و جزئیات نمایش آن ها شامل Aliasing ارسال می گردد مدیریت و نحوه پاسخگویی به این درخواست ها در این مولفه قرار داده شده است.

در شکل ۳-۶ فرآیند نمایش داده ترسیم شده است که واحدهای مرتبط با رنگ آبی نشان داده شده اند. در این سناریو ابتدا کاربر از طریق واسط کاربری گزینه نمایش داده های شی را انتخاب می کند. انتخاب کاربر در سمت سرور باعث فراخوانی توابع مرتبط می شود. در ادامه توابع بازیابی از واحد Data Manager صدا زده می شود و در نهایت اطلاعات مورد نظر بازیابی می گردد و به کاربر نمایش داده می شود.



شکل ۳-۶. سناریوی نمایش داده

### ۳-۸- مولفه Orchestrator

همان طور که در شکل ۲-۲ نیز نشان داده شد، این واحد با تمام واحدهای دیگر در ارتباط است و به نوع مدیریت این واحدها را برعهده دارد. در اینجا از kubernetes که یک بستر برای مدیریت و هماهنگ سازی کانتینرها می باشد استفاده می گردد. این بستر اپلیکشن های شما را که ممکن است از یک یا تعدادی کانتینر تشکیل شده باشد اجرا و مدیریت می کند و نیازمندی های ارتباطی آن ها را فراهم می آورد. در این بین کاربر می تواند هر یک از کانتینرهایی دلخواه خود را گسترش داده و تعداد نمونه های آن را زیاد یا کم کند. کانتینرها ممکن است نیاز به حافظه هایی جهت ذخیره ی دائمی داده ها داشته باشند که به آن ها storage گفته می شود، کوبرنتیس storage هر یک از کانتینرها را نیز مدیریت می کند.

کوبرنتیس سرویس های Discovery نیز برای اپلیکشن ها فراهم می آورد که از طریق آن می توانند با یکدیگر ارتباط برقرار کنند. در صورت خرابی یا fail شدن یک کانتینر، کوبرنتیس این موضوع را تشخیص داده و می تواند آن کانتینر را باز اجرا کند.

کوبرنتیس برای پشتیبانی از کانتینرها از موتورهای Docker و RKT پشتیبانی می کند که پشتیبانی از موتور RKT هنوز در فاز اولیه قرار دارد.

نوع طبقه بندی سند: محرمانه	کد سند: ۹۷۰۱۲۱،۰-ISRC-AUT	تاریخ: ۱۳۹۷/۰۱/۲۱	صفحه: ۲۲ از ۵۳
تمامی اطلاعات موجود در این سند متعلق به کارگروه پلتفرم، گروه پژوهشی اینترنت اشیا، دانشگاه صنعتی امیرکبیر بوده و حقوق قانونی آن محفوظ است.			

## ۴- پروتکل های ارتباطی با اشیاء

### ۴-۱- مقدمه

ارتباط با اشیاء یکی از نیازمندی های اصلی پلتفرم پیشنهادی است. پروتکل های متفاوتی برای این امر وجود دارد. در معماری ارائه شده چهار پروتکل زیر قرار دارند :

۱. پروتکل LoRaWAN

۲. پروتکل ارتباط IP از طریق LAN

۳. پروتکل خاص منظوره کارفرما

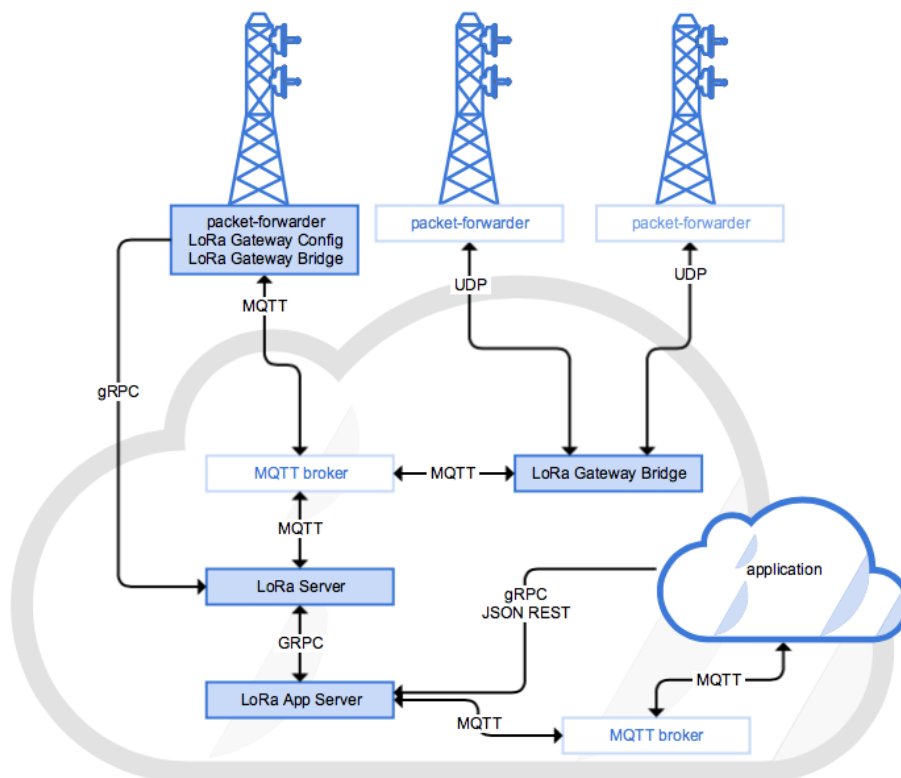
۴. پروتکل NB-IoT

با توجه به اولویت بندی انجام شده و نیازمندی های کارفرما در حال حاضر دو مدل ارتباطی یکی بر مبنای LoRaWAN و دیگری LAN در مرحله طراحی و پیاده سازی قرار گرفته اند. تشریح این پروتکل ها در ادامه آمده است.

### ۴-۲- پروتکل LoRaWAN

در این بخش برای ارتباط با اشیاء LoRa از پروژه متن باز LoraServer.io استفاده شد. این پروژه مجموعه ای از برنامه های کاربردی متن باز است که امکان ارتباط بین درگاه های دریافت کننده پیام از اشیاء LoRa و ابزارها یا پلتفرم های مدیریتی را در اینترنت اشیاء فراهم می کند. LoRaWAN یک پروتکل در سطح لایه Mac است که ارتباطات بیسیم را فراهم می کند. این پروتکل ارتباط بین اجزا را در توان پایین ولی با برد بالا فراهم می کند. در شکل ۴-۱ معماری پروژه متن باز LoraServer نمایش داده شده است.

نوع طبقه بندی سند: محرمانه	کد سند: ۹۷۰۱۲۱۰۰-ISRC-AUT	تاریخ: ۱۳۹۷/۰۱/۲۱	صفحه: ۲۳ از ۵۳
تمامی اطلاعات موجود در این سند متعلق به کارگروه پلتفرم، گروه پژوهشی اینترنت اشیاء، دانشگاه صنعتی امیرکبیر بوده و حقوق قانونی آن محفوظ است.			



شکل ۴-۱. معماری مدل ارتباطی مبتنی بر LoRaServer

معماری ارائه شده توسط LoRaServer از مولفه‌های زیر تشکیل شده است.

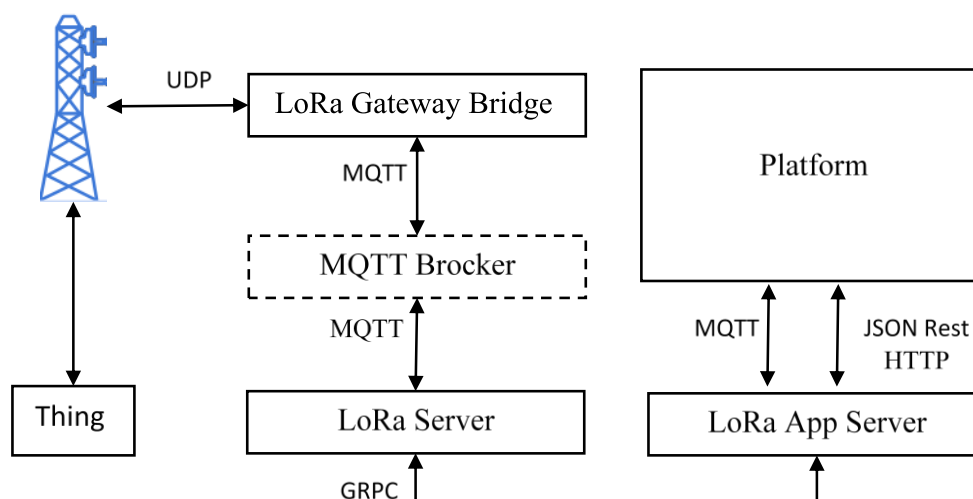
- LoRa nodes: اشیایی که داده‌ها را از طریق درگاه ارتباطی به شبکه ارسال میکنند. این دستگاه‌ها می‌توانند برای مثال حسگرهای اندازه‌گیری کیفیت هوا، دما و رطوبت باشند.
- LoRa Gateway: این مولفه وظیفه دریافت داده از اشیاء را بر عهده دارد. درگاه‌ها معمولاً از نرم‌افزارهای forwarder بسته‌ها استفاده میکنند. در برخی موارد همان‌طور که در شکل نیز نشان داده شده است امکان دارد این مولفه LoRa Gateway Bridge را نیز در خود داشته باشد که ارتباطات را از طریق MQTT به LoRa Server ارسال می‌کند.
- LoRa Gateway Bridge: این مولفه مسئول برقراری ارتباط با درگاه ارتباطی است. این مولفه پکت‌های UDP دریافتی از Gateway را به فرمت JSON بر روی MQTT می‌برد. این مولفه مزایای مانند تامین امنیت، آسان‌سازی اشکال‌یابی و ... را داراست.
- LoRa Server: مسئول هماهنگی تمام مولفه‌ها در شبکه است. این واحد از اشیاء فعال در شبکه مطلع است. هنگامی که یک شیء جدید به شبکه متصل می‌شود این واحد از App server در مورد وضعیت شیء، اطلاعات

نوع طبقه‌بندی سند: محرمانه	کد سند: ۹۷۰۱۲۱،۰-ISRC-AUT	تاریخ: ۱۳۹۷/۰۱/۲۱	صفحه: ۲۴ از ۵۳
تمامی اطلاعات موجود در این سند متعلق به کارگروه پلتفرم، گروه پژوهشی اینترنت اشیا، دانشگاه صنعتی امیرکبیر بوده و حقوق قانونی آن محفوظ است.			

می گیرد که در صورت لزوم آن دستگاه را در سیستم ثبت کند. موارد دیگر از جمله مدیریت عمل de-duplicat (حذف داده های تکراری) داده های دریافتی از سمت چند Gateway، اعتبار سنجی این داده ها و ارسال آن به سمت Application-Server و در صورت لزوم دریافت پیام برگشتی از آن را برعهده دارد.

- **LoRa App Server:** این واحد مسئول پیاده سازی کاربرد هاست به شکلی که با LoRa Server هماهنگ باشند. قابلیت های مدیریت اشیا به ازای هر کاربرد و سازمان و مدیریت گذرگاه به ازای هر سازمان را در اختیار قرار میدهد. سایر قابلیت های مربوط به مدیریت کاربران و ارتباط آن با کاربردها را نیز فراهم می کند. ارتباط با کاربردها از طریق JSON بر روی MQTT فراهم خواهد شد. API مناسبی جهت توسعه کاربردها در پروژه LoRa تدارک دیده شده است.
- **Application:** جایگاه کاربردها را نمایش می دهد. هر کاربرد از طریق یک عنوان و با استفاده از MQTT داده ها را از دستگاه ها جمع آوری می کند. برنامه کاربردی حتی قادر به ارسال داده از طریق MQTT نیز هست. در صورت لزوم نیز توانایی ارتباط API از gRPC یا JSON Rest را دارد.

بخش های ذکر شده از پروژه LoRa شامل LoRa Gateway Bridge، MQTT Broker، Lora Gateway Bridge و LoRa Server و APP Server در پلتفرم اینترنت اشیا نیز مورد استفاده قرار گرفته است. در حقیقت در حال حاضر بخشی از پلتفرم اینترنت اشیا می باشد. در ادامه جایگاه پلتفرم ارائه شده در تعامل با مولفه های ذکر شده مورد بررسی قرار گرفته است.



شکل ۴-۲. ارتباط اشیا با پلتفرم از طریق Loraserver.io

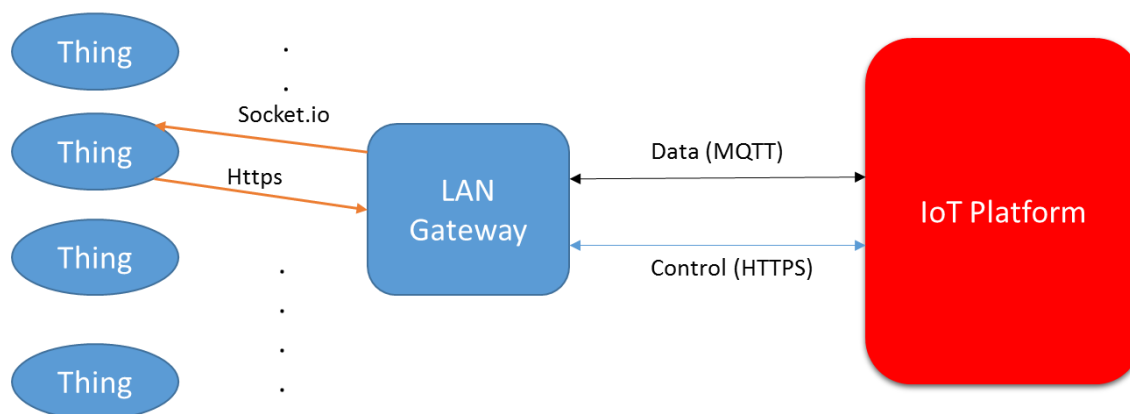


همان طور که در شکل ۲-۴ نشان داده شده است بخش LoRa App Server مسئول تعامل با پلتفرم است. در حال حاضر در معماری ارائه شده MQTT برای انتقال داده و ارتباط مبتنی بر HTTP به عنوان کنترلی استفاده می گردد.

اطلاعات مربوط به تنظیمات کلی پروتکل، تنظیمات شبکه، پروتکلی کنترلی و داده مابین پلتفرم و سرور LoRa و سایر موارد در مستند (IoT-RA-BS-v۱.۱۶) ذکر شده است.

## ۳-۴- پروتکل LAN

دیگر پروتکل ارتباطی با اشیا پروتکل LAN (خانواده IEEE ۸۰۲ و مشخصا IEEE ۸۰۲.۳، IEEE ۸۰۲.۱۱ و IEEE ۸۰۲.۱۵) می باشد. شمای کلی ارتباطات شکل ۳-۴ نمایش داده شده است.



شکل ۳-۴. معماری گذرگاه LAN جهت ارتباط با اشیا

لازم به ذکر است که این پروتکل علاوه بر پلتفرم، بر روی اشیا نیز باید پیاده سازی گردد، بنابراین استفاده از پروتکل های استاندارد موجود برای این منظور (در عمل) الزامی است. ساختار کلی طراحی در سه بخش زیر آمده است:

۱- اشیا برای اتصال به Lan Gateway در آدرس '/login' با استفاده از token وارد می شوند. برای ارسال داده از اشیا به پلتفرم، یک درخواست POST با استفاده از HTTPS به Lan Gateway ارسال می شود. سپس این داده ها برای ارسال به پلتفرم روی کانال MQTT گذاشته می شوند. درخواست POST ارسال شده از شیء به Lan Gateway به آدرس '/data' و با فرمت JSON است که شامل token و data است. این مرحله قابلیت اتصال اشیا به صورت امن را فراهم می کند.

۲- برای ارسال داده از پلتفرم به اشیا، داده ها از پلتفرم با استفاده از کانال MQTT به Lan Gateway ارسال می شوند. Lan Gateway که در حال گوش دادن به این کانال است با دریافت داده با استفاده از

نوع طبقه بندی سند: محرمانه	کد سند: ISRC-AUT-۹۷۰۱۲۱،۰	تاریخ: ۱۳۹۷/۰۱/۲۱	صفحه: ۲۶ از ۵۳
تمامی اطلاعات موجود در این سند متعلق به کارگروه پلتفرم، گروه پژوهشی اینترنت اشیا، دانشگاه صنعتی امیرکبیر بوده و حقوق قانونی آن محفوظ است.			

Socket.io داده ها را برای شیء ارسال می کند. Lan Gateway عنوان و داده ها را به صورت JSON دریافت می کند و به اشیاء ارسال می کند. این قابلیت نیز برآورده کردن امکان ارسال داده از اشیاء به پلتفرم را بر اساس نیازمندی ها تامین کند.

۳- بر روی Lan Gateway یک API با استفاده از HTTP وجود دارد تا دستورات کنترلی به آن ارسال گردد. یکی از دستورات کنترلی، دستور اضافه کردن اطلاعات یک شیء به پایگاه داده ی بر روی Lan Gateway است. این API دارای آدرس '/control' و نوع درخواست POST است که حاوی type (نوع درخواست کنترلی) و data (داده های مربوط به درخواست) است. با ارسال این درخواست توکن مربوط به شیء جدید با استفاده از JWT تولید و برگردانده می شود. این قابلیت نیز امکان ارسال داده کنترلی به اشیاء از طریق پلتفرم را فراهم می کند.

#### ۴-۴- واحدهای مرتبط به انتخاب پروتکل ارتباطی با اشیاء

همانگونه که در شکل ۲-۲ نشان داده شده است برای ارتباط با درگاه ها واحد connectivity در پلتفرم طراحی شده است که وظیفه مدیریت اتصال به درگاه ها را دارد. به عنوان مثال این واحد با استفاده از اطلاعاتی که از درگاه ها دارد و پروتکل ارتباطی مورد استفاده، تبادلات اشیاء مربوط به هر درگاه را از دیگر واحدها به آن درگاه هدایت می کند.

نوع طبقه بندی سند: محرمانه	کد سند: ۹۷۰۱۲۱،۰-ISRC-AUT	تاریخ: ۱۳۹۷/۰۱/۲۱	صفحه: ۲۷ از ۵۳
تمامی اطلاعات موجود در این سند متعلق به کارگروه پلتفرم، گروه پژوهشی اینترنت اشیا، دانشگاه صنعتی امیرکبیر بوده و حقوق قانونی آن محفوظ است.			

## ۵- پروتکل ارتباطی با برنامه های کاربردی

### ۵-۱- مقدمه

یکی از قابلیت هایی که پلتفرم در اختیار قرار می دهد، امکان توسعه برنامه های کاربردی است که می توانند از امکانات پلتفرم برای دسترسی به اشیاء و کار با پلتفرم استفاده کنند. همان طور که قبلا هم توضیح داده شد، این API های از طریق واحد API Server فراهم می گردد. هر کاربر با توجه به نیاز خود می تواند برنامه خاص خود را توسعه دهد. پلتفرم پیشنهادی واسطه های برنامه نویسی که در قالب یک سری پروتکل پیاده سازی می شوند را در اختیار این برنامه های کاربردی قرار می دهد. به عنوان مثال امکان توسعه یک برنامه کاربردی خارج از پلتفرم با قابلیت اتصال پلتفرم برای مدیریت وجود خواهد داشت.

### ۵-۲- کلیات پروتکل

کلیات پروتکل پیشنهادی مبتنی بر معماری REST است که در واسط کاربری نیز استفاده شده است. REST مخفف عبارت Representational State Transfer است و متکی بر یک پروتکل ارتباطی بدون حالت<sup>۱</sup>، کلاینت/سرور<sup>۲</sup> و با قابلیت cache کردن می باشد. در این معماری از HTTP برای تعامل بین واحدها استفاده می شود. ایده اصلی معماری REST این است که به جای استفاده از مکانیزم های پیچیده ای مانند CORBA، RPC یا SOAP برای اتصال ماشین ها از HTTP ساده برای برقراری ارتباط بین ماشین ها استفاده شود. از لحاظ رویکرد برنامه نویسی REST جایگزینی ساده برای سرویس های وب است. توسعه پذیری در تعاملات میان اجزاء، عمومیت واسط ها، توسعه مستقل اجزاء و استفاده از واسطه ها از کلیدی ترین اهداف معماری REST می باشد. لازم به ذکر است که استفاده از معماری REST در برنامه نویسی کارایی، سادگی، انعطاف پذیری و قابلیت اطمینان را افزایش می دهد.

### ۵-۳- جزئیات API مربوط به برنامه های کاربردی

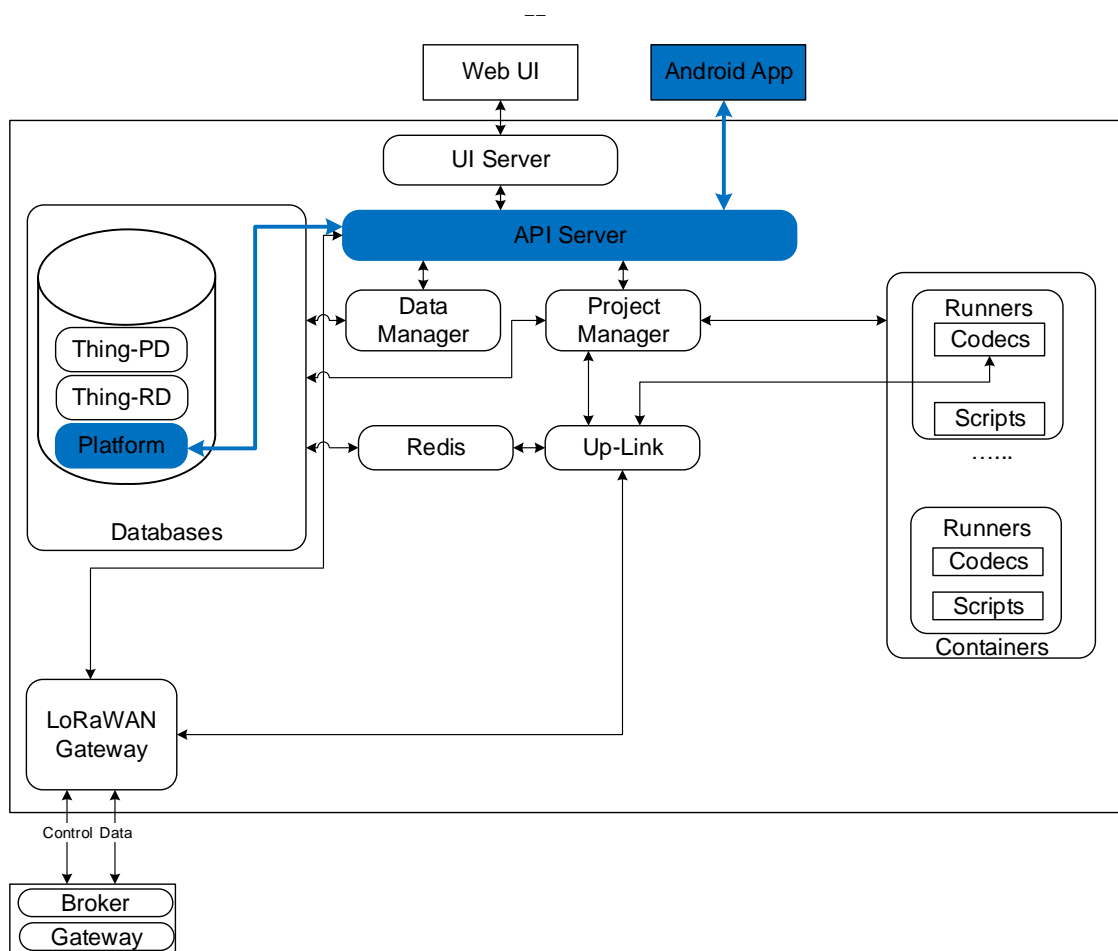
یکی از ویژگی های API ارائه شده این است که وابستگی به انواع ابزار هایی که از پلتفرم استفاده میکنند ندارد. توابع API به زبان PHP نوشته شده اند و امکان دسترسی به تمامی توابع ارائه شده توسط API-Server که در بخش ۲-۳ ارائه شده است توسط برنامه های کاربردی به صورت مستقیم وجود دارد. به عنوان مثال سناریوی ثبت نام

<sup>۱</sup> Stateless

<sup>۲</sup> Client/Server

نوع طبقه بندی سند: محرمانه	کد سند: ISRC-AUT-۹۷۰۱۲۱،۰	تاریخ: ۱۳۹۷/۰۱/۲۱	صفحه: ۲۸ از ۵۳
تمامی اطلاعات موجود در این سند متعلق به کارگروه پلتفرم، گروه پژوهشی اینترنت اشیا، دانشگاه صنعتی امیرکبیر بوده و حقوق قانونی آن محفوظ است.			

کاربر از طریق اپلیکیشن اندروید در شکل ۵-۱ نشان داده شده است. در این سناریو کاربر از طریق اپلیکیشن موجود در موبایل خود گزینه ثبت نام را انتخاب می کند. در ادامه توابع مربوط به ثبت نام از بخش API-Server فعال می گردند و اطلاعات کاربر در سیستم ثبت می گردد.



شکل ۵-۱. سناریوی ثبت نام کاربر از طریق اپلیکیشن اندروید

## ۶- پروتکل ارتباطی با ابزارهای تحلیل داده

### ۶-۱- مقدمه

با توجه به بررسی‌های انجام شده در سند تحلیل نیازمندی‌ها (IoT-RA-DP-v۱,۰)، سامانه باید قابلیت اتصال به ابزارهای تحلیل داده را فراهم کند. در حقیقت در پلتفرم‌های اینترنت اشیا با توجه به حجم داده زیادی که از اشیا دریافت می‌کنند امکان اتصال به ابزار تحلیل داده بسیار حائز اهمیت می‌باشد. این سرویس در واقع محل اتصال پلتفرم اینترنت اشیا به ابزارهای تحلیل داده است که داده‌های جمع‌آوری شده را در اختیار ابزار تحلیل داده قرار می‌دهد.

### ۶-۲- اتصال به ابزارهای داده

در معماری ارائه شده برای اینترنت اشیا که در شکل ۲-۲ نشان داده شده است واحد Data Analyses مسئول پیاده‌سازی سرویس اتصال به ابزار تحلیل داده است. با توجه به ارزیابی صورت گرفته در تحلیل نیازمندی‌ها، ابزار Spark برای تحلیل داده در نظر گرفته شد که از زبان Java استفاده می‌کند. ابزار انتخاب شده برای پردازش داده‌های با مقیاس بزرگ مناسب می‌باشد. این واحد از سه API به نام‌های SparkTextRead, SparSQLDataRead و SparkMongoDataRead تشکیل شده است.

SparkTextRead جهت تعامل با منابع داده‌ای متنی در نظر گرفته شده است. در این بخش، برنامه‌های طراحی شده با استفاده از کتابخانه‌های مختلفی مانند map و reduce که توسط ابزار spark فراهم می‌گردد، می‌تواند پردازش‌ها و عملیات مختلفی را بر روی داده‌های منابع متنی انجام دهند. بر حسب نوع داده مورد نیاز، API‌های مختلفی را می‌توان در این قسمت جهت ارتباط با پلتفرم قرار داد.

SparSQLDataRead نیز جهت تعامل با پایگاه داده‌های ساخت یافته پیاده‌سازی شده است. در صورتی که پلتفرم در آینده نیاز به پایگاه داده‌های از نوع SQL داشته باشد از API‌های توسعه یافته در این بخش می‌توان استفاده کرد. جهت اتصال به پایگاه داده SQL نیز از توابع load/save یا توابع jdbc استفاده می‌شود و سپس می‌توان از توابع مختلف ابزار spark استفاده کرد. در حال حاضر نیز اتصالات مربوط به پایگاه داده PostgreSQL فراهم شده است.

SparkMongoDataRead نیز برای اتصال به پایگاه داده‌ی Mongo از رابط پیشنهادی آن برای زبان جاوا استفاده می‌کند. بدین ترتیب می‌توان داده‌های جمع‌آوری شده در پایگاه داده‌ی پلتفرم را بازیابی کرد. برای اتصال به پایگاه داده از کلاس MongoClient استفاده می‌گردد. پس از بازیابی داده‌ها، مانند خواندن از پایگاه داده‌های ساخت یافته عمل می‌کنیم و پس از آن می‌توان از توابع مختلف ابزار spark استفاده کرد.

## ۷- حسابرسی

### ۷-۱- مقدمه

پلتفرم ارائه شده قابلیت ارائه سرویس جهت انجام امور تراکنش ها و مدیریت پرداخت کاربران را دارد. برای استفاده رایگان نیز، اکانت یک یا دو ماهه برای یک یا دو سنسور ایجاد گردیده است. با توجه به اینکه پکیج انتخابی به صورت تعداد سنسور در ماه می باشد نوع پرداخت به شکل prepaid می باشد. در ادامه نوع سرویس های فراهم شده در این بخش مورد بررسی قرار خواهد گرفت

### ۷-۲- سرویس های ارائه شده در این بخش

سرویس های ارائه شده در این بخش دو کاربر نهایی و مدیر پلتفرم را پوشش می دهد.

#### • سمت کاربر

- خرید بسته توسط کاربر با امکان انتخاب یکی از درگاه های بانکی
- امکان مشاهده و پرداخت فاکتورهای تخفیفی تعریف شده از سمت ادمین در بخش فاکتورهای پرداخت نشده
- مشاهده وضعیت بسته های خریداری شده
- مشاهده تراکنش های انجام شده

#### • مدیریت بسته ها در سمت مدیر پلتفرم (ادمین)

- نمایش کل تراکنش های موفق و ناموفق به ازای کل پلتفرم و به ازای هر کاربر
- تعریف بسته های جدید
- تعریف تخفیف برای کاربران (صدور فاکتور)
- فعال و غیر فعال کردن درگاه های پرداخت تعریف شده
- محاسبه ی قیمت پیشنهادی برای بسته ها بر اساس فرمول از پیش تعیین شده
- مشاهده بسته های تعریف شده در سیستم

### ۷-۳- فرایند خرید و پرداخت

نحوه ارتباط با سیستم های بانکی و پرداخت از طریق درگاه بانکی است. با توجه به زمانبر بودن دریافت نماد الکترونیکی، در حال حاضر عمل پرداخت به صورت ساختگی از طریق درگاه زرین پال فراهم شده و تست گردیده است. به هنگام تحویل به کارفرما و فراهم شدن نماد الکترونیکی با توجه به درگاه های انتخابی از سمت کاربر این بخش تکمیل خواهد شد و جزئیات آن در اختیار کارفرما قرار خواهد گرفت

نوع طبقه بندی سند: محرمانه	کد سند: ۹۷۰۱۲۱،۰-ISRC-AUT	تاریخ: ۱۳۹۷/۰۱/۲۱	صفحه: ۳۲ از ۵۳
تمامی اطلاعات موجود در این سند متعلق به کارگروه پلتفرم، گروه پژوهشی اینترنت اشیا، دانشگاه صنعتی امیرکبیر بوده و حقوق قانونی آن محفوظ است.			

## ۸- پایگاه داده

### ۸-۱- مقدمه

پایگاه داده MongoDB در پلتفرم استفاده شده است. پایگاه داده MongoDB را می توان یکی از پرمخاطب ترین پایگاه های داده موجود در جمع اعضای خانواده NoSQL دانست. این پایگاه داده یک مدل منعطف، پویا و سندگرا را ارائه می کند که ساختاری با خروجی بسیار بالا و قابلیت مقیاس پذیری آسان را دارا است. این نوع پایگاه داده در پروژه ها و سیستم هایی که با حجم بزرگی از داده ها مواجه هستند مورد نیاز است. بدلیل متغیر بودن ساختار داده ها و حجم زیاد داده ها در پلتفرم اینترنت اشیا، یک پایگاه داده ی NoSQL که در آن از جدول بندی داده ها استفاده نمی شود، باید استفاده گردد. به همین دلیل یکی از مناسب ترین گزینه ها استفاده از MongoDB می باشد. در این بخش مفاهیم مطرح، طراحی پایگاه داده و همچنین پایگاه داده های استفاده شده در سیستم تعریف می گردند.

### ۸-۲- خوشه بندی در پایگاه داده MongoDB

در پایگاه داده MongoDB می توان داده ها را خوشه بندی کرد، به این معنا که همزمان چندین پایگاه داده را راه اندازی کرده و سپس با سیستم Clustering آن ها را مدیریت کرد. این قابلیت مزایای زیر را دارد:

- سرعت خواندن و نوشتن بیشتر
- استفاده بهینه تر از فضا
- خواندن و نوشتن به صورت Partial

در این پایگاه داده دو نوع خوشه بندی وجود دارد:

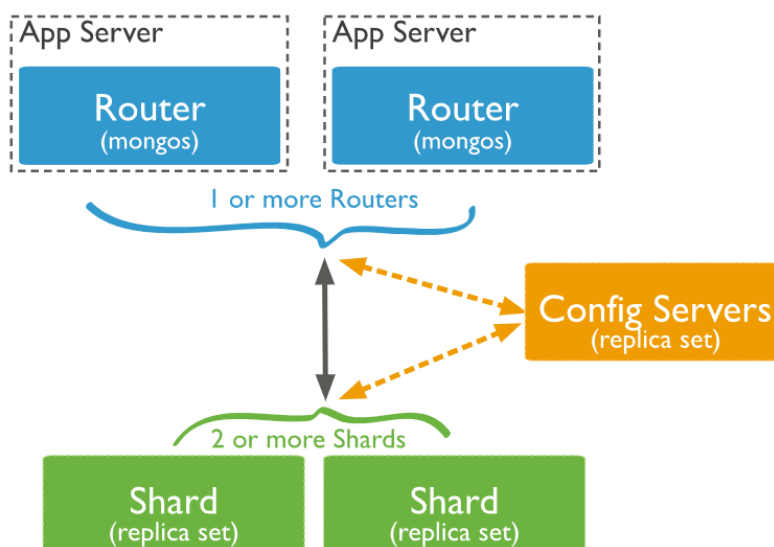
- Hashed Sharding: در این نوع خوشه بندی، MongoDB داده ها را با روش Hash کردن، میان چندین پایگاه داده تقسیم می کند. در این حالت اگر یکی از پایگاه داده ها از سرویس خارج گردید و داده های آن از بین رفت، بقیه داده ها قابل دسترس هستند.
- Ranged Sharding: در خوشه بندی مرتب سازی شده، یک داده به قسمت های مختلفی تقسیم شده و در چندین پایگاه داده ذخیره می گردد.

با توجه به طراحی فعلی و نوع استفاده از داده های پلتفرم در طراحی از نوع Hashed Sharding استفاده شده است.

معماری ارائه شده در شکل ۸-۱ ساختار خوشه بندی ذکر شده را نمایش می دهد.

نوع طبقه بندی سند: محرمانه	کد سند: ISRC-AUT-۹۷۰۱۲۱،۰	تاریخ: ۱۳۹۷/۰۱/۲۱	صفحه: ۳۳ از ۵۳
تمامی اطلاعات موجود در این سند متعلق به کارگروه پلتفرم، گروه پژوهشی اینترنت اشیا، دانشگاه صنعتی امیرکبیر بوده و حقوق قانونی آن محفوظ است.			





شکل ۸-۱. معمای خوشه ای

- config: پایگاه داده مبدا را که بقیه MongoDBها به آن متصل می شوند را Config یا همان Configuration می نامند که تنظیمات مربوط به Clustering را در خود ذخیره می کند.
- Shard: پایگاه داده هایی را که به Config متصل می شوند و داده ها را به صورت خوشه بندی شده در خود ذخیره می کنند.
- Mongos: این قسمت مثل یک روتر عمل می کند و رابطی بین برنامه ی کاربر و شاردهای کلاستر است.

## ۸-۳- تکثیر پایگاه داده

امکان دارد تحت شرایط خاصی، ماشینی که پایگاه داده روی آن اجرا می شود از سرویس خارج گردد و داده های آن از بین برود. پایگاه داده Mongo برای این موضوع راه حل تکثیر (Replication) را پیشنهاد می کند، به صورتی که چندین پایگاه داده Mongo راه اندازی کرده و یک کپی از هر داده در آن ها نگهداری می شود. از مزایای تکثیر می توان به موارد زیر اشاره کرد:

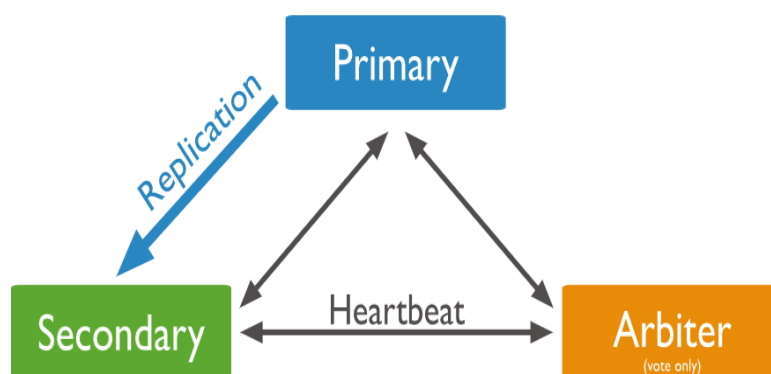
- دسترسی پایدار به داده ها
- از بین نرفتن داده ها
- کاربر می تواند درخواست خواندن داده را، همزمان به چندین سرور ارسال کند

نوع طبقه بندی سند: محرمانه	کد سند: ISRC-AUT-۹۷۰۱۲۱،۰	تاریخ: ۱۳۹۷/۰۱/۲۱	صفحه: ۳۴ از ۵۳
تمامی اطلاعات موجود در این سند متعلق به کارگروه پلتفرم، گروه پژوهشی اینترنت اشیا، دانشگاه صنعتی امیرکبیر بوده و حقوق قانونی آن محفوظ است.			

- استفاده از نسخه های اضافی برای اهداف اختصاصی مانند: بازیابی فوری، تهیه نسخه پشتیبان (Backup) یا گزارش دهی

فرآیندها یا پایگاه داده های Mongo که یک مجموعه داده تکراری را نگه داری می کنند با عنوان Replica set معرفی می گردد. در هر Replica Set تنها یک پایگاه داده اصلی (primary) وجود دارد. عملیات خواندن در این نوع معماری را می توان طوری تنظیم کرد که هم توسط پایگاه داده اصلی و هم توسط پایگاه داده ثانویه (secondary) قابل انجام باشد. در نتیجه می توان از طریق افزودن پایگاه داده secondary به Replica set، کارایی در عملیات خواندن از پایگاه داده را افزایش داد. اما در عملیات نوشتن در پایگاه داده، تنها در پایگاه داده اصلی نوشته می شود و سپس به سایر پایگاه داده های موجود در Replica-set پخش می شود. در نتیجه افزودن پایگاه داده های ثانویه به مجموعه تاثیر در سرعت عملیات نوشتن ندارد.

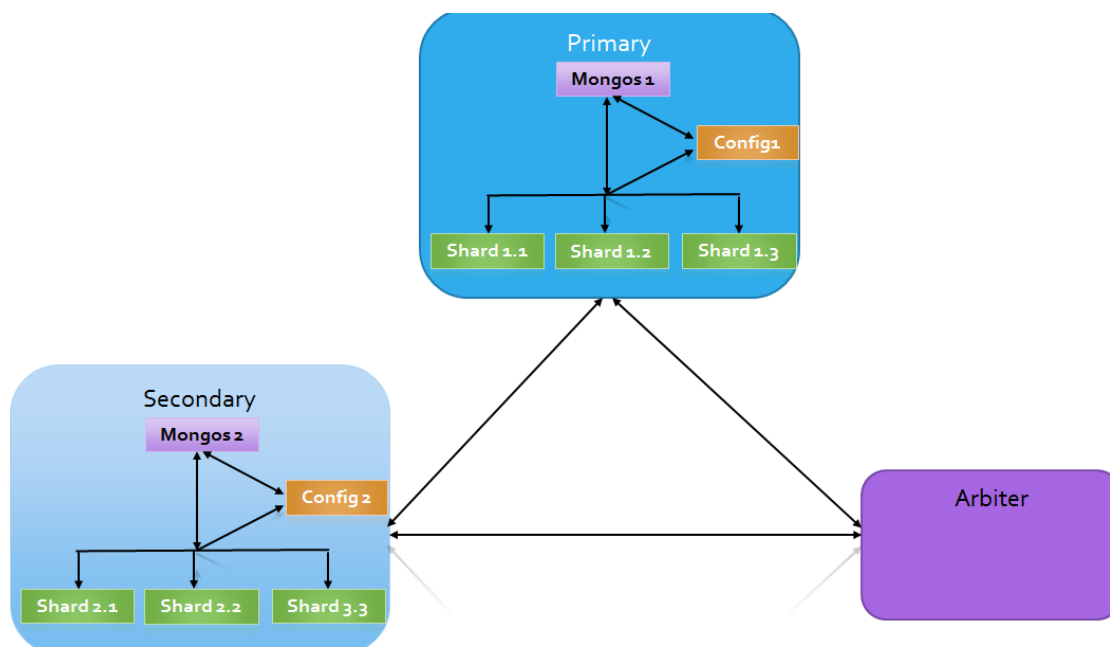
در این نوع معماری در حالتی که چندین پایگاه Secondary وجود داشته باشد، در صورت قطع ارتباط و از دست رفتن پایگاه داده اصلی، سایر اعضا یا Secondary ها در رابطه با انتخاب پایگاه داده اصلی تصمیم گیری می کنند. اما در صورتی که در Replica Set تنها پایگاه داده اصلی و ثانویه وجود داشته باشد، از مولفه ای به نام داور (Arbiter) استفاده می گردد. معماری آن در شکل ۸-۲ نشان داده شده است. Arbiter مانند بقیه اجزا داده ای را نگه داری نمی کند. در این صورت در حالتی که پایگاه داده اصلی از دسترس خارج گردد، Arbiter در انتخاب پایگاه داده ثانویه به عنوان پایگاه داده اصلی نقشی اساسی بازی می کند. باید در نظر داشت که نباید arbiter در سیستم هایی که میزبان Replication های اصلی و ثانویه است راه اندازی گردد.



شکل ۸-۲. معماری تکثیر پایگاه داده

## ۸-۴ - معماری پیشنهادی

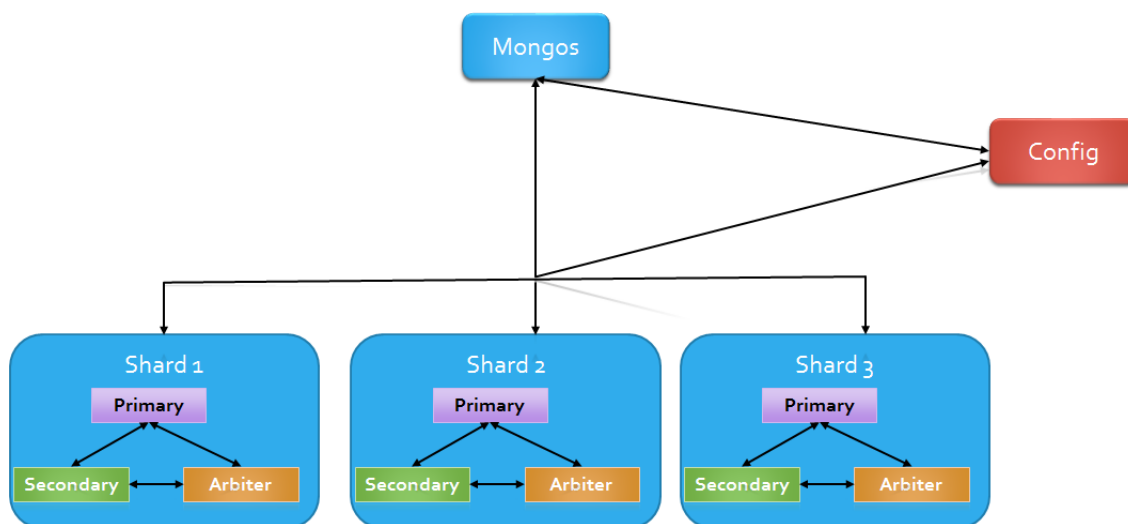
معماری های ارائه شده عموماً در دو مدل ترکیب می گردند. در شکل ۸-۳، پایه و اساس بر اساس معماری تکثیر است ولی در داخل primary و secondary از تکنولوژی sharding استفاده شده است. در این حالت ۶ پایگاه داده جهت خوشه بندی و دو Mongo برای Config و پایگاه داده ای برای داور بودن راه اندازی می گردد. و در آخر نیز دو mongos راه اندازی کرده که replSet آن ها یکسان قرار می گیرد. در این مدل هم سرعت خواندن و هم سرعت نوشتن به صورت جداگانه در primary و Secondary افزایش پیدا کرده است.



شکل ۸-۳. معماری تکثیر با استفاده از خوشه بندی

نوع دیگری از ترکیب معماری در شکل ۸-۴ است. این نوع معماری متداول تر بوده است. در این حالت mongos درخواست های دریافتی ما بین shard های مختلف تقسیم می کند. که در داخل هر shard نیز یک Replica-set تعریف شده است که پشتیبانی داده های موجود در آن shard را فراهم می کند. در این حالت با تقسیم داده ها به صورت متناسب ما بین shard ها و تقسیم درخواست بر اساس بار هر shard می توان سرعت خواندن و نوشتن را بسیار بهبود بخشید. در نهایت availability داده ها نیز افزایش پیدا می کند. در پلتفرم پیشنهادی این ترکیب از Replication و Sharding پیشنهاد و تست شده است که باعث می گردد قابلیت تحمل و ترمیم نسبت به خرابی بهبود یابد و هم افزایش سرعت و کاهش بار روی ماشین های مرتبط به پایگاه داده ایجاد گردد. در نهایت بر اساس میزان فضای در اختیار قرار داده شده و اهمیت داده های مربوط به پایگاه داده های سیستم میزان اعمال آن بر

تمام یا بخشی از پایگاه داده های سیستم تصمیم گیری خواهد شد. ولی این معماری برای پایگاه داده های پارس شده ضروری می باشد.



شکل ۸-۴. معماری خوشه بندی با استفاده از تکثیر

## ۸-۵- انواع پایگاه داده های پلتفرم

### ۸-۵-۱- پایگاه داده های خام

در این بخش از پایگاه داده اطلاعات سنسورها بدون پردازش خاص ذخیره سازی می گردد. در معماری پلتفرم اینترنت اشیا که در شکل ۲-۲ نشان داده شده است این بخش با نام Thing-RD مشخص شده است.

### ۸-۵-۲- پایگاه داده های پارس شده

در این بخش از پایگاه داده اطلاعات پردازش شده از سنسورها ذخیره سازی می گردد. در معماری پلتفرم اینترنت اشیا که در شکل ۲-۲ نشان داده شده است این بخش با نام Thing-PD مشخص شده است.

### ۸-۵-۳- پایگاه داده های پلتفرم

در این بخش از پایگاه داده اطلاعات مربوط به دیگر بخش های پلتفرم اینترنت اشیا مانند کاربران، پرداخت و پروژه ها ذخیره سازی می گردد. در معماری پلتفرم اینترنت اشیا که در شکل ۲-۲ نشان داده شده است این بخش با نام Platform به عنوان یکی از اجزای Databases مشخص شده است.

نوع طبقه بندی سند: محرمانه	کد سند: ISRC-AUT-۹۷۰۱۲۱،۰	تاریخ: ۱۳۹۷/۰۱/۲۱	صفحه: ۳۷ از ۵۳
تمامی اطلاعات موجود در این سند متعلق به کارگروه پلتفرم، گروه پژوهشی اینترنت اشیا، دانشگاه صنعتی امیرکبیر بوده و حقوق قانونی آن محفوظ است.			

#### ۸-۶- سایر موارد مرتبط به پایگاه داده ها

همانگونه که در شکل ۲-۲ نشان داده شده است برای ارتباط با پایگاه داده در Up-Link از Redis نیز استفاده شده است. این واحد نقش یک واسطه (حافظه میانی) را بین پایگاه داده و واحد Up-Link ایفا می کند. این واحد اطلاعات مربوط به اشیاء را پیش از ثبت شدن در پایگاه داده نگهداری می کند. به هنگامی که حجم داده های جمع آوری شده از اشیاء افزایش میابد این واحد در کاهش بار کاری پایگاه داده موثر خواهد بود. سناریوی استفاده از این مولفه در بخش مولفه Up-link شرح داده شده است.

نوع طبقه بندی سند: محرمانه	کد سند: ISRC-AUT-۹۷۰۱۲۱،۰	تاریخ: ۱۳۹۷/۰۱/۲۱	صفحه: ۳۸ از ۵۳
تمامی اطلاعات موجود در این سند متعلق به کارگروه پلتفرم، گروه پژوهشی اینترنت اشیا، دانشگاه صنعتی امیرکبیر بوده و حقوق قانونی آن محفوظ است.			

## ۹- طراحی فرآیند مدیریت اشیا

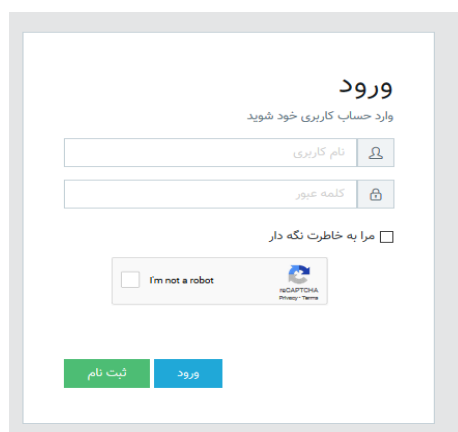
با توجه به نیازهای کارفرما و طراحی انجام شده در رابطه با مدیریت اشیا که تعریف کامل آنها در فصل ۳ با عنوان مولفه ها و فرآیندهای پلتفرم اشیا آورده شده است. اهم موارد مربوط به مدیریت اشیا به شرح زیر می باشد:

- مراحل شناسایی و دریافت داده یک شی دارای پیش نیازهایی است
    - ابتدا باید یک پروژه برای آن شی ساخته شود.
    - گذرگاه مربوطه که به شی متصل است تعریف گردد.
    - پروفایل شی ساخته گردد.
  - پس از انجام پیش نیازهای مربوطه مراحل زیر جهت فعال سازی شی انجام می گردد.
    - در مرحله تعریف شی، اطلاعات آن به همراه پروفایل آن شی معرفی می گردد.
    - در مرحله آخر فعال سازی شی با استفاده از وارد کردن اطلاعاتی مانند Device Address, Application Session Key, Network Session Key و... انجام می گردد.
  - پس از انجام مراحل بالا شی آماده ارسال کدک به آن و نمایش داده می باشد.
- علاوه بر موارد ذکر شده امکانات زیر نیز در رابطه با اشیا در اختیار کاربر قرار داده شده است:
- افزودن، حذف و ویرایش تکی اطلاعات هر شی
  - افزودن، حذف و ویرایش اطلاعات اشیا به صورت دستی
  - امکان تعریف شی در چند پروژه کاربر
  - امکان بررسی لاگ گذرگاه جهت عیب یابی
  - و ...

## ۱۰- طراحی و mockup پنل کاربر

در این بخش با توجه به تغییر مداوم در طراحی واسط گرافیکی کاربری سامانه برخی از صفحاتی که طراحی شده اند و تا حدی مورد تایید کارفرما بوده است به صورت شکل نمایش داده شده است. شایان ذکر است که این صفحات تمام صفحات پنل کاربری نمی باشند. لیست تقریباً کاملی از صفحات کاربر در بخش ۳-۲ مولفه API-Server با عنوان کاربر نهایی آورده شده است. اطلاعات کامل این واسط های گرافیکی و توضیحات آنها در مستندات نهایی پلتفرم آورده خواهد شد.

شکل ۱-۱۰، شکل ۲-۱۰ و شکل ۳-۱۰ صفحات اولیه شامل ورود، ثبت نام و ویرایش اطلاعات کاربران را نمایش می دهد.



شکل ۱-۱۰. صفحه ورود کاربران



شکل ۲-۱۰. صفحه ثبت نام کاربران

نوع طبقه بندی سند: محرمانه	کد سند: ۹۷۰۱۲۱۰۰-ISRC-AUT	تاریخ: ۱۳۹۷/۰۱/۲۱	صفحه: ۴۰ از ۵۳
تمامی اطلاعات موجود در این سند متعلق به کارگروه پلتفرم، گروه پژوهشی اینترنت اشیا، دانشگاه صنعتی امیرکبیر بوده و حقوق قانونی آن محفوظ است.			

### تغییر رمز عبور

رمز عبور فعلی:

رمز عبور جدید:

ذخیره تغییرات

### ویرایش اطلاعات حساب کاربری

نام و نام خانوادگی:

تلفن ثابت:

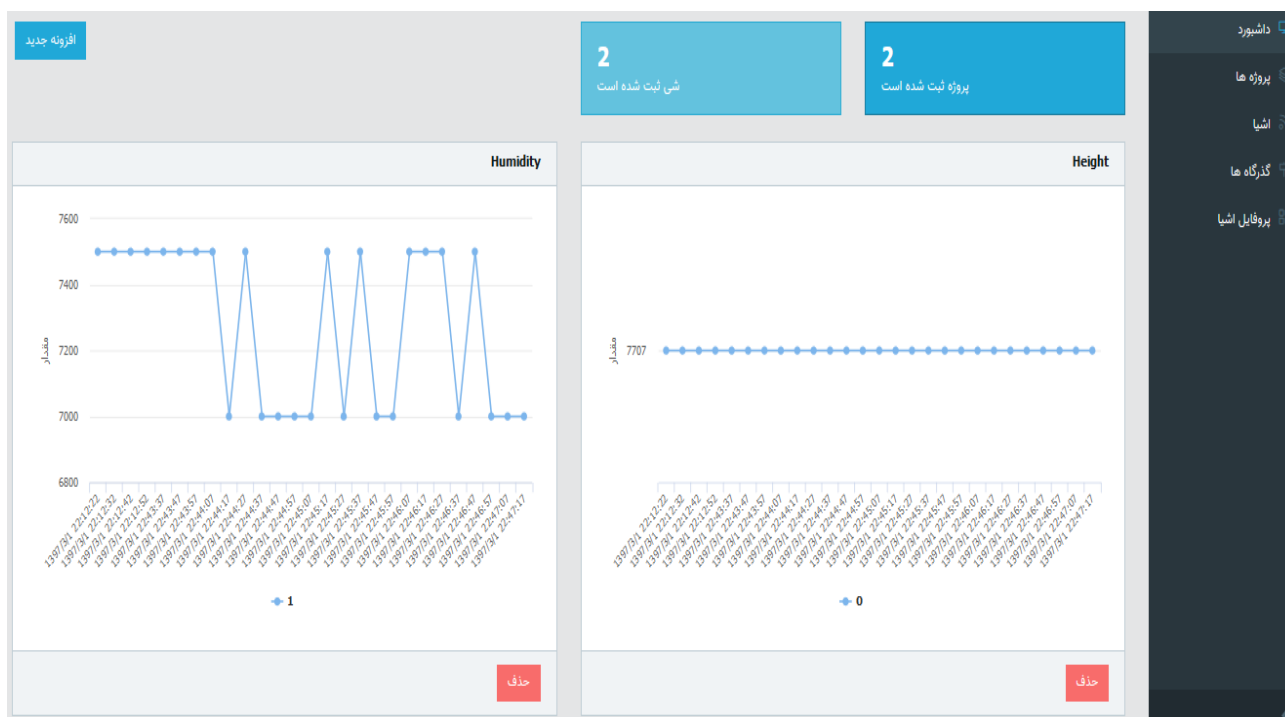
تلفن همراه:

نشانی:

ذخیره تغییرات

شکل ۱۰-۳. صفحه ویرایش اطلاعات حساب کاربری

شکل ۱۰-۴ داشبورد کاربر سامانه را نمایش می دهد که در حال حاضر امکان نمایش تعداد پروژه ها، اشیا و همچنین امکان افزودن نمودار در آن وجود دارد. در شکل ۱۰-۵ امکان افزودن افزونه نمایش داده شده است.



شکل ۱۰-۴. داشبورد سامانه

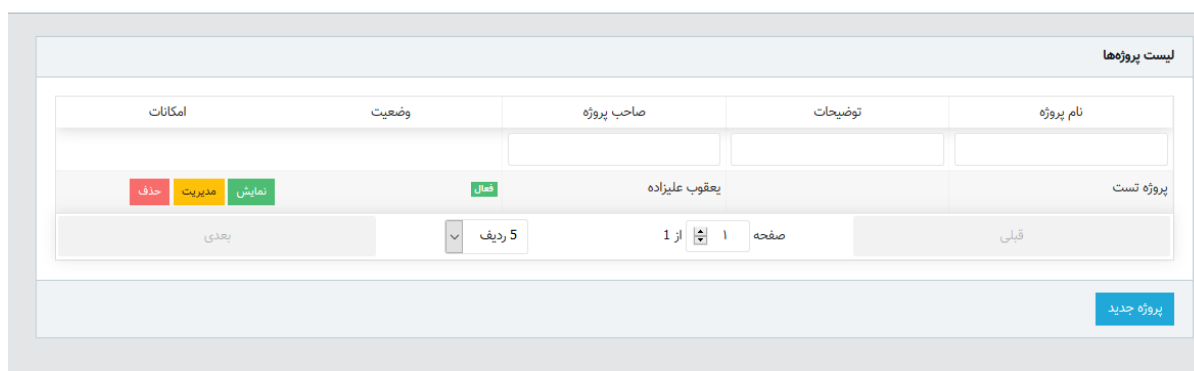




شکل ۱۰-۵. صفحه افزودن افزونه جدید به داشبورد

شکل ۱۰-۶ تا شکل ۱۱-۱۰ امکانات مربوط به مدیریت پروژه را نمایش می دهد. که شامل بخش های مربوط به لیست پروژه ها و امکانات مدیریتی ، ، لیست اشیا و امکانات مرتبط، لیست کدک ها و سناریوها و امکانات مدیریتی آن می باشد.

خانه / لیست پروژه ها



شکل ۱۰-۶. نمایش لیست پروژه ها

پروژه جدید

نام پروژه :

توضیحات :

ذخیره انصراف

شکل ۱۰-۷. صفحه تعریف پروژه جدید

خانه / لیست پروژه ها

تغییر اطلاعات پروژه

نام پروژه:

پروژه تست

توضیحات:

ثبت اطلاعات

نام مستعار کلید داده ها

حذف	نام مستعار	مقدار اصلی
	یافت نشد	
قبلی	صفحه ۱ از 1	بعدی

افزافه کردن

نام مستعار

مقدار اصلی

ثبت

شکل ۱۰-۸. صفحه مدیریت پروژه (۱)

اشیا متصل شده به پروژه

نام شی	آدرس	نوع	امکانات
thing 1	0000000000000006	ABP	فعال سازی ویرایش ارسال codeC ارسال داده (داون لینک) حذف شی وضعیت

قبلی صفحه ۱ از 1 10 ردیف

افزودن شی افزودن دسته ای شی خروجی اکسل

شکل ۱۰-۹. صفحه مدیریت پروژه (۲)

انتخاب سناریو پروژه

عنوان

امکانات

سناریو 1

حذف ویرایش

قبلی

صفحه 1 از 1

10 ردیف

بعدی

افزودن سناریو

شکل ۱۰-۱۰. صفحه مدیریت پروژه (۳)

لیست قالب های codec

عنوان

امکانات

کدک 1

حذف ویرایش

قبلی

صفحه 1 از 1

10 ردیف


بعدی

افزودن قالب

باز کردن

شکل ۱۱-۱۰. صفحه مدیریت پروژه (۴)

با توجه به امکاناتی که در صفحه مدیریت پروژه قرار داده شده است در شکل ۱۰-۱۲ تا شکل ۱۰-۱۶ صفحات امکانات مربوطه شامل افزودن شی، افزودن دسته ای اشیاء، فعال سازی شی، ارسال داده به شی و صفحه ارسال و بررسی کدک نمایش داده شده است.

اطلاعات شی	
نام شی:	<input type="text" value="شی خودکار"/>
توضیحات:	<input type="text" value="شی طبقه سوم"/>
نوع اتصال:	<input type="text" value="انتخاب کنید"/>
پریود ارسال داده:	<input type="text" value="1 دقیقه"/>
شناسه پکت (devEUI):	<input type="text" value="0000000000000000"/>
پرو فایل شی:	<input type="text" value="پرو فایل شی مورد نظر را انتخاب کنید"/>
عرض جغرافیایی:	<input type="text" value="۳۵٫۷۰۲۴۸۵۲"/>
طول جغرافیایی:	<input type="text" value="۵۱٫۴۰۲۳۴۴"/>
<input type="button" value="ثبت"/>	
محل قرارگیری شی	
	

شکل ۱۰-۱۲. صفحه افزودن شی جدید

آپلود فایل Excel	
<p>برای ساخت دسته ای شی مانند فایل اکسل نمونه را در سامانه بارگذاری نمایید</p> <p>توجه کنید برای استفاده از فایل نمونه DevEUI و پرو فایل شی را باید تغییر دهید</p> <p>No file selected. <input type="button" value="...Browse"/></p> <p><input type="button" value="بارگذاری"/></p>	
نتیجه بارگذاری	
#	آدرس فیزیکی
وضعیت افزوده شدن	

شکل ۱۰-۱۳. صفحه افزودن دسته ای اشیا

Activation By Personalization

Device Address:

26011F9C

Application Session Key:

D40C38926D320E99C8878878E4E4AC08

Network Session Key:

ECDE83F7C9F699104EDD76484727879C

Downlink Frame Counter:

0

Downlink Frame Counter:

0

Disable Frame Counter Validation:

☒

انصراف

ارسال

شکل ۱۰-۱۴. صفحه فعال سازی اشیا

Decoder

فایب را انتخاب کنید

▼

1

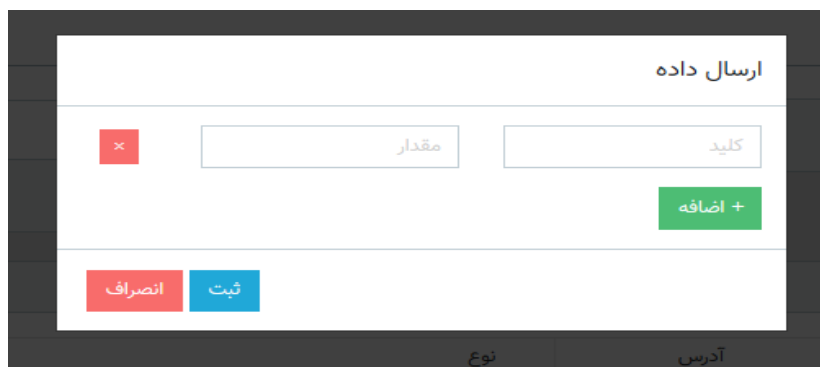
ارسال

بررسی کدک

نتیجه بررسی کدک

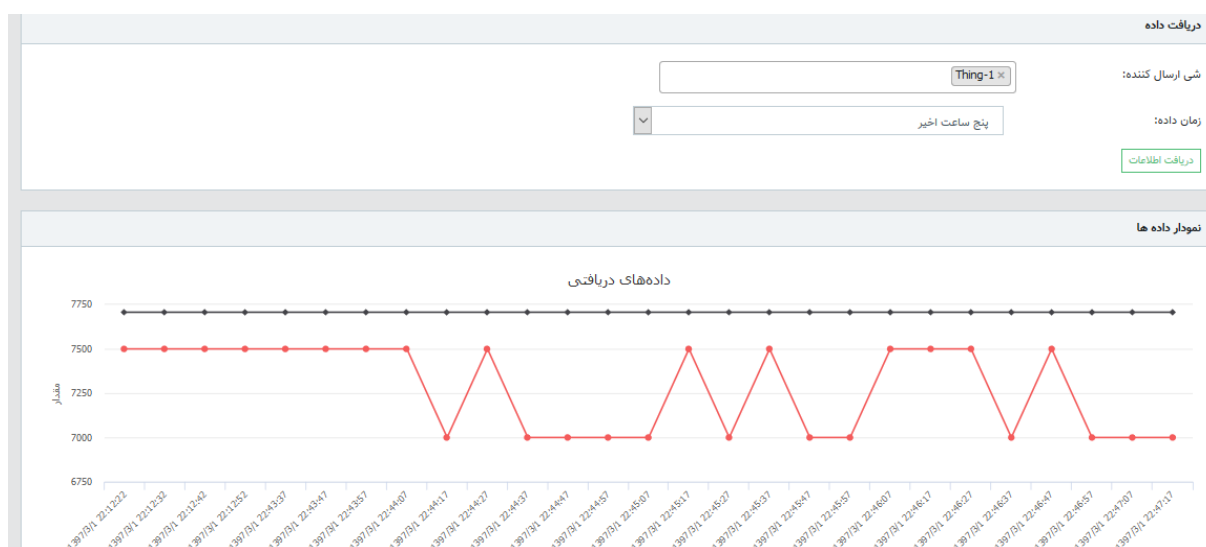
شکل ۱۰-۱۵. صفحه ارسال کدک

نوع طبقه بندی سند: محرمانه	کد سند: ۹۷۰۱۲۱،۰-ISC-AUT	تاریخ: ۱۳۹۷/۰۱/۲۱	صفحه: ۴۶ از ۵۳
تمامی اطلاعات موجود در این سند متعلق به کارگروه پلتفرم، گروه پژوهشی اینترنت اشیا، دانشگاه صنعتی امیرکبیر بوده و حقوق قانونی آن محفوظ است.			



شکل ۱۰-۱۶. صفحه ارسال داده به شی

در شکل ۱۰-۱۷ و شکل ۱۰-۱۸ صفحه نمایش پروژه که دریافت اطلاعات از اشیا مربوط به پروژه را به صورت نموداری و جدولی بر عهده دارد را نشان داده شده است.



شکل ۱۰-۱۷. صفحه نمایش پروژه (۱)

نوع طبقه بندی سند: محرمانه	کد سند: ۹۷۰۱۲۱،۰-ISRC-AUT	تاریخ: ۱۳۹۷/۰۱/۲۱	صفحه: ۴۷ از ۵۳
تمامی اطلاعات موجود در این سند متعلق به کارگروه پلتفرم، گروه پژوهشی اینترنت اشیا، دانشگاه صنعتی امیرکبیر بوده و حقوق قانونی آن محفوظ است.			

داده های جمع آوری شده		
زمان دریافت داده	شی فرستاده	داده دریافت شده
22:47:17 1397/3/1	0000000000000073	{ "ورقویت": "7000" "ارتفاع": "7707" }
22:47:07 1397/3/1	0000000000000073	{ "ورقویت": "7000" "ارتفاع": "7707" }
22:46:57 1397/3/1	0000000000000073	{ "ورقویت": "7000" "ارتفاع": "7707" }
22:46:47 1397/3/1	0000000000000073	{ "ورقویت": "7500" "ارتفاع": "7707" }
22:46:37 1397/3/1	0000000000000073	{ "ورقویت": "7000" "ارتفاع": "7707" }

شکل ۱۰-۱۸. صفحه نمایش پروژه (۲)

در ادامه شکل ۱۰-۱۹ از منوی اصلی لیست اشیا را نمایش می دهد.

لیست اشیا					
#	نام شی	توضیحات	آدرس فیزیکی	پروژه های متصل	امکانات
1	Thing-1	این شی برای ارسال دان لینک است.	0000000000000073	New Project	مشاهده پروژه
2	Thing-2	Tester 2	0000000000000074	New Project	مشاهده پروژه

شکل ۱۰-۱۹. صفحه لیست اشیا

شکل ۱۰-۲۰ و شکل ۱۰-۲۱ امکانات مربوط به گذرگاه را نمایش داده اند که شامل نمایش لیست گذرگاه ها،

حذف و افزودن گذرگاه می باشد.

Gateways List				
نام گذرگاه (gateway)	توضیحات	شناسه گذرگاه	وضعیت	عملیات
Real Gateway	این گذرگاه يك گذرگاه واقعي است	b827ebffff47d1a5	فعال	نمایش حذف
GW868-GPS2	گذرگاه تستی	b827ebffff66232d	فعال	نمایش حذف
Test	This is a test	b827ebffff66232e	غیرفعال	نمایش حذف
قبلی		صفحه ۱ از ۱	10 ردیف	بعدی
ساخت جدید				

شکل ۱۰-۲۰. صفحه لیست گذرگاه ها

**افزودن Gateway**

اسم:


شناسه گذرگاه:

توضیحات:

عرض جغرافیایی:

طول جغرافیایی:

ارتفاع:



شکل ۱۰-۲۱. صفحه افزودن گذرگاه

شکل ۱۰-۲۲ و شکل ۱۰-۲۳ نیز امکانات مرتبط به مدیریت پروفایل اشیا شامل لیست، مشاهده، حذف و ساخت را در اختیار کاربر قرار می دهد.

**پروفایل اشیا**

عنوان	شناسه پروفایل	عملیات
ABP	37	<div style="display: flex; gap: 5px;"> <div style="background-color: #007bff; color: white; padding: 2px 5px;">مشاهده</div> <div style="background-color: #dc3545; color: white; padding: 2px 5px;">حذف</div> </div>
AUT-HW	40	<div style="display: flex; gap: 5px;"> <div style="background-color: #007bff; color: white; padding: 2px 5px;">مشاهده</div> <div style="background-color: #dc3545; color: white; padding: 2px 5px;">حذف</div> </div>
Test1	41	<div style="display: flex; gap: 5px;"> <div style="background-color: #007bff; color: white; padding: 2px 5px;">مشاهده</div> <div style="background-color: #dc3545; color: white; padding: 2px 5px;">حذف</div> </div>

قبلی

۱۰ ردیف

۱ از ۱

بعدی

ساخت پروفایل

شکل ۱۰-۲۲. صفحه لیست پروفایل اشیا



ساخت پروفایل جدید

اطلاعات کلی فعال سازی کلاس B کلاس C

Name:

LoRaWAN MAC Version:

LoRaWAN Regional Parameters revision:

Max EIRP:

ثبت اطلاعات

شکل ۱۰-۲۳. صفحه ساخت پروفایل اشیا

نوع طبقه بندی سند: محرمانه	کد سند: ۹۷۰۱۲۱،۰-ISRC-AUT	تاریخ: ۱۳۹۷/۰۱/۲۱	صفحه: ۵۰ از ۵۳
تمامی اطلاعات موجود در این سند متعلق به کارگروه پلتفرم، گروه پژوهشی اینترنت اشیا، دانشگاه صنعتی امیرکبیر بوده و حقوق قانونی آن محفوظ است.			

## ۱۱- طراحی و mockup پنل مدیریت

در پنل مدیریت سامانه نیز به مانند پنل کاربری، لیست تمام صفحات در این گزارش به دلیل تغییرات مداوم ذکر نشده است. لیست اولیه ای از صفحات پنل مدیریت در بخش ۳-۲ مولفه API-Server با عنوان کاربر مدیر پلتفرم آورده شده است. در ادامه لیستی از طراحی های انجام شده آورده شده است.

در شکل ۱۱-۱ لیست کاربران سامانه به همراه قابلیت جستجو و مشاهده وضعیت آن ها نمایش داده شده است.

لیست کاربران		
نام کاربر	ایمیل	نوع کاربر
سجاد رهنما	sajjad.rahnama7@gmail.com	حقیقی
سپهر صبور	sepehr.sabour@gmail.com	حقیقی
آرش یادگاری	aryadegari@gmail.com	حقیقی
گروه پلتفرم	parham.alvani@gmail.com	حقیقی
yaghouh alizadeh	yaghouh.al@gmail.com	حقیقی
قبلی	صفحه ۱ از ۵ ردیف	بعدی

شکل ۱۱-۱. صفحه لیست کاربران سامانه

برای هر کاربر اطلاعات آن در چهار بخش اطلاعات عمومی، بسته های خریداری شده، تراکنش ها، پروژه ها و اشیا در شکل ۱۱-۲ و شکل ۱۱-۳ نمایش داده شده است.

اطلاعات کاربر

اطلاعات حساب کاربری			
نام کاربر	تست	پست الکترونیکی	تست
تلفن ثابت	تست	تلفن همراه	تست
نوع کاربری	تست	نوع مجموعه	تست

پروژه های کاربر			
#	نام پروژه	وضعیت	
۱	پروژه اول	تست	
۲	پروژه اول	تست	
۳	پروژه اول	تست	
۴	پروژه اول	تست	
۵	پروژه اول	تست	

تراکنش های کاربر				
#	نام بسته	تاریخ	هزینه	وضعیت
۱	بسته اول	۱۳۹۳-۱۰-۰۵	۳۰۰۰۰	تست
۲	بسته اول	۱۳۹۳-۱۰-۰۵	۳۰۰۰۰	تست
۳	بسته اول	۱۳۹۳-۱۰-۰۵	۳۰۰۰۰	تست
۴	بسته اول	۱۳۹۳-۱۰-۰۵	۳۰۰۰۰	تست
۵	بسته اول	۱۳۹۳-۱۰-۰۵	۳۰۰۰۰	تست

شکل ۱۱-۲. صفحه اطلاعات کاربر (۱)

اشیا کاربر		
#	نام شی	نوع
۱	شی اول	تایپ ۱
۲	شی اول	تایپ ۱
۳	شی اول	تایپ ۱
۴	شی اول	تایپ ۱
۵	شی اول	تایپ ۱

بسته های کاربر			
#	نام بسته	هزینه	وضعیت
۱	بسته اول	۳۰۰۰۰	فعال
۲	بسته اول	۴۰۰۰۰	فعال
۳	بسته اول	۳۰۰۰۰	فعال
۴	بسته اول	۳۰۰۰۰	فعال
۵	بسته اول	۳۰۰۰۰	فعال

شکل ۱۱-۳. صفحه اطلاعات کاربر (۲)

در نهایت شکل ۱۱-۴ تا شکل ۱۱-۶ صفحات مربوط به لیست بسته ها (شامل نمایش، حذف و ایجاد بسته جدید)، مدیریت درگاه های پرداخت و لیست تراکنش ها را نمایش می دهد.

لیست بسته‌ها

مبلغ قابل پرداخت	0 ریال
تعداد سنسور	3
مهلت بسته	30

تغییرات

حذف

مبلغ قابل پرداخت	100000 ریال
تعداد سنسور	50
مهلت بسته	45

تغییرات

حذف

مبلغ قابل پرداخت	180000 ریال
تعداد سنسور	100
مهلت بسته	90

تغییرات



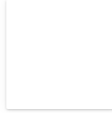
حذف

بسته جدید

شکل ۱۱-۴. صفحه لیست بسته های موجود در سامانه

اطلاعات بسته	
نام بسته :	<input type="text"/>
تعداد سنسور :	<input type="text"/>
مهلت بسته :	<input type="text"/>
مبلغ بسته (تومان) :	<input type="text"/>
<input type="button" value="ثبت بسته"/>	

شکل ۱۱-۵. صفحه افزودن بسته جدید

درگاه های پرداخت					
#	لوگو درگاه	نام درگاه	وبسایت	وضعیت	فعال سازی
1		درگاه بانک ملت	http://bankmellat.ir	فعال	<input type="button" value="فعال سازی"/>
2		درگاه بانک پارسیان	http://parsian-bank.ir	غیر فعال	<input type="button" value="فعال سازی"/>
3		درگاه بانک آینده	https://www.ba24.ir	فعال	<input type="button" value="فعال سازی"/>

شکل ۱۱-۶. صفحه مدیریت درگاه های بانکی