



دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

پایان نامه کارشناسی

گرایش مهندسی فناوری اطلاعات

عنوان

طراحی و پیاده سازی واسط کاربری موبایل پلتفرم اینترنت اشیا

نگارش

سارا واسعی

استاد راهنما

دکتر بهادر بخشی

استاد مشاور

دکتر مسعود صبائی

فروردین ۱۳۹۸

صفحه فرم ارزیابی یا تصویب پایان نامه- اعضاء کمیته دفاع

در این صفحه برگه تایید و تصویب پایان نامه موسوم به فرم کمیته دفاع را قرار دهید.

نکته مهم:

نگارش پایان نامه باید به زبان فارسی و بر اساس آخرین نسخه دستورالعمل و راهنمای نگارش پایان نامه های دانشگاه صنعتی امیرکبیر باشد.(دستورالعمل و راهنمای حاضر)

اینجانب سارا واسعی متعهد می‌شوم که مطالب مندرج در این پایان نامه حاصل کار پژوهشی اینجانب تحت نظارت و راهنمایی اساتید دانشگاه صنعتی امیرکبیر بوده و به دستاوردهای دیگران که در این پژوهش از آنها استفاده شده است مطابق مقررات و روال متعارف ارجاع و در فهرست منابع و مآخذ ذکر گردیده است. این پایان نامه قبلاً برای احراز هیچ مدرک هم‌سطح یا بالاتر ارائه نگردیده است.

در صورت اثبات تخلف در هر زمان، مدرک تحصیلی صادر شده توسط دانشگاه از درجه اعتبار ساقط بوده و دانشگاه حق پیگیری قانونی خواهد داشت.

کلیه نتایج و حقوق حاصل از این پایان نامه متعلق به دانشگاه صنعتی امیرکبیر می‌باشد. هرگونه استفاده از نتایج علمی و عملی، واگذاری اطلاعات به دیگران یا چاپ و تکثیر، نسخه‌برداری، ترجمه و اقتباس از این پایان نامه بدون موافقت کتبی دانشگاه صنعتی امیرکبیر ممنوع است. نقل مطالب با ذکر مآخذ بلامانع است.

سارا واسعی

امضا

سپاس مخصوص خداوند است که به انسان توانایی و دانایی بخشید تا به بندگان شفقت ورزد.

از استاد ارجمند جناب آقای دکتر بخشی، به جهت راسمنایی بایشان در طول انجام این پروژه کمال تشکر و قدردانی را دارم.

چکیده

اینترنت اشیا، فناوری روبه‌رشدی است که همه جنبه‌های زندگی بشر را تحت تأثیر قرار خواهد داد. در این اکوسیستم، اشیا با استفاده از حسگرهای داخلی خود می‌توانند داده را جمع‌آوری کرده برای تجزیه و تحلیل به یک سامانه مرکزی ارسال کرده و دستورات دریافتی از آن اجرا نمایند. برای این منظور لازم است تا بستری در دسترس افراد باشد تا بتوانند به واسطه‌ی آن بر مدیریت این وسایل و محیط خود فائق آیند. در معماری چهار لایه اینترنت اشیا، از این بستر به عنوان پلتفرم اینترنت اشیا نام برده می‌شود. از جمله نیازمندی‌های این گونه پلتفرم‌ها، ذخیره داده‌های جمع‌آوری شده توسط اشیا و نمایش آنهاست. این قابلیت باید در یک واسطه کاربری مناسب در اختیار کاربران قرار گیرد. در نسخه فعلی توسعه داده شده از پلتفرم اینترنت اشیا در دانشکده، واسطه کاربری پلتفرم محیط وب است. با همه‌گیر شدن گوشی‌های هوشمند در سالهای اخیر، اکنون بسیاری از کاربران ترجیح می‌دهند که امور روزمره خود را از طریق برنامه‌های کاربردی گوشی‌های موبایل انجام دهند. بنابراین در این پروژه هدف این است که نرم‌افزار موبایلی طراحی و پیاده‌سازی شود که امکان مدیریت اشیا فراهم می‌کند. این برنامه با استفاده از رابط‌های برنامه‌نویسی کاربردی که پلتفرم در اختیار قرار می‌دهد توسعه داده خواهد شد. این برنامه قابلیت‌های متعددی در اختیار کاربر قرار می‌دهد که از آن جمله می‌توان به اضافه/حذف اشیا، مشاهده داده‌ها، خرید بسته‌ها و ... اشاره کرد.

واژه‌های کلیدی:

اینترنت اشیا، رابط برنامه‌نویسی کاربردی، اپلیکیشن موبایل، معماری MVC

۱	فصل اول مقدمه	۱
۳	۱.۱ ساختار پایان نامه	۳
۵	۲ فصل دوم طرح مسئله، مراحل و روش کار	۵
۶	۱.۲ طرح مسئله	۶
۷	۲.۲ مراحل و روش انجام	۷
۸	۳.۲ نیازمندی‌های پروژه	۸
۹	۱.۳.۲ نمودار مورد کاربرد پروژه	۹
۹	۲.۴ جمع‌بندی	۹
۱۰	۳ فصل سوم اینترنت اشیا	۱۰
۱۱	۱.۳ تاریخچه به وجود آمدن اینترنت اشیا	۱۱
۱۱	۲.۳ اینترنت اشیا چیست؟	۱۱
۱۲	۳.۳ معماری اینترنت اشیا	۱۲
۱۳	۱.۳.۳ لایه ادراک و شناسایی	۱۳
۱۳	۲.۳.۳ لایه شبکه	۱۳
۱۵	۳.۳.۳ لایه پردازش اطلاعات	۱۵
۱۵	۴.۳.۳ لایه کاربرد	۱۵
۱۶	۴.۳ کاربردهای اینترنت اشیا	۱۶
۱۶	۱.۴.۳ سلامت الکترونیک	۱۶
۱۶	۲.۴.۳ شهر هوشمند	۱۶
۱۷	۳.۴.۳ خانه هوشمند	۱۷
۱۷	۴.۴.۳ مدیریت هوشمند انرژی	۱۷
۱۸	۵.۴.۳ مانیتورینگ محیطی	۱۸
۱۸	۶.۴.۳ تدارکات هوشمند	۱۸
۱۸	۵.۳ چالش‌های پیش روی اینترنت اشیا	۱۸
۱۸	۱.۵.۳ پردازش داده‌های کلان	۱۸
۱۹	۲.۵.۳ امنیت	۱۹
۱۹	۳.۵.۳ حریم خصوصی	۱۹
۱۹	۴.۵.۳ سازگاری بین تجهیزات	۱۹
۲۰	۳.۶ LoRaWAN	۲۰
۲۰	۱.۶.۳ معماری فنی شبکه لورا	۲۰
۲۱	۲.۶.۳ لایه فیزیکی و مدولاسیون لورا	۲۱
۲۲	۳.۶.۳ مدل توسعه شبکه لورا	۲۲

۲۳	جمع‌بندی	۳.۷
۲۴	فصل چهارم پیاده‌سازی	۴
۲۶	معماری نرم‌افزار	۱.۴
۲۶	معماری MVC	۱.۱.۴
۲۷	زبان توسعه	۲.۴
۲۷	دات	۱.۲.۴
۲۸	PHP	۲.۲.۴
۲۸	چارچوب توسعه	۴.۳
۲۸	فلاتر	۱.۳.۴
۲۹	پلتفرم دات	۱.۱.۳.۴
۲۹	موتور فلاتر	۲.۱.۳.۴
۳۰	کتابخانه‌های اساسی	۳.۱.۳.۴
۳۰	ویجت‌های پیش‌طراحی‌شده	۴.۱.۳.۴
۳۰	لاراول	۲.۳.۴
۳۰	مدل (پایگاه‌های اطلاعاتی)	۴.۴
۳۱	View	۵.۴
۳۱	معرفی معماری MVC درون اپلیکیشن	۱.۵.۴
۳۱	مدل	۱.۱.۵.۴
۳۲	کنترلر	۲.۱.۵.۴
۳۲	بخش ظاهری اپلیکیشن موبایل	۳.۱.۵.۴
۳۳	صفحه احراز هویت	۲.۵.۴
۳۴	صفحه خانه	۳.۵.۴
۳۶	منوی دراور	۴.۵.۴
۳۷	پروژه‌ها	۴.۵.۵
۳۷	لیست پروژه‌های دریافتی	۱.۵.۵.۴
۳۸	منو عملیات‌های مدیریتی	۲.۵.۵.۴
۳۹	ایجاد پروژه	۳.۵.۵.۴
۴۰	نمایش پروژه	۴.۵.۵.۴
۴۴	مدیریت پروژه	۵.۵.۵.۴
۴۵	حذف پروژه	۶.۵.۵.۴
۴۷	پرو فایل اشیا	۶.۵.۴
۵۲	اشیا (متصل به پروژه)	۷.۵.۴
۵۴	گذرگاه‌ها	۸.۵.۴
۵۶	بسته‌ها	۹.۵.۴
۵۸	کنترلر	۴.۶
۵۹	توضیحات اجمالی از برخی موارد تخصصی	۷.۴

۶۱ جمع‌بندی	۴.۸
۶۲ فصل پنجم جمع‌بندی و نتیجه‌گیری	۵
۶۴ منابع و مراجع	

صفحه

فهرست اشکال

شکل ۱/۱ پیش‌بینی روند رشد تعداد اشیا و افراد در اینترنت اشیا [۳]	۳
شکل ۲/۱ نمودار مورد کاربرد پروژه	۹
شکل ۳/۱ ابعاد مختلف اینترنت اشیا	۱۲
شکل ۳/۲ مدل چهار لایه‌ای اینترنت اشیا	۱۳
شکل ۳/۳ فناوری‌های مختلف ارتباطی در اینترنت اشیا	۱۴
شکل ۳/۴ معماری ساختار شبکه لورا	۲۰
شکل ۳/۵ جایگاه لورا در معماری لایه‌ای	۲۲
شکل ۴/۱ ارتباط سه مولفه MVC با یکدیگر	۲۷
شکل ۴/۲ صفحه ورود به سامانه	۳۳
شکل ۴/۳ صفحه خانه	۳۴
شکل ۴/۴ نمایش داده خام دریافتی	۳۵
شکل ۴/۵ منو دراور	۳۶
شکل ۴/۶ همه پروژه‌های کاربر	۳۷
شکل ۴/۷ منو عملیات‌های مدیریتی	۳۸
شکل ۴/۸ ایجاد پروژه	۳۹
شکل ۴/۹ نمایش پروژه	۴۰
شکل ۴/۱۰ انتخاب شی در نمایش پروژه	۴۱
شکل ۴/۱۱ انتخاب زمان آغاز و پایان از تقویم جلالی	۴۲
شکل ۴/۱۲ اطلاعات دریافتی در نمایش پروژه	۴۳
شکل ۴/۱۳ صفحه مدیریت پروژه	۴۴
شکل ۴/۱۴ پرسش حذف پروژه	۴۵
شکل ۴/۱۵ تاییدیه حذف پروژه	۴۶
شکل ۴/۱۶ لیست پروفایل اشیا	۴۷
شکل ۴/۱۷ منوی عملیاتی پروفایل اشیا	۴۸
شکل ۴/۱۸ مشاهده پروفایل اشیا	۴۹
شکل ۴/۱۹ پرسش حذف پروفایل اشیا	۵۰
شکل ۴/۲۰ تاییدیه حذف پروفایل اشیا	۵۱
شکل ۴/۲۱ لیست اشیا متصل به پروژه‌ها	۵۲
شکل ۴/۲۲ جزییات اطلاعات اشیا	۵۳
شکل ۴/۲۳ لیست گذرگاه‌ها	۵۴
شکل ۴/۲۴ جزییات اطلاعات گذرگاه‌ها	۵۵

۵۶ شکل ۴/۲۵ بسته خریداری شده توسط کاربر
۵۷ شکل ۴/۲۶ صفحه خرید بسته

۱

فصل اول

مقدمه

مقدمه

اینترنت اشیا یک زیرساخت سراسری برای ایجاد مجموعه داده‌ها، ارائه سرویس‌های پیشرفته با استفاده از اتصال متقابل اشیاء (فیزیکی و مجازی) است که مبتنی بر فناوری‌های برهم‌کنش‌پذیر^۱ اطلاعات و ارتباطات است [۱]. اگرچه این اکوسیستم در ذات خویش یک پیچیدگی روزافزون دارد؛ مفهوم اینترنت اشیا، مفهومی ساده و درعین حال قدرتمند است. ریشه‌ی فعالیت‌های اینترنت اشیا به دانشگاه ام‌آی‌تی بازمی‌گردد. در سال ۱۹۹۹ یک گروه پژوهشی این دانشگاه که در زمینه شناسایی شبکه‌ای بازشناسی امواج رادیویی^۲ و فناوری سنجش کار می‌کردند، مأمور به طراحی معماری اینترنت اشیا شدند. اینترنت اشیا، پس از اینترنت و شبکه تلفن همراه، سومین موج از فناوری‌های اطلاعات و ارتباطات^۳ و اولین تحول واقعی اینترنت با جهش عظیمی در توانایی جمع‌آوری، تحلیل و توزیع داده‌ها و تبدیل آن‌ها به اطلاعات، دانش و خرد است. به‌علاوه قابلیت بالایی برای تحول عظیم در نحوه زندگی، یادگیری، کار و تفریح مردم دارد. به همین سبب اینترنت اشیا از اهمیت ویژه‌ای برخوردار است. همانطور که در شکل ۱-۱ مشاهده می‌شود، کارشناسان پیش‌بینی می‌کنند که تا سال ۲۰۲۰ بیش از ۵۰ میلیارد دستگاه به بخشی از اینترنت اشیا بدل خواهند شد و درآمدهای اقتصادی حاصل از این فناوری از ۳/۰۲ تریلیارد دلار تجاوز خواهد کرد [۲]. این فناوری با افزودن حسگرها (دماسنج، فشارسنج، نورسنج، رطوبت‌سنج و ...) به اینترنت، ما را به سمت کنش بیشتر و واکنش کمتر سوق می‌دهد.

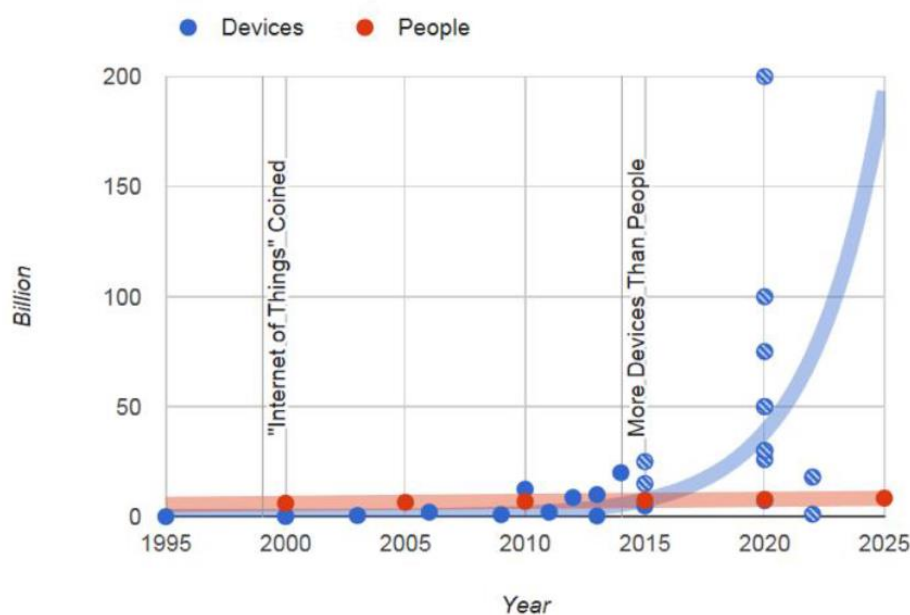
از مهم‌ترین بخش‌های این فناوری، نظارت، تحلیل و مدیریت انسان روی فعالیت این اشیا است بنابراین موانعی با گسترش آن، ابزارهایی برای این هدف توسعه داده می‌شوند.

در گذشته اعضای گروه پژوهشی اینترنت اشیا دانشگاه صنعتی امیرکبیر، نرم‌افزاری با ویژگی‌های اشاره-شده در بستر وب توسعه داده بودند اما با گسترش روزافزون دستگاه‌های هوشمند قابل حمل، نیاز به توسعه‌ی یک نرم‌افزار در بستر موبایل نتیجه شد.

^۱ Interoperable

^۲ Networked radio frequency identification

^۳ ICT



شکل ۱.۱ پیش‌بینی روند رشد تعداد اشیا و افراد در اینترنت اشیا [۳]

موارد پایه‌ای و مهم زیر در هر دو نرم‌افزار (تحت وب و موبایل) موجود هستند:

- گذرگاه و امکانات مرتبط
- پروفایل اشیا
- تعریف پروژه
- نمایش پروژه
- مدیریت پروژه
- مشاهده و مدیریت اشیا هر کاربر
- بسته‌های کاربر

۱.۱ ساختار پایان‌نامه

در فصل دوم با ارائه‌ی توضیحاتی به شفاف کردن نیازمندی‌های پروژه و تعریف دقیق مسئله می‌پردازیم. فصل سوم ضمن توضیح پیرامون اینترنت اشیا، کاربردهای آن معرفی شده و چالش‌های پیش روی این

فناوری را عنوان می‌کنیم. در فصل چهارم الگوهای به کار گرفته شده در جهت تولید نرم‌افزار و پیاده‌سازی سیستم گام برمی‌داریم و در فصل پایانی هم به یک جمع‌بندی کوتاه از کار می‌پردازیم.

۲

فصل دوم

طرح مسئله، مراحل و روش کار

طرح مسئله، مراحل و روش کار

پیش از آغاز پیاده‌سازی نرم‌افزار لازم است درک جامع و کاملی از مسئله پیدا کنیم و فهرست دقیقی از نیازمندی‌های آن تهیه نماییم.

در این فصل سعی شده است با بیان مطالب مورد نیاز، کلیات مسئله، نیازمندی‌ها و روش حل آن مشخص شوند.

۱.۲ طرح مسئله

با توجه به افزایش نفوذ اینترنت و اشیائی که در بستر اینترنت به یکدیگر متصل هستند، لازم است سامانه‌ای موجود باشد تا امکان رصد جریان داده‌ی انتقالی و وضعیت اشیا را در هر زمان به کاربران بدهد. با افزایش چشمگیر دستگاه‌های قابل حمل مانند موبایل، شاهد رقابت کسب‌وکارها هستیم که می‌خواهند نرم‌افزارهای خود را روی این بسترهای پرطرفدار نیز توسعه دهند.

اسماعیل ثنایی، استاد دانشگاه صنعتی شریف در همایش «شهر هوشمند زیرساخت و فرصت‌های سرمایه‌گذاری» که در شهریور ۱۳۹۴ برگزار شد، اعلام کرد: «اگر می‌خواهیم شهر هوشمند داشته باشیم باید زیرساختی ایجاد کنیم که شامل لایه‌هایی از حسگرها، دوربین‌ها، ردیاب و شبکه باشد تا ارتباطات را به هم وصل کرده و سپس داده‌های ارتباطات دیتا^۱، چیزها در بخش داده‌های بزرگ^۲ پیاده و پردازش شده و گزارش‌ها استخراج و قابل استفاده شود.» [۴]

همچنین وزیر ارتباطات و فناوری محمود واعظی در همین همایش، داشتن زیرساخت برای شهر هوشمند را لازم ولی کافی ندانست و افزود: «از جمله موارد مهم دیگری که باید فراهم باشد تا به اهداف شهر هوشمند رسید خدمات، محتوا و اپلیکیشن‌هاست.» [۴]

در همین راستا پروژه‌ای تعریف شد که هدف آن پیاده‌سازی لایه نهایی فناوری اینترنت اشیا بود.

در حقیقت نرم‌افزاری برای موبایل به منظور نمایش و مدیریت اشیا و اطلاعات آنها تولید شده است.

Data ^۱

Big data ^۲

۲.۲ مراحل و روش انجام

پیروی از یک رویه منظم تولید نرم افزار به تولیدکنندگان نرم افزار کمک می کند امور مربوط به تولید نرم افزار را منظم کرده و پروژه را در حداقل زمان ممکن و با کارایی بالایی انجام دهند.

برای مدیریت فرایند تولید نرم افزار و کنترل چرخه حیات تولید نرم افزار، متدولوژی ها و رویه های مختلفی وجود دارد که بنابر نیازهای پروژه، ترجیحات مشتری، دیدگاه های مدیریتی و ... می توان یکی از این روش ها و یا ترکیبی از این الگوها را در فرایند تولید نرم افزار به کار گرفت.

با توجه به این که پیاده سازی به شکل شی گرا خواهد بود، متدولوژی انتخاب شده برای پروژه باید متناسب با طراحی شی گرا باشد. از مجموعه متدولوژی ها مناسب برای کنترل پروژه های نرم افزاری شی گرا متدولوژی ^۱ RUP که استاندارد شده، انتخاب شده است. از آنجایی که هدف از این متدولوژی طراحی و تولید سریع و با کیفیت بالای نرم افزارهای شی گرا است، برای این پروژه بسیار مناسب است. متدولوژی های ساخت یافته ^۲ متدهایی هستند که تمرکز بسیار زیادی روی تولید مستندات می گذارند، در مقابل متدولوژی های چابک ^۳ متدولوژی هایی هستند که هدفشان رضایت مشتری است و تولید مستندات انجام نمی گیرد و به مرور هزینه تغییرات بالا می رود و متدولوژی های تکراری ^۴ که متدولوژی هایی هستند که به دلیل تکرارها و تغییرات زیاد کیفیت محصول کاهش می یابد. RUP یک متدولوژی تکراری است که تقریباً از نقطه قوت سه دسته بالا استفاده می کند. متدولوژی RUP شامل چهار فاز آغاز^۵، تکمیل^۶، ساخت^۷ و انتقال^۸ است.

^۱ Rational Unified Process

^۲ Structural

^۳ Agile

^۴ Iterative

^۵ Inception

^۶ Elaboration

^۷ Construction

^۸ Transition

۳.۲ نیازمندی‌های پروژه

با توجه به موارد مطرح‌شده در بخش ۱.۲ نیازمندی‌های پروژه مشخص شد:

- امکان ایجاد، مشاهده اطلاعات، ویرایش و حذف پروژه
- امکان ایجاد، مشاهده اطلاعات، ویرایش و حذف شی
- مشاهده اطلاعات گذرگاه‌ها
- مشاهده اطلاعات دریافت‌شده توسط اشیا با توجه به محدوده‌ی زمانی دلخواه
- مشاهده بسته فعال کاربر و خرید بسته جدید
- مشاهده اطلاعات کاربری

برمبنای این نیازمندی‌ها وظایف پروژه مشخص شد:

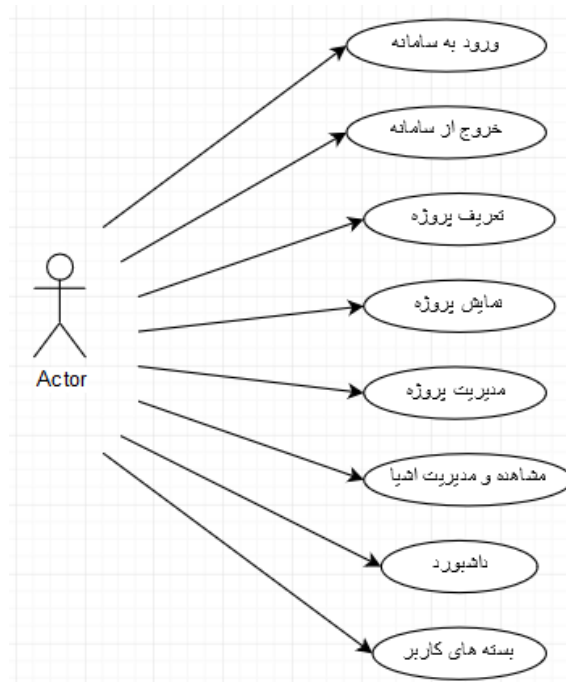
۱. بررسی مدل‌های موجود در پایگاه داده و ارتباطات میان موجودیت‌ها
۲. تست رابط‌های برنامه‌نویسی کاربردی
۳. طراحی و پیاده‌سازی معماری کلاس‌ها
۴. طراحی ظاهر نرم‌افزار با داده‌های تستی و مطابق با استانداردهای طراحی اپلیکیشن^۱ موبایل
۵. پیاده‌سازی سیستم احراز هویت در فلاتر^۲
۶. اتصال کلاس‌ها به پایگاه داده با کمک رابط‌های برنامه‌نویسی کاربردی
۷. تست‌های نهایی اپلیکیشن

^۱ Application

^۲ Flutter

۱.۳.۲ نمودار مورد کاربرد پروژه^۱

نمودار مورد کاربرد به خوبی می‌تواند نیازهای کاربران سیستم را نمایش دهد و بخش‌هایی از سیستم را که به آن دسترسی دارند مشخص کند:



شکل ۱.۲ نمودار مورد کاربرد پروژه

۴.۲ جمع‌بندی

برای یافتن راه حل یک مسئله، در ابتدا باید درک جامع و کاملی از مسئله پیدا کنیم و فهرستی از نیازمندی‌ها تهیه نماییم. پس از آن مسئله را به بخش‌های کوچک‌تر تقسیم می‌کنیم و سپس اقدام به پیدا کردن راه‌حل برای هر قسمت می‌نماییم. در این پروژه از متدولوژی RUP برای کنترل فرایند تولید نرم‌افزار استفاده شده است.

^۱ Use case diagram

۳

فصل سوم اینترنت اشیا

اینترنت اشیا

تنها حدود دو دهه است که اینترنت راه خود را به خانه های مردم باز کرده است. با این وجود تغییری که در نحوه ارتباطات ایجاد کرده، شگرف بوده است. اینترنت عاداتهای روزمره ما همچون خرید، تماس با آشنایان، سفر و ... را دچار دگرگونی کرده است. در آینده نفوذ اینترنت به مراتب بیش از امروز خواهد بود و این دانشمندان و نظریه پردازان حوزه فناوری اطلاعات را به فکر ایده پردازی در اینترنت اشیا انداخته است.

اینترنت اشیا، حضور فراگیر اشیا متنوع همچون برچسب های RFID، حسگرها، عملگرها، گوشی های هوشمند و ... در اطراف ماست که با شبکه شدن و داشتن آدرس های منحصر بفرد قادرند با یکدیگر تعامل کرده و سرویس هایی را به ما ارائه دهند. [۵]

۱.۳ تاریخچه به وجود آمدن اینترنت اشیا

عبارت اینترنت اشیا، برای نخستین بار در سال ۱۹۹۹ توسط کوین اشتون مورد استفاده قرار گرفت و جهانی را توصیف کرد که در آن هر چیزی، از جمله اشیا بی جان، برای خود هویت دیجیتال داشته باشند و به کامپیوترها اجازه دهند آن ها را سازماندهی و مدیریت کنند. اینترنت در حال حاضر همه مردم را به هم متصل می کند و با اینترنت اشیا تمام اشیا به هم متصل می شوند. البته پیش از آن و در همان سال کوین کلی در کتاب قوانین نوین اقتصاد در عصر شبکه ها موضوع ابزارهای کوچک هوشمند مانند حسگر باز و بسته نمودن درب را (که به شبکه جهانی اینترنت متصل می باشند) مطرح کرد.

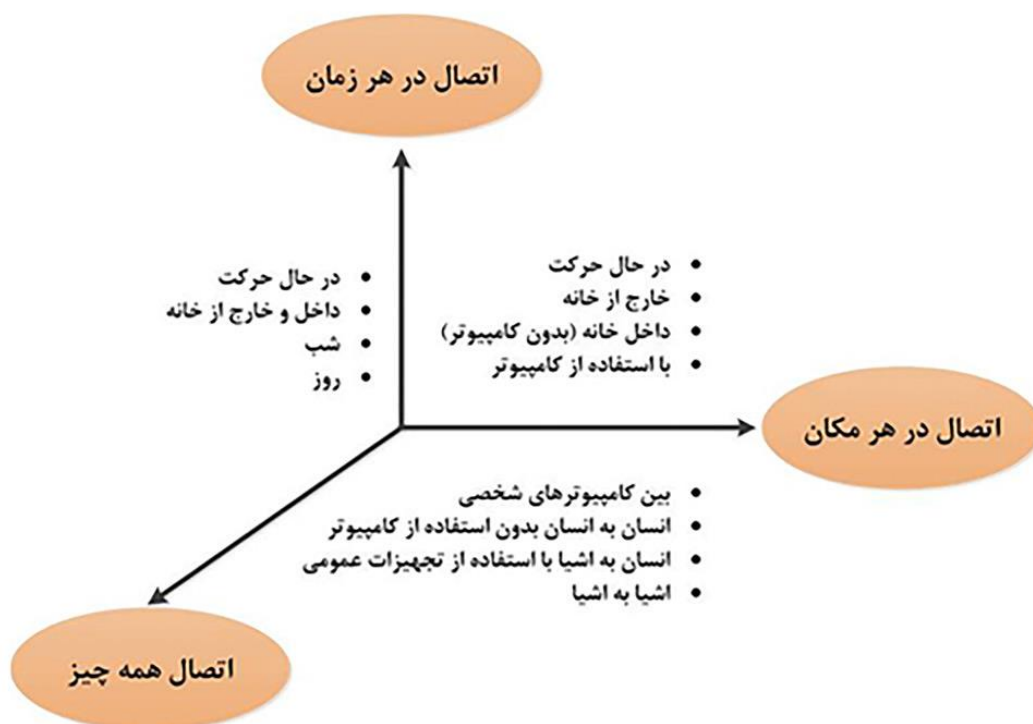
۲.۳ اینترنت اشیا چیست؟

اینترنت اشیا یا IoT^۱ مفهومی جدید در دنیای فناوری و ارتباطات است و به طور کلی به بسیاری از اشیا و وسایل محیط پیرامونمان اشاره دارد که به شبکه اینترنت متصل شده و می توان آنها را توسط اپلیکیشن های موجود در تلفن های هوشمند و تبلت کنترل و مدیریت کرد. اینترنت اشیا فناوری مدرنی

^۱ Internet of Things

است که در آن برای هر موجودی اعم از انسان، حیوان و یا اشیا قابلیت ارسال داده از طریق شبکه های ارتباطی (اینترنت و اینترنت) فراهم می شود و جهانی را توصیف می کند که در آن هر چیزی از جمله اشیای بی جان برای خود هویت دیجیتال دارند و به کامپیوتر ها اجازه می دهند آن ها را ساماندهی کنند.

تعریفی که اتحادیه بین المللی مخابرات از «اینترنت اشیا» دارد بدین صورت می باشد: در هر زمان، هر مکان، برای هر کسی از ما، اتصالی برای هر چیزی خواهیم داشت. شکل زیر نشان دهنده ابعاد مختلف «اینترنت اشیا» می باشد.



شکل ۱.۳ ابعاد مختلف اینترنت اشیا

۳.۳ معماری اینترنت اشیا

مدل ها و معماری های مختلفی برای فناوری اینترنت اشیا پیشنهاد شده است اما مدلی که در شکل ۳.۲ آورده شده است متداول ترین مدل پیشنهادیست که کلیان معماری این فناوری را به خوبی نمایش می دهد.

در ادامه لایه های مختلف فناوری اینترنت اشیا آورده و درباره هر قسمت توضیح داده شده است:

معماری چهار لایه‌ای	
لایه کاربرد	محیط زیست، شهر هوشمند، خانه هوشمند، انرژی، حمل و نقل و ...
لایه پردازش اطلاعات	داده کاوی، امنیت اطلاعات، تصمیم‌گیری، موتور جست‌وجو، مراکز داده و ...
لایه شبکه	Internet, WWAN, WPAN, WLAN, WMAN
لایه ادراک و شناسایی	حسگرها، دستگاه اندازه‌گیری، دستگاه هوشمند، RFID, GPS و ...

شکل ۲.۳ مدل چهار لایه‌ای اینترنت اشیا

همان‌طور که در شکل ۲/۳ مشخص است، اینترنت اشیا از چهار لایه کلی تشکیل شده است:

۱.۳.۳ لایه ادراک و شناسایی

این لایه مشابه لایه فیزیکی OSI است که از سنسورهای فیزیکی مختلف (برای مثال RFID, ZigBee, QR، مادون قرمز و ...)، دستگاه‌ها و امان‌های محیطی تشکیل شده است. در این لایه سنسورها اطلاعات را شناسایی و جمع‌آوری می‌کنند و به لایه بالاتر می‌فرستند. اطلاعات جمع‌آوری شده می‌تواند از جنس موقعیت، سرعت باد، لرزش، درجه PH، رطوبت، مقدار غبار موجود در هوا و ... باشد. این اطلاعات از طریق لایه شبکه به دلیل ارتباطات مطمئن آن به سیستم پردازش داده مرکزی منتقل می‌شوند. داده‌های حجیمی که در اینترنت اشیا تولید می‌شوند، از این لایه آغاز می‌شود.

۲.۳.۳ لایه شبکه

این لایه مشابه لایه شبکه OSI است که شامل دو نوع توانایی است:

- توانایی شبکه: کاربردهای مرتبط با کنترل اتصال شبکه را فراهم می‌نماید. مانند کاربرد کنترل دستیابی و مبدا انتقال، مدیریت تحرک یا احراز هویت، اجازه و حسابداری (AAA).

- توانایی انتقال: نقش مهمی در انتقال ایمن ایفا می کند و اطلاعات محرمانه حساس از حسگرها را پیش خود نگه داشته و از طریق ارتباط ۳G، ۴G، Wi-Fi، WiMAX، مادون قرمز، ماهواره و ... منتقل می نماید. این انتقال مستقل از نوع دستگاه حسگر، به سیستم پردازش داده مرکزی صورت می گیرد. به همین دلیل، مسئول انتقال اطلاعات از لایه ادراک به لایه بالاتر می باشد. اطلاعات مربوط به فناوری ارتباطی در شکل ۳/۳ قابل ملاحظه است.

فناوری ارتباطی بی-سیم	کاربردهای کلیدی	برد	حداکثر گذردهی
Cellular(۲G, ۳G, LTE)	اتصال همیشگی، نرخ داده بالا، قابلیت تحرک	۳۰ کیلومتر	۱Gbps
NFC	بازاریابی از طریق تلفن همراه، پرداخت از طریق تلفن همراه، حسگرهای پوشیدنی	۱۰ سانتی-متر	۲۰Kbps
Wi-Fi	وسیله نقلیه، خانه هوشمند، لوازم مصرفی الکترونیکی، بازاریابی از طریق تلفن همراه، لوازم خانگی، حسگرهای پوشیدنی	۱۰۰ تا ۳۰۰ متر	۳۰۰Mbps
Bluetooth	بازاریابی از طریق تلفن همراه، پرداخت از طریق تلفن همراه، خانه هوشمند، استفاده شخصی	۱۰ تا ۱۰۰ متر	۱Mbps
ZigBee	خانه هوشمند، لوازم مصرفی الکترونیکی، صنعت	۱۰ تا ۱۰۰ متر	۲۰Kbps-۲۵۰Kbps
RFID	عوارضی جاده‌ها، فروشگاه هوشمند	۱۰ تا ۲۰۰ سانتی متر	۴۰Kbps

شکل ۳.۳ فناوری‌های مختلف ارتباطی در اینترنت اشیا

۳.۳.۳ لایه پردازش اطلاعات

در این لایه امکانات متعددی به منظور مدیریت سرویس ارائه شده است. توجه به امکانات این لایه خصوصا از منظر معماری سرویس محور بسیار حائز اهمیت است چرا که بسیاری از برنامه‌ها در لایه کاربرد مستقیما تحت تاثیر این معماری قرار خواهند گرفت. وظایفی که برای این لایه تعریف می‌شود به صورت زیر است:

- مسئولیت ذخیره، تجزیه و تحلیل اطلاعات، کنترل امنیت، مدلسازی فرآیندها و مدیریت دستگاه‌ها
- داده‌هایی که به طور متناوب ارسال می‌گردد نیازمند فیلترینگ می‌باشند. چرا که حجم داده‌های بسیار زیاد است و لازم است با توجه به نوع خواسته، فرآیند فیلترینگ بر روی آن‌ها اعمال شود.
- با توجه به حجم بالای داده، نیاز به فرآیند استخراج اطلاعات وجود خواهد داشت تا بتوان بر اساس پردازش انجام شده به یک دید مناسب از داده‌های کلان دست یافت.

۴.۳.۳ لایه کاربرد

بالاترین لایه معماری در اینترنت اشیا که نزدیکترین لایه به خلق ارزش برای مشتری است. این لایه طیف گسترده‌ای از برنامه‌های کاربردی را شامل می‌شود که ممکن است مختص یک صنعت به خصوص (Vertical Market) و یا صنایع مختلف (Horizontal Market) باشند. مسئولیت این لایه ساخت مدل و گزارشات تجاری، گراف‌ها، فلوچارت‌ها و ... بر اساس داده‌های دریافتی از لایه پایین‌تر است.

موارد زیر در مورد این لایه قابل ذکر هستند:

- صنایع مختلف می‌توانند با استفاده از برنامه‌های متنوع از اینترنت اشیا برای بهبود خدمات خود استفاده نمایند.
- برخی از برنامه‌های اینترنت اشیا مختص استفاده فردی و یا به کارگیری در منازل طراحی و پیاده‌سازی می‌گردند. ابعاد این نوع برنامه‌ها کوچک است و صرفا قادر به تامین خواسته تعداد اندک و محدودی از کاربران می‌باشند (کاربرد اینترنت اشیا در خانه‌های هوشمند)

- برخی از برنامه‌های اینترنت اشیا در ابعاد بسیار بزرگی طراحی و پیاده‌سازی می‌شوند و می‌توان آنها را در سطح یک سازمان یا شهر و کشور به خدمت گرفت (کاربرد اینترنت اشیا در صنایع، کشاورزی، پزشکی و شهرهای هوشمند)

۴.۳ کاربردهای اینترنت اشیا

۱.۴.۳ سلامت الکترونیک

اینترنت اشیا در حوزه سلامت الکترونیک فرصت‌های زیادی ایجاد می‌کند. در واقع این فناوری می‌تواند خدمات سلامت را بهبود داده و منجر به نوآوری‌های مختلفی در این ارتباط شود. به طور مشخص با کمک IoT فرآیندهای مراقبت پزشکی آسان‌تر شده و خدمات درمانی از طریق هوشمندتر شدن سیستم، ارتقا می‌یابد. به طور مشخص IoT بستری برای ارتقا کیفی زندگی، تسهیل زندگی روزمره افراد معلول و مبتلا به بیماری‌های مزمن فراهم می‌آورد. همچنین این فناوری ارائه خدمات درمانی فراگیر و کم‌هزینه‌تر را ممکن می‌سازد.

خدمات نوینی که با بکارگیری اینترنت اشیا در حوزه سلامت قابل ارائه هستند، عبارتند از:

- جمع‌آوری داده‌های مرتبط با علائم حیاتی بیمار از طریق شبکه سنسورهای مثل به تجهیزات پزشکی و بدن بیمار
- ارسال داده‌های جمع‌آوری شده به بستر ابری مرکز پزشکی مربوطه (برای مثال، یک بیمارستان) برای ذخیره سازی و پردازش آن‌ها
- تحلیل و مدیریت اطلاعات فراهم شده از طریق سنسورها
- تضمین دسترسی فراگیر (از هر جا و در هر زمان) و به اشتراک‌گذاری داده‌های پزشکی

۲.۴.۳ شهر هوشمند

به طور خلاصه، در یک شهر از ICT (اطلاعات و فناوری‌های ارتباطی) برای دستیابی به اهداف زیر استفاده می‌شود:

- افزایش کیفیت و کارایی خدمات شهری

- کاهش هزینه ها و مصرف منابع
- تسهیل ارتباط مابین شهروندان و نهادهای حاکمیتی

در این راستا، اینترنت اشیا می‌تواند یک میان افزار مشترک برای توسعه سرویس‌های آینده‌گرا در حوزه شهرهای هوشمند، جمع‌آوری و تجمیع اطلاعات از سنسورهای مختلف و بعضا ناهمگن، دسترسی به انواع گوناگونی از فناوری‌های IoT و موقعیت‌یابی (همچون برچسب‌گذاری جغرافیایی و ارائه تصویر سه بعدی از طریق سنسورهای RFID) و عرضه اطلاعات به شکل یکدست و یکنواخت را فراهم آورد.

۳.۴.۳ خانه هوشمند

برای اینترنت اشیا مبتنی بر بستر ابری می‌توان کاربردهای زیادی در محیط خانه متصور شد. در واقع با ترکیب قابلیت‌های تجهیزات نهفته در محیط خانه و بستری ابری، انجام خودکار بسیاری از فعالیت‌های خانگی وجود دارد. همچنین امکان اتصال به وسایل منزل از طریق اینترنت به منظور مانیتورینگ رفتار آنها از راه دور (برای مثال مانیتورینگ میزان انرژی مصرفی وسایل منزل و بکارگیری این اطلاعات برای بهبود الگوی مصرف) یا کنترل آنها از راه دور (برای مثال مدیریت هوشمند روشنایی، گرما و تهویه هوا) وجود دارد. قابل ذکر است که ۱۹ درصد مصرف برق خانگی متعلق به روشنایی است که از طریق مدیریت هوشمند ۴۵ درصد کاهش خواهد یافت.

۴.۴.۳ مدیریت هوشمند انرژی

ترکیب بستر ابری و IoT کاربردهای متنوعی را در حوزه مدیریت هوشمند توزیع انرژی ممکن می‌سازد. اغلب توان حسگری، پردازشی و قابلیت‌های شبکه‌ای سنسورهای این حوزه محدود است. لذا برای پردازش اطلاعات و اتخاذ تصمیمات جامع، قابلیت‌های بستر ابری هستیم. ناهمگونی سنسورها، اندازه بزرگ داده، نرخ جمع‌آوری داده، تاخیر متغیر، یکپارچه‌سازی داده‌های جمع‌آوری شده از منابع مختلف با مالکیت متفاوت، امنیت و حریم خصوصی از چالش‌های این حوزه هستند.

۵.۴.۳ مانیتورینگ محیطی

ترکیب بستر ابری و IoT می‌تواند به توسعه سریع‌تر کاربردهای متنوع مانیتورینگ محیطی همچون اندازه‌گیری سطح آب، میزان آلودگی هوا، رطوبت خاک، شرایط نور محیط، تشخیص وقوع آتش‌سوزی و ردیابی حیوانات اشاره کرد. همچنین می‌توان به مانیتورینگ امنیت غذایی، آبیاری قطره‌ای، محافظت و نگهداری از درختان و جنگل‌ها اشاره کرد. چالش‌های اصلی این حوزه شرایط محیطی متغیر و امنیت سنسورهای محیطی (که از نظر فیزیکی هم در معرض خطر هستند) است.

۶.۴.۳ تدارکات هوشمند

فناوری‌های IoT مبتنی بر بستر ابری، سناریوهای جذاب جدیدی را در حوزه تدارکات هوشمند معرفی می‌کنند. به طور کلی مدیریت آسان و خودکار زمان، هزینه و مسیر انتقال کالاها از مبدا تا مقصد (به کمک فناوری‌های موقعیت‌یابی) را فراهم می‌آورند. چالش این حوزه عمدتاً مرتبط با ناهمگون بودن منابع که راهکارهای آن مجازی‌سازی و انتخاب سرویس می‌باشد.

۵.۳ چالش‌های پیش روی اینترنت اشیا

با وجود کاربردها و امکانات فراوانی که فناوری اینترنت اشیا در اختیار جامعه قرار می‌دهد، چالش‌هایی را نیز به وجود خواهد آورد که برطرف کردن آن‌ها می‌تواند پیشرفت این فناوری را تسریع بخشد. چالش‌هایی از جمله پردازش داده‌های کلان، امنیت شبکه‌ها و دستگاه‌ها، حریم خصوصی کاربران، سازگاری بین تجهیزات و ...

۱.۵.۳ پردازش داده‌های کلان

از نظر مقیاس پذیری برنامه‌های کاربردی IoT به تعداد زیادی از دستگاه نیاز دارد که پیاده‌سازی آنها به دلیل محدودیت‌های زمان، حافظه و پردازش مشکل است. به عنوان مثال محاسبه تغییرات روزانه دمایی در محدوده یک کشور به دستگاه‌های زیادی نیازمند است و پردازش اطلاعاتی زیادی را می‌طلبد.

۲.۵.۳ امنیت

توزیع‌شدگی بیشتر شبکه و به تبع آن نقاط ورود به سیستم، یکی از چالش‌های اساسی است. همچنین اشیایی که قرار است به اینترنت متصل شوند، معمولاً ساختار و معماری ساده‌تری نسبت به کامپیوترها دارند و این موضوع فراهم کردن امنیت در آنها را دشوار می‌سازد. در حقیقت می‌توان گفت اینترنت اشیا خیلی بیشتر از اینترنت فعلی به زندگی واقعی نزدیک شده است و نفوذ به چنین شبکه‌ای معادل نفوذ به زندگی روزمره کاربران خواهد بود.

۳.۵.۳ حریم خصوصی

مفهوم حریم خصوصی همواره همراه با امنیت به کار برده شده است، اما در این بحث مناسب است که توجه جداگانه‌ای به آن شود، چرا که در اینترنت اشیا اطلاعات خصوصی بیشتری نسبت به وضعیت کنونی بر روی شبکه قرار می‌گیرد. یکی از این اطلاعات خصوصی، سبک زندگی افراد است. این است که چه ساعتی را در خانه به سر می‌بریم، چه فیلم‌هایی تماشا می‌کنیم و حتی اینکه چه غذایی می‌خوریم!

۴.۵.۳ سازگاری بین تجهیزات

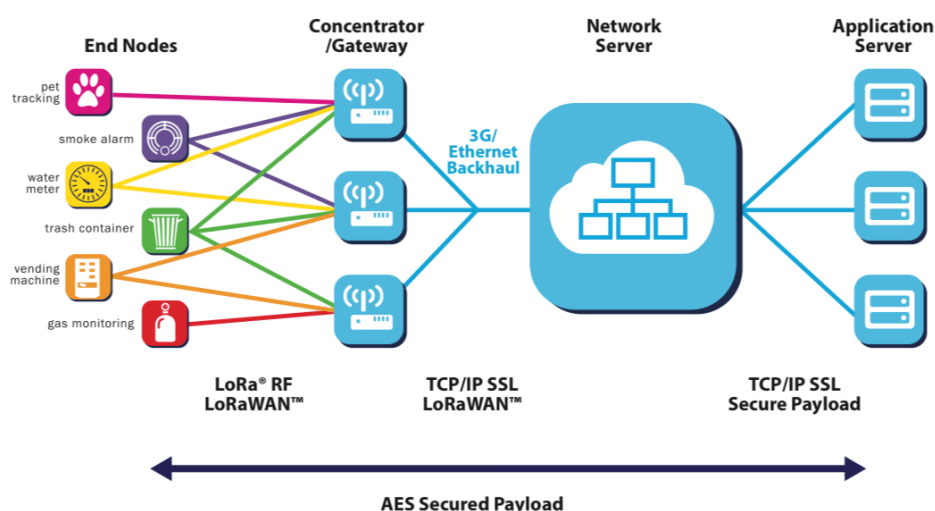
افزایش تقاضا برای خرید تجهیزات با قابلیت اتصال به شبکه، افزایش رقابت را در این حوزه به دنبال خواهد داشت. تولیدکنندگان و خدمات‌دهندگان در سراسر دنیا، پروتکل‌ها، پلتفرم‌ها و فناوری‌های خودشان را عرضه خواهند نمود و در نتیجه ممکن است سازگاری بین تجهیزات مختلف دچار مشکل شود. از آن جایی که هنوز ساختار IoT در حال تکامل است، ورود نهادهای استانداردسازی از جمله IEEE نیز ممکن است پیشرفت در این حوزه را محدود سازد.

۶.۳ LoRaWAN^۱

شبکه لورا یکی از پروتکل‌های اصلی دوربرد توان پایین^۲ ویژه اینترنت اشیا است. این فناوری به سیگنال‌ها اجازه می‌دهد تا حتی در سطوح پایین‌تر از نویز نیز منتشر و بازیابی شوند. تجهیزات مبتنی بر شبکه لورا می‌توانند تا سال‌ها فقط با یک باتری کار کنند. شاید مهمترین مشخصه شبکه لورا که توانسته است در کنار مزیت‌های فنی این پروتکل زمینه رشد سریع آن را فراهم کند، رویکرد غیر انحصاری توسعه این پروتکل بر بستر یک جامعه آزاد و با مشارکت مجموعه‌های مختلف فناوری باشد.

۱.۶.۳ معماری فنی شبکه لورا

همان‌طور که اشاره شد، شبکه لورا یک پروتکل ارتباطی LPWAN ویژه اینترنت اشیا در باندهای فرکانسی بدون نیاز به مجوز (ISM) است که می‌تواند محدوده وسیعی را با توان مصرفی پایین تحت پوشش قرار دهد. این فناوری توسط شرکت Semtech و جامعه‌ای از شرکت‌های بزرگ حوزه فناوری همچون (IBM, Cisco, HP, Foxconn) که LoRa Alliance نام دارد، توسعه یافته و پشتیبانی می‌شود.



شکل ۴.۳ معماری ساختار شبکه لورا

^۱ Long Range Wide Area Network

^۲ Low Power Wide Area Network

معماری ساختار یک شبکه لورا همان طور که در شکل بالا آمده است از دستگاه‌های انتهایی مبتنی بر لورا (سنسورها و عملگرها که اصطلاحاً End-Device خوانده می‌شوند)، گذرگاه‌ها، سرور شبکه و نهایتاً اپلیکیشن و نرم‌افزار کاربر تشکیل شده است. توپولوژی شبکه لورا به صورت ستاره‌ای^۱ است. دستگاه‌های انتهایی اطلاعات را به از طریق شبکه لورابه گذرگاه ارسال می‌کنند. پس از دریافت داده توسط گذرگاه، گذرگاه اطلاعات را بر روی یک لینک ارتباطی مبتنی بر اینترنت به سمت سرور شبکه می‌فرستد. این لینک ارتباطی می‌تواند توسط شبکه ۳G/LTE، Ethernet و یا شبکه‌های داخلی طراحی شود. سپس اطلاعات توسط سرور شبکه در اختیار نرم‌افزار کاربران قرار می‌گیرد. در حقیقت گذرگاه و سرور شبکه مانند یک واسطه بین نرم‌افزار کاربر و دستگاه‌های انتهایی عمل می‌کند و امکان رسیدن داده به نرم‌افزار را فراهم کند. در شبکه لورا داده‌ها به صورت کامل (End-to-End) بین دستگاه‌ها و اپلیکیشن کاربر از طریق رمزگذاری AES ارسال می‌شود. از این رو امنیت اطلاعات کاربران نیز تضمین می‌شود.

فناوری LoRaWAN با بکارگیری لینک متقارن، امکان ارتباط کاملاً دو سویه را فراهم می‌کند؛ این مساله به ویژه در سرویس‌های اینترنت اشیا که نیاز به ارسال دستورهای کنترلی از سمت سرور به تجهیزات انتهایی را دارند، بسیار با اهمیت است.

در شبکه لورا نرخ ارسال داده مبتنی بر پروتکل لایه فیزیکی LoRa ۲۷ kb/s است و هر گذرگاه می‌تواند داده‌های هزاران دستگاه انتهایی را جمع‌آوری کند. همچنین پوشش رادیویی هر گذرگاه شبکه لورا در مناطق باز و حومه شهر تا ۱۵ کیلومتر نیز می‌رسد.

۲.۶.۳ لایه فیزیکی و مدولاسیون لورا

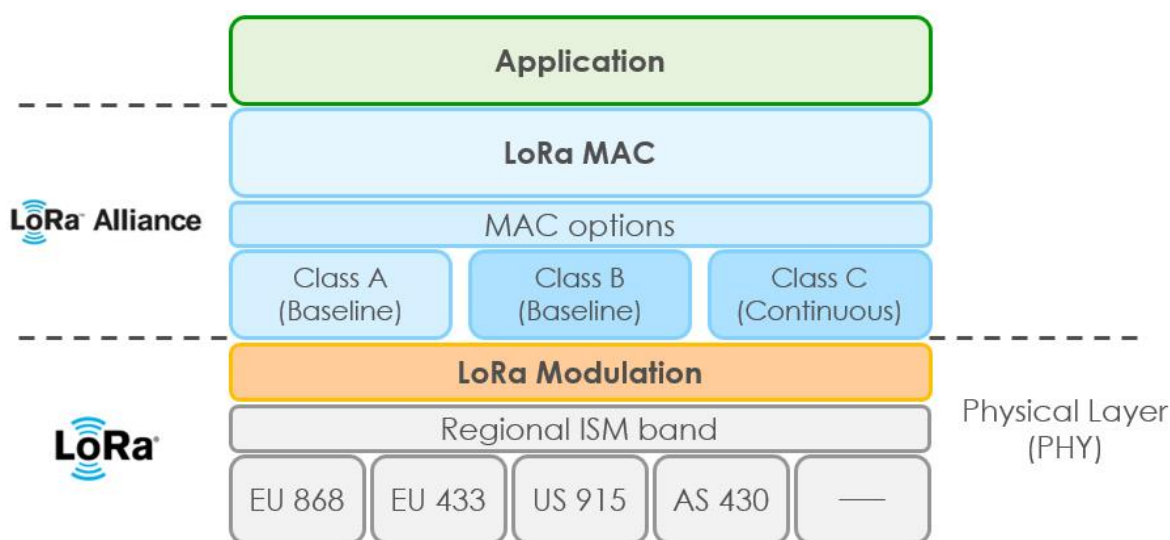
اگرچه در بسیاری موارد شبکه لورا در کلام بطور مختصر لورا خوانده می‌شود، اما از نظر فنی، این دو متفاوت هستند. لورا پروتکل لایه فیزیکی یا مدولاسیون بیسیمی است که به منظور ایجاد لینک ارتباطی با ناحیه پوشش وسیع استفاده می‌شود. بسیاری از سیستم‌های پیشین به منظور رسیدن به توان پایین از مدولاسیون^۲ در لایه فیزیکی بهره می‌بردند. اما لورا مبتنی بر مدولاسیون CSS^۱ است که علاوه بر فراهم

^۱ Star of Stars

^۲ FSK Frequency Shift Keying

آوردن خاصیت توان پایین مدولاسیون FSK، ناحیه پوشش و نفوذپذیری را نیز به طور قابل توجهی افزایش می‌دهد. چندین دهه چنین مدولاسیونی به دلیل مقاوم بودن در برابر تداخل^۲ و ناحیه پوشش وسیع، تنها در کاربردهای نظامی و ارتباطات فضایی استفاده می‌شد. اما لورا اولین پیاده‌سازی کم هزینه و مناسب برای کاربردهای تجاری از چنین سیستمی است که در باندهای بدون نیاز به مجوز کار می‌کند.

پروتکل شبکه لورا در حقیقت پروتکلی در لایه بالاتر (MAC) است که بر پایه پروتکل لورا توسعه یافته و کار می‌کند و امکان راه‌اندازی یک شبکه کامل را فراهم می‌سازد.



شکل ۵.۳ جایگاه لورا در معماری لایه‌ای

۳.۶.۳ مدل توسعه شبکه لورا

از نظر مدل توسعه، شبکه لورا در مقابل شبکه SigFox استراتژی کاملاً متفاوتی را اتخاذ کرده و مشارکت در تمام قسمت‌ها باز و ممکن است. عضویت در LoRa Alliance و مشارکت در توسعه و استفاده از استانداردهای این فناوری برای همه امکان‌پذیر است. هر شرکت سخت‌افزاری می‌تواند دستگاه‌های انتهایی و گذرگاه را مطابق با استانداردهای شبکه لورا تولید کند. حتی تولید ماژول‌های

^۱ Chirp Spread Spectrum

^۲ Interference

رادیویی (لایه فیزیکی) که تا دو سال پیش تنها توسط Semtech انجام می‌گرفت، با فروش license به کمپانی‌های NXP و Microchip، از انحصار یک کمپانی خاص خارج شده است.

از این رو شبکه لورا راهبرد توسعه بسیار منعطفی را پیش گرفته که در نتیجه آن به توسعه شبکه به یک شرکت خاص وابسته نیست. همین امر رشد این پروتکل را سرعت بخشیده است و علی‌رغم شروع دیرتر نسبت به SigFox هم‌اکنون در مناطق بیشتری پوشش رادیویی دارد. شبکه لورا تاکنون (سپتامبر ۲۰۱۸) در ۹۵ کشور وجود دارد و این روند توسعه در آینده سیاره‌ای هوشمند خواهد ساخت. همچنین مدل‌های متنوعی نیز از شبکه‌های کاملاً خصوصی و خارج از بستر اینترنت، تا شبکه‌های عمومی با طرح‌های تجاری مختلف، بر بستر این فناوری شکل گرفته است. از سوی دیگر اکوسیستم باز شبکه لورا موجب شده است که این فناوری در بخش فنی نیز به سرعت توسعه و در این زمینه نسبت به سایر پروتکل‌ها پیشی گیرد [۶].

۷.۳ جمع‌بندی

اینترنت اشیا مبحث تقریباً جدیدی است که کاربردهای فراوانی در جهت بهبود کیفیت زندگی در جوامع امروزی دارد. حرکت به سمت پیاده‌سازی بسترهای اینترنت اشیا در شهرهای مدرن و استفاده از آن در تمام جنبه‌های زندگی امری اجتناب‌ناپذیر است. البته چالش‌های فراوانی نیز در این راه وجود دارد که مهندسی و مدیران باید برای حل آنها چاره‌اندیشی نمایند.

۴

فصل چهارم پیاده‌سازی

پیاده‌سازی

در ابتدا این سوال مطرح می‌شود که «چه نیازی به پیاده‌سازی یک اپلیکیشن مستقل وجود دارد؟» در ادامه به اختصار به مزایای این نوع پیاده‌سازی نسبت به پیاده‌سازی واکنش‌گرا^۱ نسخه سابق (نسخه وب) می‌پردازیم:

- تجربه کاربری بهتر

از مهم‌ترین دلایل این انتخاب، راحتی کاربر در ارتباط با کارکردن با اپلیکیشن‌های مستقل است؛ زیرا این برنامه‌ها بسیار سریع، قابل اتکا، غنی از لحاظ بصری و منعطف از لحاظ یادگیری هستند و امکانی مثل ارسال اعلان^۲ در آن وجود دارد.

- گرافیک قوی

این نوع از توسعه، اپلیکیشن‌هایی با گرافیک سریع‌تر، انیمیشن‌های سیال و انتقال‌های نرم را در بر دارد.

- امنیت بیشتر

امکان پیاده‌سازی ساده‌تر احراز هویت دوعامله^۳، دسترسی به ویژگی‌های داخلی امنیتی دستگاه و اعمال گواهی^۴ موجود است.

- ارائه بهتر امکانات مشترک

در برخی موارد، هر دو نوع توسعه، امکاناتی را در اختیار ما می‌گذارند اما کیفیت این امکانات در توسعه مستقل بهتر است برای مثال می‌توان به دسترسی به سخت‌افزار و ذخیره‌سازی در حافظه موقت پنهانی اشاره کرد.

^۱ Responsive

^۲ Notification

^۳ two-factor authentication

^۴ Certificate pinning

حال پس از مشخص شدن نیازمندی‌ها وارد فاز تحلیل، طراحی و پیاده‌سازی نرم‌افزار می‌شویم. ابتدا توضیح مختصری در مورد معماری ساختار نرم‌افزار ارائه می‌گردد و پس از آن ساختار جداول پایگاه داده مورد بررسی قرار خواهد گرفت.

۱.۴ معماری نرم‌افزار

معماری نرم‌افزار از کلیدی‌ترین بخش‌های تولید نرم‌افزار است. معماری نرم‌افزار در واقع انتخاب یک ساختار کلی برای پیاده‌سازی یک پروژه نرم‌افزاری بر مبنای مجموعه‌ای از نیازهای کاربری و تجاری یک سیستم نرم‌افزاری است؛ به عبارت دیگر معماری نرم‌افزار در یک سیستم محاسباتی عبارت است از ساختار یا ساختارهای سیستم که شامل اجزای نرم‌افزاری و ارتباطات میان آنها می‌باشد. این پروژه و بخش مربوط به رابط‌های برنامه‌نویسی کاربردی تحت معماری MVC^۱ هستند.

۱.۱.۴ معماری MVC

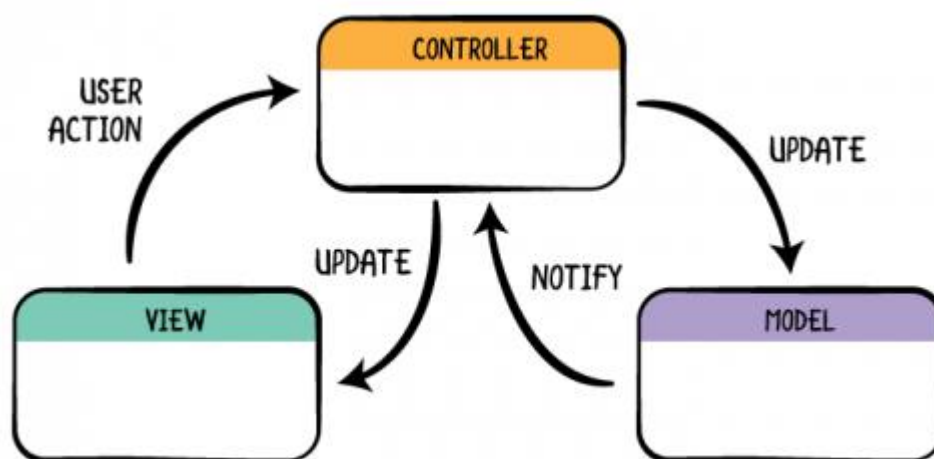
MVC یک الگوی طراحی معماری است که از سه مولفه Model، View و Controller تشکیل شده است. در واقع MVC معماری چندلایه‌ای جهت جداسازی قسمت‌های مختلف برنامه و به طور دقیق‌تر جداکردن بخش‌های منطقی برنامه از لایه نمایش یا در واقع همان لایه‌ای که مستقیماً با کاربر نهایی در ارتباط است، قرار می‌گیرد. [۷] پس بر اساس توضیحات بالا می‌توانیم هر یک از بخش‌های معماری MVC را تعریف کنیم:

Model در واقع بار اصلی معماری MVC را برعهده دارد و منطق تجاری نرم‌افزار را نگهداری می‌کند. هر آنچه که به نحوی مرتبط با ذخیره‌سازی اطلاعات یا دسترسی به منابع اطلاعاتی داخلی یا خارجی باشد در این بخش قرار می‌گیرد. منظور از داده الزاماً ارتباط با پایگاه‌های داده همچون MySQL و MongoDB و ... نیست، منبع داده‌ها در بخش Model می‌تواند یک آرایه از اعداد و یا هرچیز دیگری باشد.

^۱ Model-View-Controller

View قسمتی است که تمامی ارتباط کاربر با برنامه از طریق آن برقرار می‌شود. این بخش می‌تواند مستقیماً در قالب کد HTML، CSS و جاوااسکریپت خلاصه شود و یا مانند پروژه حاضر اپلیکیشن موبایلی باشد که دورادور در ارتباط است.

Controller در واقع واسطی بین دو بخش Model و View می‌باشد و همانطور که از نام آن مشخص است یک بخش کنترل‌کننده است. نحوه عملکرد این مولفه به این صورت است که بخش View درخواست‌ها را به بخش Controller فرستاده و این بخش با برقراری ارتباط با بخش Model درخواست‌های کاربر را پردازش کرده و پس از پایان پردازش زمانی که خروجی درخواست داده شده آماده گردید بخش View را آگاه می‌سازد تا خود را بر اساس تغییرات جدید به روز سازد.



شکل ۱.۴ ارتباط سه مولفه MVC با یکدیگر

۲.۴ زبان توسعه

۱.۲.۴ دارت^۱

دارت زبان برنامه‌نویسی است که توسط گوگل توسعه داده می‌شود. هدف دارت جایگزین کردن جاوااسکریپت که زبان داخلی مرورگرهای وب است، می‌باشد. دارت راه حلی برای مشکلات موجود در

^۱ Dart

جاوا اسکریپت (به‌طور مثال مشکل حافظه) می‌باشد که کارایی بهتر، قابلیت استفاده ساده‌تر برای پروژه‌های بزرگ و امنیت بیشتری را فراهم می‌کند. گوگل همچنین بسیار تلاش دارد تا دارت را پیچیده‌تر بسازد و ویژگی‌ها و قابلیت‌های فراوانی به آن ببخشد.

دارت زبانی برپایه کلاس، وراثت یگانه و شی‌گرایی است که گرامر آن شبیه زبان C بوده و دارای Interface، reified generics، Abstract و Optional typing می‌باشد. Type annotation های ایستا تأثیری بر روی سمانتیک کد در زمان اجرا ندارد. در عوض می‌تواند پرونده‌هایی برای ابزارهایی مانند checker های ایستا و check های پویای در زمان اجرا را فراهم نماید. این پروژه توسط لارس بک و کسپر لاند پایه‌گذاری شده است. [۸]

۲.۲.۴ PHP

زبان توسعه در بخش بک‌اند^۱، زبان PHP است.

۳.۴ چارچوب^۲ توسعه

یک چارچوب عبارت است از مجموعه‌ای از کلاس‌ها که با ترکیب و کار در کنار هم یک طراحی با قابلیت استفاده مجدد برای دسته‌ای خاص از نرم‌افزار ایجاد می‌کنند.

۱.۳.۴ فلاتر^۳

فلاتر یک چارچوب منبع‌باز^۴ برای توسعه‌ی اپلیکیشن‌های موبایل است که سال ۲۰۱۷ توسط گوگل ایجاد شد. با کمک این چارچوب می‌توان مشترکاً اپلیکیشن‌های متناسب با سیستم عامل اندروید^۱ و iOS تولید کرد. [۹]

Back end ^۱

Framework ^۲

Flutter ^۳

Open-source ^۴

مولفه^۲های اساسی فلاتر عبارتند از:

۱.۱.۳.۴ پلتفرم^۳ دارت

از آنجایی که اپلیکیشن‌های فلاتر با زبان دارت نوشته می‌شوند، از بسیاری از امکانات پیشرفته-ی این زبان استفاده می‌کنند. در سیستم‌عامل‌های اندروید، ویندوز^۴، مک^۵ و لینوکس^۶ فلاتر در یک ماشین مجازی دارت اجرا می‌شود که امکان اجرای در لحظه^۷ را به آن می‌دهد اما در سیستم عامل iOS به صورت کامپایل جلوتر از زمان^۸ اجرا می‌شود. یکی از مهم‌ترین ویژگی‌های پلتفرم دارت، اجرای مجدد داغ^۹ است که قابلیت ذخیره وضعیت^{۱۰} فعلی اپلیکیشن و اعمال تغییرات جدید روی آن را دارد بنابراین نیاز نیست در هر بار بررسی تغییرات، اپلیکیشن مجدداً به وضعیت اجرای اولیه بازگردد.

۲.۱.۳.۴ موتور فلاتر^{۱۱}

موتور فلاتر عمدتاً به زبان C++ نوشته شده است. این موتور کتابخانه‌های هسته‌ی فلاتر، شامل انیمیشن^۱ و گرافیک، فایل و شبکه ورودی/خروجی، حمایت دسترسی^۲ و معماری پلاگین^۳ را داراست.

Android ^۱

Component ^۲

platform ^۳

windows ^۴

macOS ^۵

linux ^۶

Just-in-time ^۷

Ahead-of-time Compilation ^۸

Hot reload ^۹

state ^{۱۰}

Flutter engine ^{۱۱}

۳.۱.۳.۴ کتابخانه‌های اساسی

همه‌ی کلاس‌های پایه‌ای نوشته شده در زبان دارت که برای ساخت اپلیکیشن‌های فلاتر لازم است مانند رابط‌های برنامه‌نویسی کاربردی برای ارتباط با موتور، در این کتابخانه‌ها موجودند.

۴.۱.۳.۴ ویجت‌های پیش‌طراحی‌شده

همه‌ی قسمت‌های ساده‌ی ظاهری مانند گرافیکی‌ها، متن، اشکال هندسی، انیمیشن، تصاویر و ... به صورت ویجت‌های از پیش مشخص‌شده موجود هستند و برای ساخت موجودیت‌های پیچیده‌ی دلخواه، کفایت این ویجت‌ها را با یکدیگر ترکیب کنیم.

۲.۳.۴ لاراول^۵

لاراول یک چارچوب متن‌باز PHP است که توسط تیلور اتول، برای توسعه نرم‌افزارهای وب بر پایه معماری MVC طراحی شده‌است. بعضی از ویژگی‌های لاراول به صورت ماژولار و بسته‌بندی شده ارائه شده‌است. در مارس ۲۰۱۵، لاراول به عنوان معروف‌ترین چارچوب php شناخته شد.

۴.۴ مدل (پایگاه‌های اطلاعاتی)

هر سیستم نرم‌افزاری بنابر نیازمندی‌های خود، از انواع پایگاه‌های اطلاعاتی برای ذخیره و بازیابی اطلاعات مورد نیاز خود بهره می‌برد. در این پروژه از پایگاه داده‌ی MySQL استفاده شده است. در ادامه به معرفی جدول‌های اصلی خواهیم پرداخت:

^۱ animation

^۲ accessibility support

^۳ plugin architecture

^۴ widget

^۵ Laravel

Users: اطلاعات مربوط به تمامی کاربران سیستم، اعم از کاربران عادی، مدیران و ...

Projects: اطلاعات مربوط به پروژه‌ها

Things: اطلاعات مربوط اشیا متصل به پروژه‌ها

ThingProfiles: اطلاعات مربوط به پروفایل اشیا

Gateways: اطلاعات مربوط به گذرگاه‌ها

Packages: بسته‌هایی که کاربر با خرید هر کدام می‌تواند مقدار مشخصی پروژه/شی بسازد.

View ۵.۴

آن قسمت از نرم‌افزار که به کاربر نشان داده می‌شود و باعث تعامل وی می‌شود view نام دارد که بخش دوم معماری MVC به حساب می‌آید. اگر بخواهیم دقیق‌تر بیان کنیم، همین قسمت که خود بخشی از یک معماری MVC بزرگ است، متشکل از یک معماری MVC می‌باشد که در ادامه به آن خواهیم پرداخت.

۱.۵.۴ معرفی معماری MVC درون اپلیکیشن

۱.۱.۵.۴ مدل

در این اپلیکیشن نیز لازم است مدل‌های مربوط به API وجود داشته باشند تا پردازش‌ها بتوانند بر روی آن‌ها به درستی انجام شده و با فرمت مناسب نمایش داده شوند. این مدل‌ها در قالب کلاس‌های دات که دارای ویجت مخصوص خود هستند، وجود دارند که لیست آنها قابل مشاهده است:

- Root
- User_login_page
- Login_failed
- Profile

- Projects
- Project_create
- Project_manage
- Project_details
- Project_view
- Thing
- Thing_profile
- Thing_profile_create
- Thing_project
- Thing_view
- Gateway
- Cart
- Current_package
- Buy_package

۲.۱.۵.۴ کنترلر

در تمامی کلاس‌های بالا، بخشی وجود دارد که کنترل مدل در دست آن است، برای مثال کلاس Root کنترل می‌کند که اگر کاربر قبلاً ورود نکرده است صفحه‌ی ورود (User_login_page) نمایش داده شود در غیر این صورت اطلاعات نشانه کاربر را بردارد و به صفحه پروفایل (Profile) منتقل کند. در ادامه به معرفی صفحات اپلیکیشن می‌پردازیم:

۳.۱.۵.۴ بخش ظاهری اپلیکیشن موبایل

در بخش بعدی به صورت کامل به معرفی این بخش می‌پردازیم

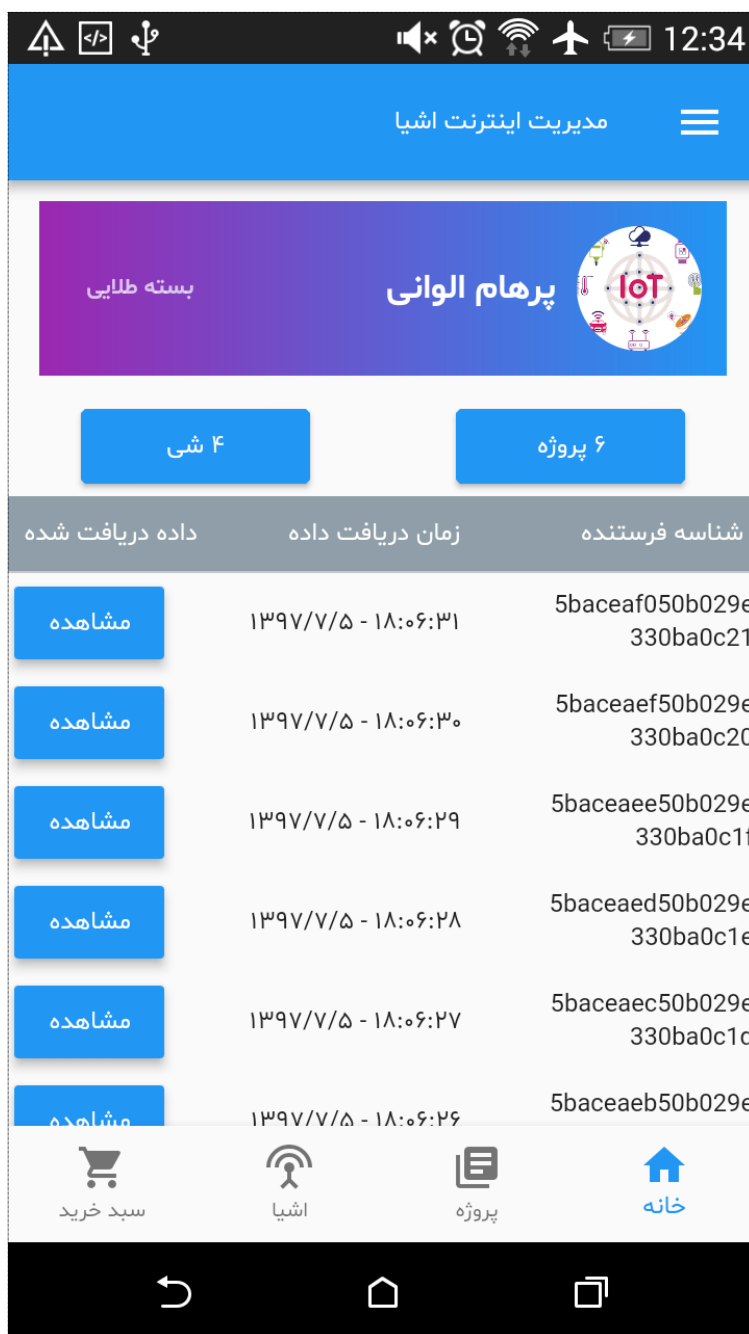
۲.۵.۴ صفحه‌ی احراز هویت

کاربر برای ورود به سامانه، باید ایمیل و رمز عبور خود را وارد نماید. پس از فشردن دکمه ورود، یک درخواست برای بخشی از API که به عنوان مرکز اعطای نشانه مشخص شده است، ارسال می‌شود. این مرکز تطابق ایمیل و رمز عبور دریافتی را با داده‌های موجود در جدول کاربران پایگاه داده بررسی کرده و در صورت تطابق، یک نشانه جدید برای کاربر تولید کرده و آن را برای اپلیکیشن موبایل ارسال می‌کند:

شکل ۲.۴ صفحه ورود به سامانه

۳.۵.۴ صفحه خانه

کاربر پس از ورود موفق به سامانه، به این صفحه منتقل می‌شود که اطلاعات خود از جمله نام، تصویر، بسته‌ی فعال، تعداد پروژه‌ها و تعداد اشیاء متعلق به خود در آن قابل رویت است. همچنین آخرین داده‌های دریافتی به همراه زمان دقیق و داده خام در این صفحه نمایش داده می‌شود:



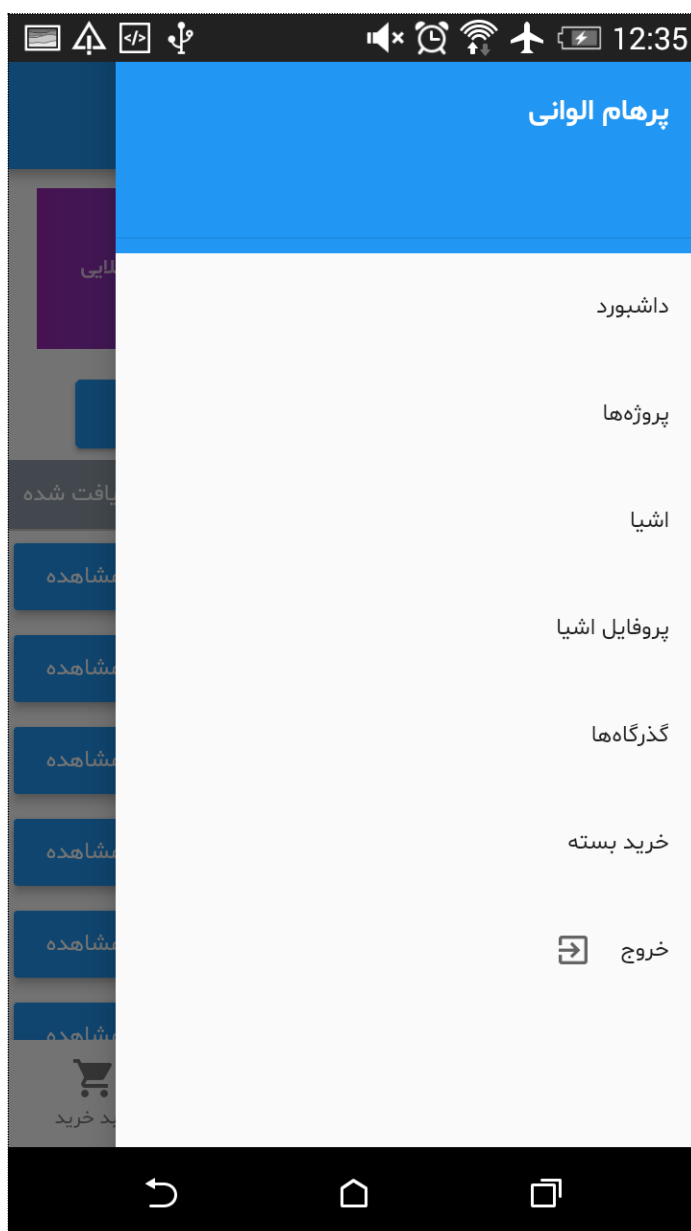
شکل ۳.۴ صفحه خانه



شکل ۴.۴ نمایش داده خام دریافتی

۴.۵.۴ منوی دراور^۱

در طرف راست یک منو برای دسترسی سریع تعبیه شده تا کاربر در هر صفحه‌ای بود بتواند سریعاً به تمامی صفحات برود:



شکل ۵.۴ منو دراور

^۱ Drawer

۵.۵.۴ پروژه‌ها

پروژه‌های نرم‌افزار ویژگی‌ها و عملیات مختص خود را دارند. هر پروژه دارای تعدادی شی است که وظیفه‌ی تبادل داده را برعهده دارند:

۱.۵.۵.۴ لیست پروژه‌های دریافتی



نام پروژه	وضعیت	امکانات
سنجش رطوبت	فعال	عملیات
Weather Station	فعال	عملیات
Car management	فعال	عملیات
Temperature management	فعال	عملیات
sarah	فعال	عملیات
d1	فعال	+

شکل ۶.۴ همه‌ی پروژه‌های کاربر

۲.۵.۵.۴ منو عملیات‌های مدیریتی

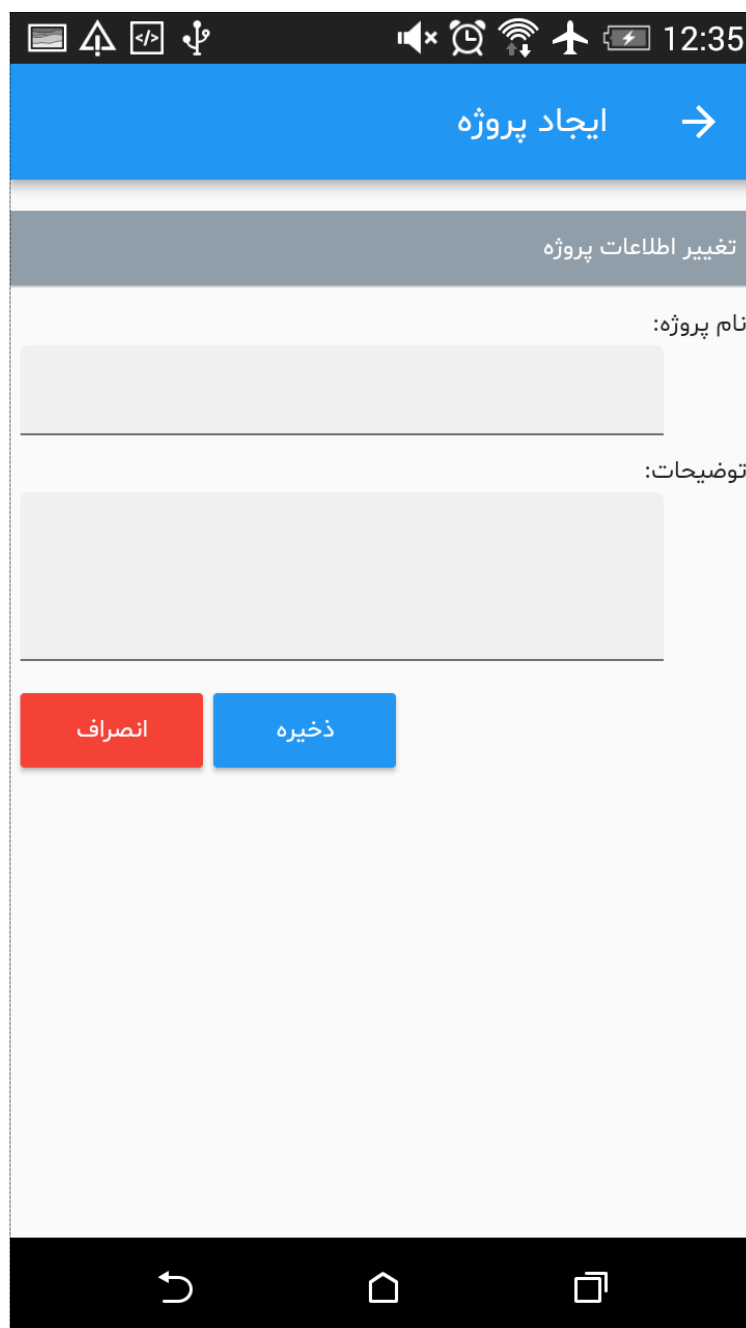
برای هر پروژه، یک منو برای انتخاب صفحه نمایش، مدیریت و حذف پروژه وجود دارد و همچنین می‌توان با کمک دکمه بعلاوه(+) یک پروژه جدید ساخت:



شکل ۷.۴ منو عملیات‌های مدیریتی

۳.۵.۵.۴ ایجاد پروژه

در این صفحه با نوشتن نام و توضیحات پروژه می‌توان پروژه‌ی جدید ایجاد کرد و پس از آن می‌توان اشیا مدنظر را به پروژه متصل نمود :



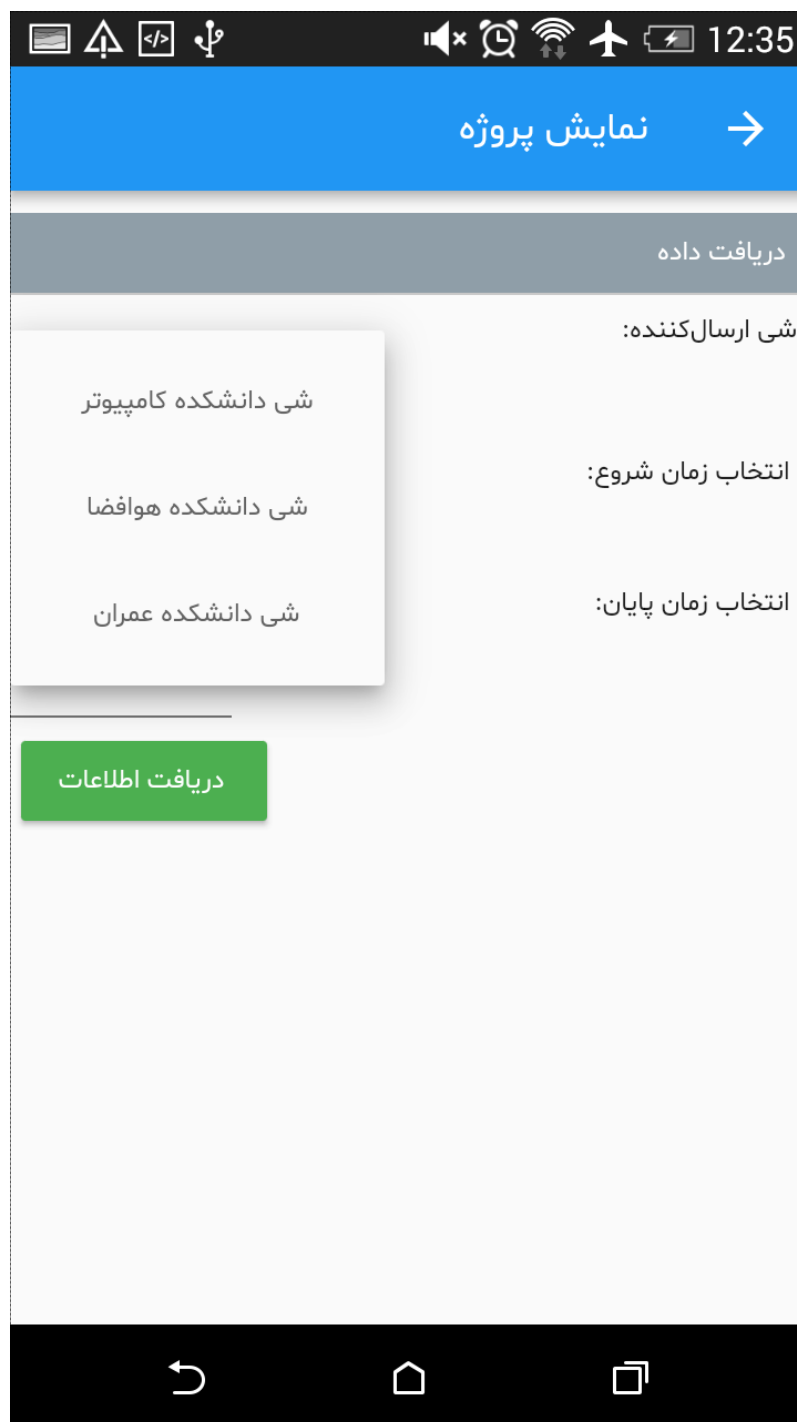
شکل ۸.۴ ایجاد پروژه

۴.۵.۵.۴ نمایش پروژه

در این بخش با انتخاب شی از لیست اشیا پروژه و بازه زمانی شروع و پایان مورد نظر می‌توان داده‌های خام و زمان دقیق دریافت آن‌ها را مشاهده کرد:

شکل ۹.۴ نمایش پروژه

شی از میان اشیا همان پروژه انتخاب می‌شود:



شکل ۱۰.۴ انتخاب شی در نمایش پروژه

بازه زمانی داده‌ها را از تقویم جلالی مشخص می‌کنیم اما برای ارسال به API، این تاریخ‌ها باید به برچسب‌زمان^۱ تبدیل شوند:



شکل ۱۱.۴ انتخاب زمان آغاز و پایان از تقویم جلالی

^۱ timestamp

با کلیک روی دکمه دریافت اطلاعات، پارامترهای لازم برای فیلتر داده، به API ارسال شده و اطلاعات دریافت شده از آن نمایش داده می‌شوند:

نمایش پروژه →

دریافت داده

شی ارسال کننده:

شی دانشکده کامپیوتر ▼

انتخاب زمان شروع:

1397/7/1

انتخاب زمان پایان:

1397/10/30

دریافت اطلاعات

شناسه فرستنده	زمان دریافت داده	داده دریافت شده
5baceaf050b029e330ba0c21	۱۳۹۷/۷/۵ - ۱۸:۰۶:۳۱	مشاهده
5baceaf50b029e330ba0c20	۱۳۹۷/۷/۵ - ۱۸:۰۶:۳۰	مشاهده
5baceae50b029e330ba0c1f	۱۳۹۷/۷/۵ - ۱۸:۰۶:۲۹	مشاهده
5baceaed50b029e330ba0c1e	۱۳۹۷/۷/۵ - ۱۸:۰۶:۲۸	مشاهده

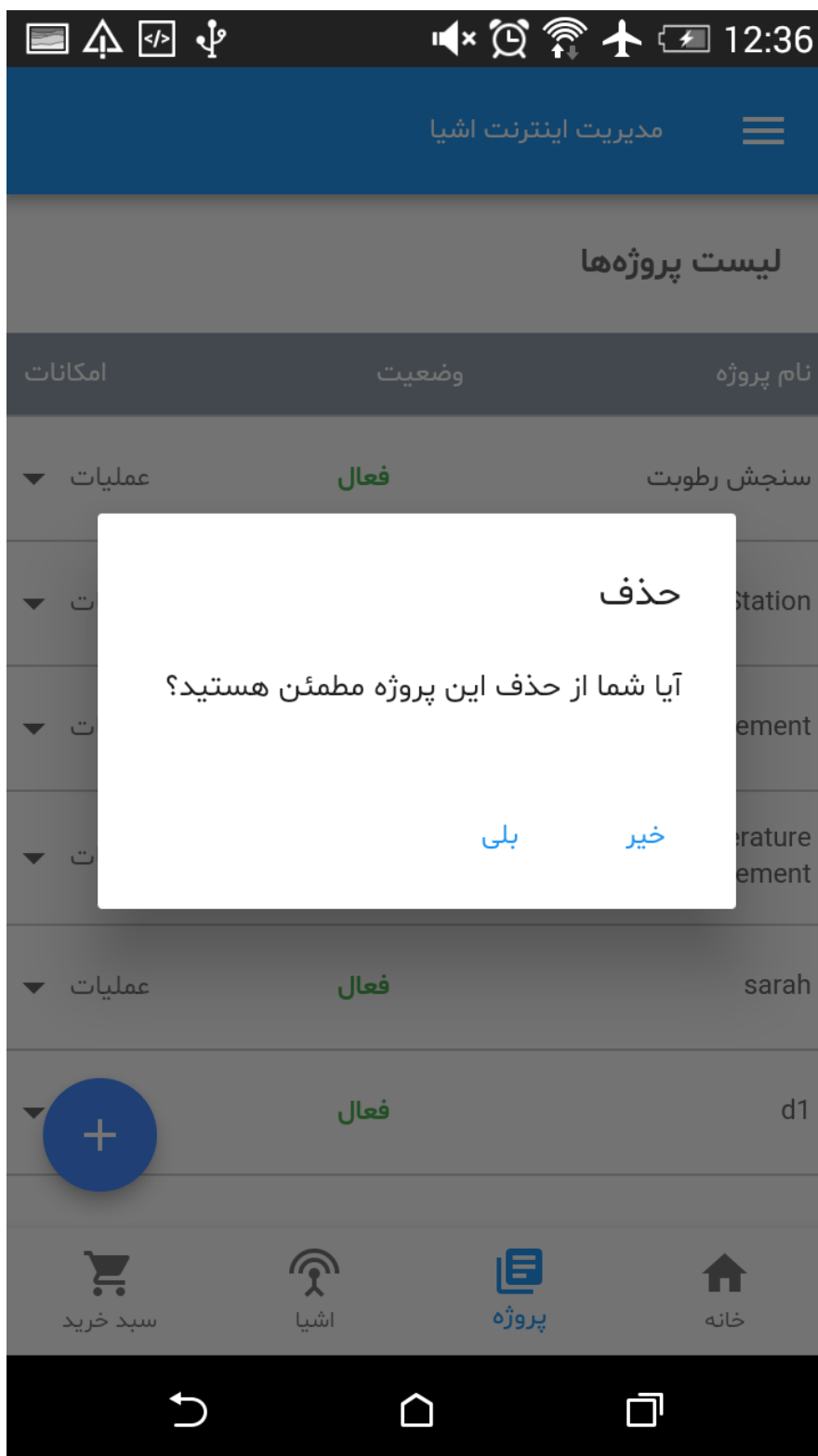
شکل ۱۲.۴ اطلاعات دریافتی در نمایش پروژه

۵.۵.۵.۴ مدیریت پروژه

از طریق این صفحه، کاربر می‌تواند نام، توضیحات و وضعیت فعالیت پروژه را تغییر دهد:

شکل ۱۳.۴ صفحه مدیریت پروژه

۶.۵.۵.۴ حذف پروژه



شکل ۱۴.۴ پرسش حذف پروژه

هنگامی که تاییدیه حذف موفق از جانب سرور بیاید، یک نواراسنک^۱ برای اطلاع کاربر در پایین صفحه نمایش داده می‌شود:



شکل ۱۵.۴ تاییدیه حذف پروژه

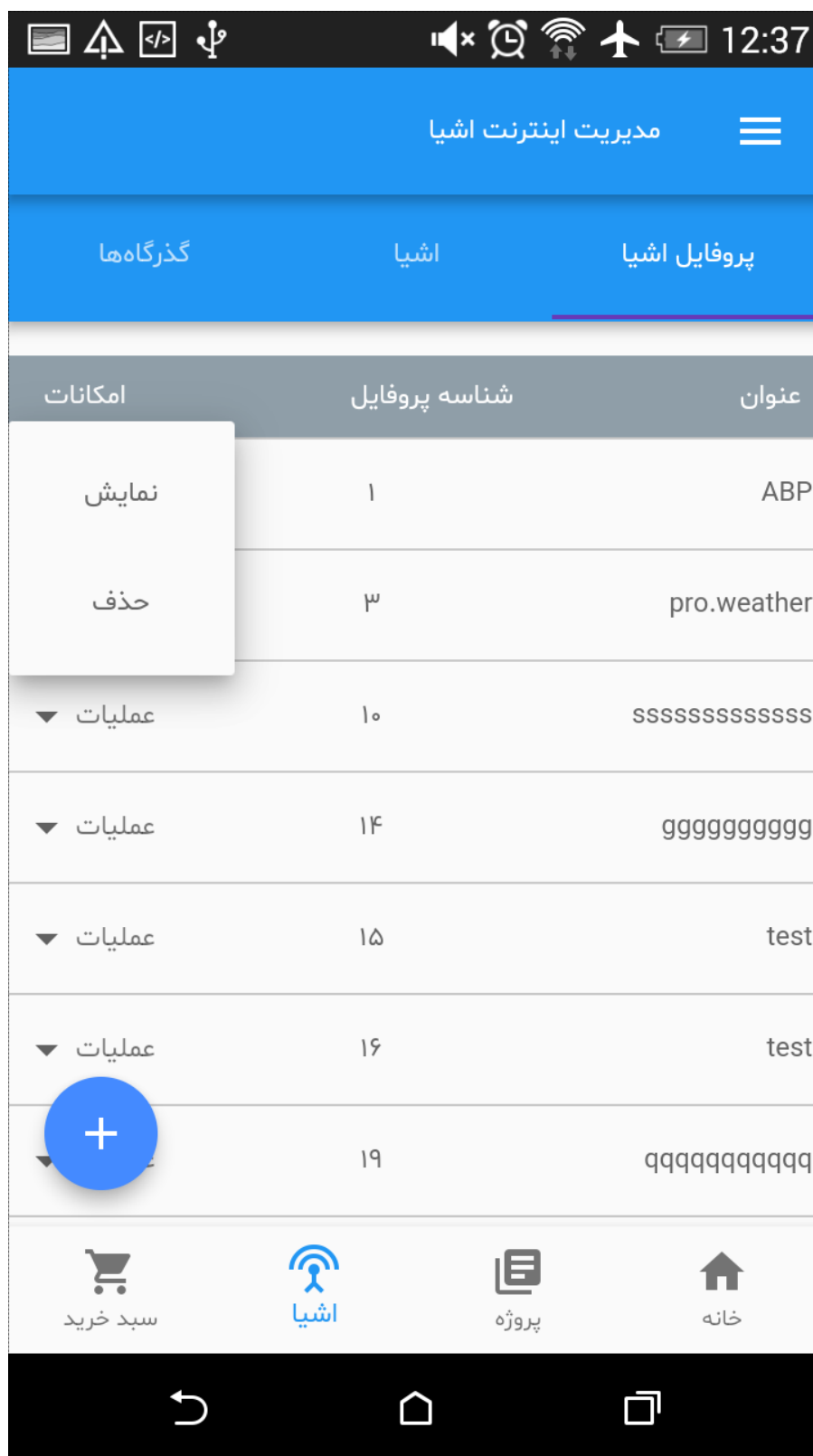
^۱ Snackbar

۶.۵.۴ پروفایل اشیا


پروفایل اشیا اطلاعات خام مربوط به اشیا را در بر دارد و کاربر می‌تواند از هر کدام از آنها با توجه به نیاز خود، برای اتصال به پروژه‌اش استفاده کند:

عنوان	شناسه پروفایل	امکانات
ABP	۱	عملیات ▼
pro.weather	۳	عملیات ▼
ssssssssssss	۱۰	عملیات ▼
ggggggggggg	۱۴	عملیات ▼
test	۱۵	عملیات ▼
test	۱۶	عملیات ▼
qqqqqqqqqqq	۱۹	عملیات ▼

شکل ۱۶.۴ لیست پروفایل اشیا



شکل ۱۷.۴ منوی عملیاتی پروفایل اشیا

 Saving screenshot...

مشاهده اشیا →

اطلاعات کلی

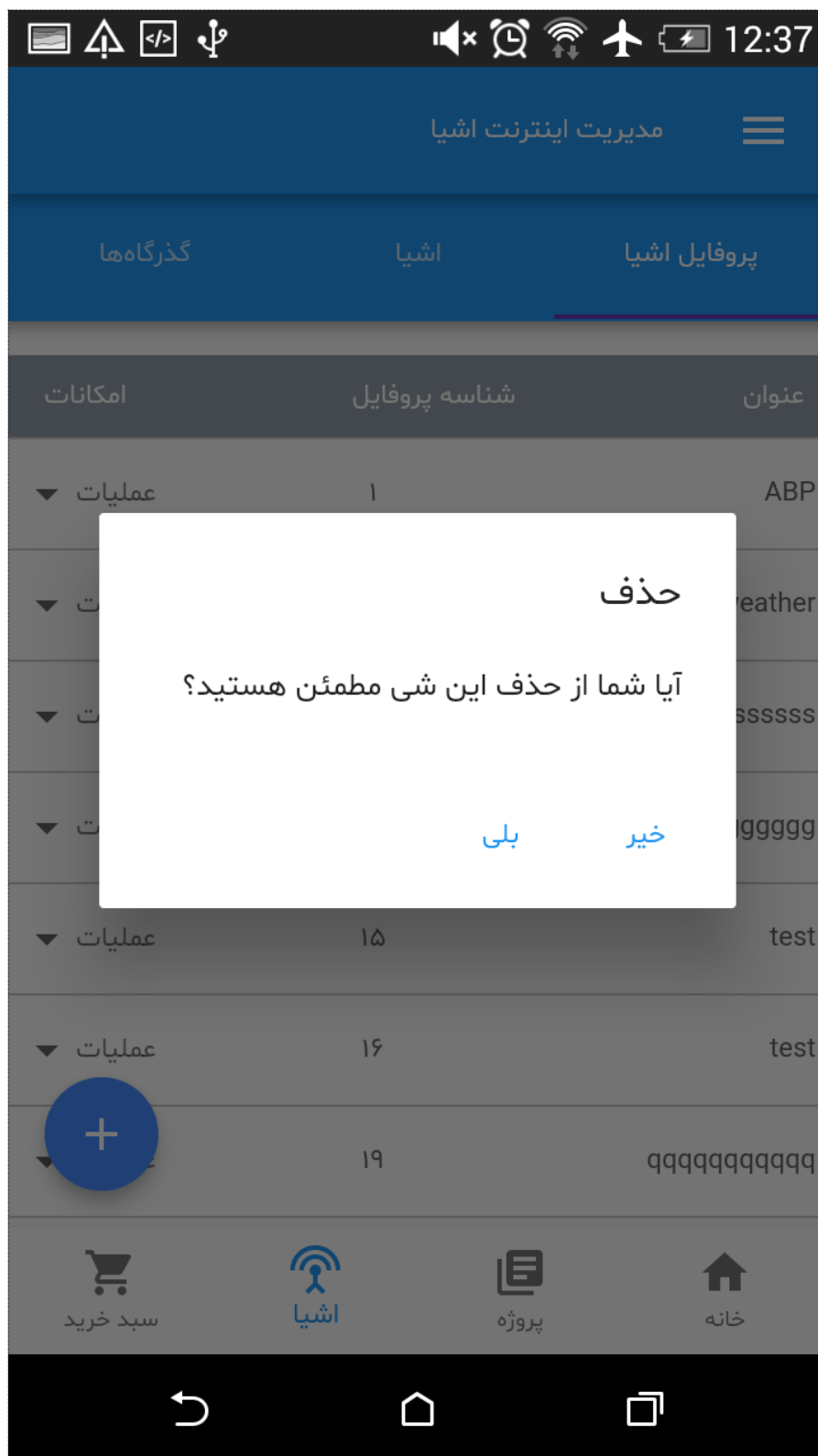
Name	ABP
LoRaWAN MAC Version	1.0.0
LoRaWAN Regional Parameters revision	A
Max EIRP	0

فعالسازی

Supports Join (OTAA)	false
RX1 Delay	0
RX1 Data Rate Offset	0

← 🏠 📄

شکل ۱۸.۴ مشاهده پروفایل اشیا



شکل ۱۹.۴ پرسش حذف پروفایل اشیا

هنگامی که تاییدیه حذف موفق از جانب سرور بیاید، یک نواراسنک^۱ برای اطلاع کاربر در پایین صفحه نمایش داده می‌شود:



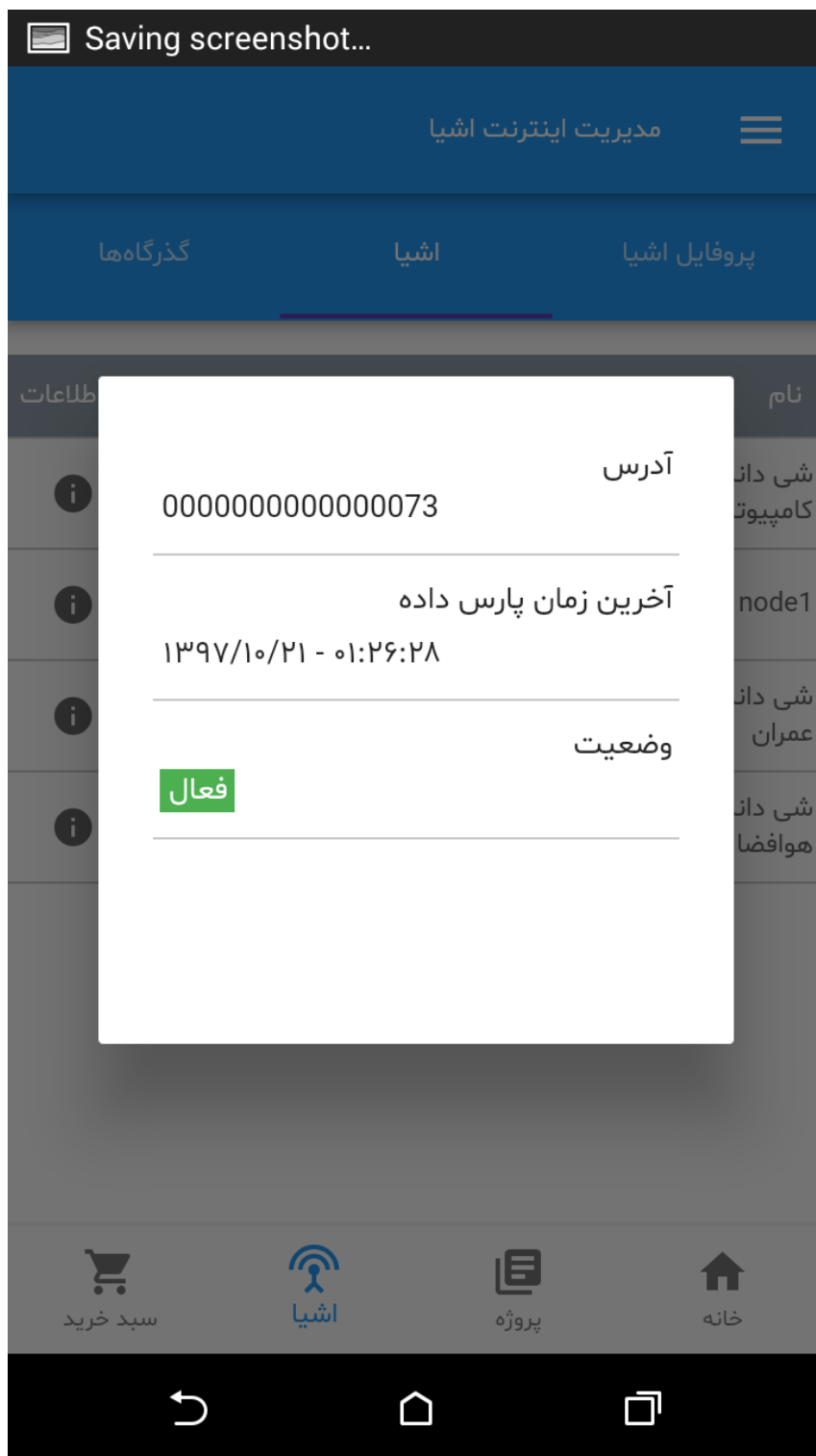
شکل ۲۰.۴ تاییدیه حذف پروفایل اشیا

^۱ Snackbar

۷.۵.۴ اشیا (متصل به پروژه)

مدیریت اینترنت اشیا			
پرو فایل اشیا			
اشیا			
گذرگاه‌ها			
نام	پروژه	نوع	اطلاعات
شی دانشکده کامپیوتر	سنجش رطوبت	ABP	i
node1	Weather Station	ABP	i
شی دانشکده عمران	سنجش رطوبت	ABP	i
شی دانشکده هوافضا	سنجش رطوبت	ABP	i
خانه	پروژه	اشیا	سبد خرید

شکل ۲۱.۴ لیست اشیا متصل به پروژه‌ها

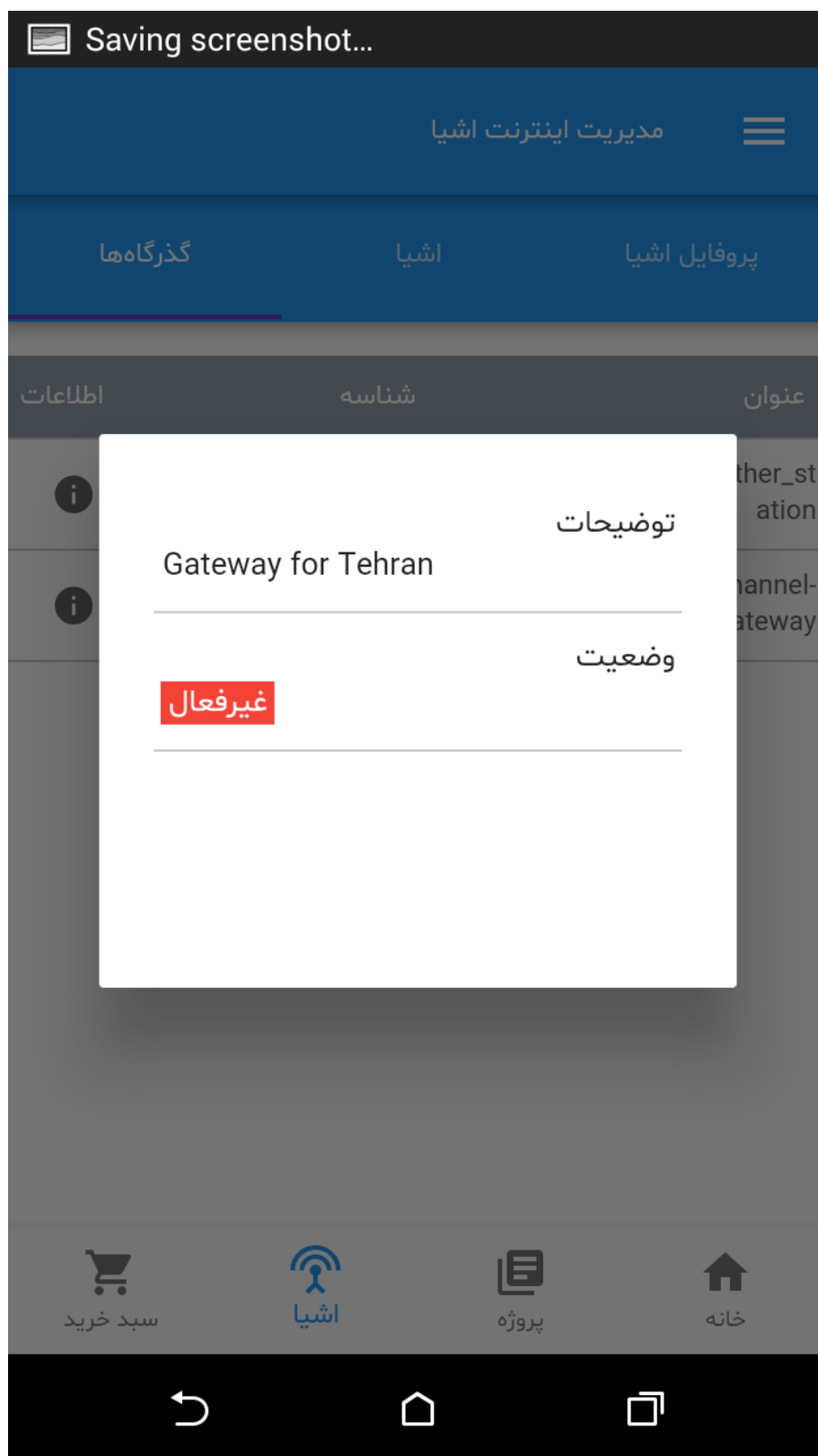


شکل ۲۲.۴ جزئیات اطلاعات اشیا

۸.۵.۴ گذرگاه‌ها



شکل ۲۳.۴ لیست گذرگاه‌ها



شکل ۲۴.۴ جزئیات اطلاعات گذرگاه‌ها

۹.۵.۴ بسته‌ها

بسته‌های خرید، مشخص می‌کنند که هر کاربر چه مدت، چه تعداد پروژه و شی می‌تواند بسازد یا استفاده کند.



شکل ۲۵.۴ بسته خریداری شده توسط کاربر



شکل ۲۶.۴ صفحه خرید بسته

۶.۴ کنترلر^۱

کنترلرها، مدل‌ها و viewها را به یکدیگر متصل می‌کنند. همانطور که در ابتدای فصل ذکر شد، بخش view درخواست‌های کاربر را به کنترلر می‌فرستد و کنترلر پس از دریافت اطلاعات از مدل و پردازش آنها، نتیجه را دوباره به view باز می‌گرداند.

از آنجاییکه در این پروژه، ارتباط با بک‌اند^۲ از طریق تماس با رابط برنامه‌نویسی کاربردی^۳، برقرار می‌شود از بیان جزئیات کنترلرها خودداری کرده و به بیان جزئیات ارتباط می‌پردازیم.

در تمامی این تماس‌ها لازم است یک نشانه^۴ همراه با درخواست^۵ ارسال شود تا پس از شناسایی کاربر با توجه به نشانه او، داده‌ها متناسب با دسترسی‌های وی در پاسخ^۶ ارسال شوند. در ابتدای امر لازم است به نشانی ورود^۷ درخواستی بفرستیم که حاوی ایمیل و گذرواژه‌ی کاربر است و در صورت تطابق اطلاعات ارسالی با اطلاعات یکی از کاربرهای پایگاه داده، نهایتاً نشانه در پاسخ ارسال شده و در اپلیکیشن ذخیره می‌شود تا در درخواست‌های بعدی این نشانه در قسمت سربرگ HTTP به عنوان Authorization ارسال شود.

^۱ Controller

^۲ Back end

^۳ API Callback

^۴ Token

^۵ request

^۶ response

^۷ login

۷.۴ توضیحات اجمالی از برخی موارد تخصصی

همانطور که پیشتر اشاره شد، اساس کارکرد اپلیکیشن با بک‌اند، برقراری تماس فراخوانی^۱ می‌باشد؛ لذا در تمام ویجت‌ها^۲ توابعی وجود دارند که این فراخوانی را انجام داده و پاسخ را به برنامه تحویل می‌دهند. برای مثال در ویجت `projects` تابع `getProjects` این کار را انجام می‌دهد. نوع متغیر خروجی این نوع از توابع هرچه باشد، از نوع آینده است مانند عدد^۳ آینده، رشته^۴ آینده و ...؛ یعنی این تابع زمانی پاسخ^۵ را آماده خواهد کرد که نتیجه در زمانی پس از زمان ارسال درخواست^۶، دریافت شده و در قالب ازپیش‌تعیین‌شده (برای مثال عدد، رشته و ...) قرار بگیرد.

برای ارسال درخواست لازم است ابتدا کتابخانه‌ی `http` را روی اپلیکیشن نصب و به لیست کتابخانه‌های ویجت وارد^۷ کنیم. با ارائه نشانی^۸ `API` به `httpClient`، یک `HttpRequest` می‌سازیم. دو مورد مهم باید در سربرگ‌های درخواست تنظیم شود؛ اولی نوع پاسخ درخواستی ما و دیگری نشانه کاربر. از آنجایی که در اپلیکیشن‌های استاندارد مرسوم است که هنگام بارگذاری داده تا زمانی که داده به طور کامل دریافت شود، یک تصویر متحرک^۹ نمایش داده شود. موارد بالا در قطعه کد زیر موجود هستند:

^۱ Callback

^۲ Widget

^۳ Integer

^۴ String

^۵ Response

^۶ Request

^۷ Import

^۸ URL

^۹ GIF

```

Future<void> getProjects() async {
    String getUrl =
    'http://platform.ceit.aut.ac.ir:۵۰۰۰۸/api/v\project'

    HttpClient httpClient = new HttpClient();

    HttpClientRequest request = await
    httpClient.getUrl(Uri.parse(getUrl));

    request.headers.set('content-type', 'application/json');
    request.headers.set('Authorization','Bearer ' +
    globals.userToken);

    HttpClientResponse response = await request.close();
    if (response.statusCode == ۲۰۰) {
        var json = await response.transform(utf8.decoder).join();
        var data = jsonDecode(json);
        if (data['code'] == ۲۰۰) {
            fetchedData = data['result']['projects'];
            setState(() {
                isLoading = false;
            });
        }
    }
}

```

تابع `fetchUser` نیز مانند توابعی است که در بالا به آن‌ها اشاره شده اما بهتر است بیشتر با نوع ذخیره‌سازی اطلاعات کاربر در اپلیکیشن آشنا شویم. این تابع پس از ارسال درخواست و دریافت پاسخ، اطلاعات کاربر را در `sharedpreferences` ذخیره می‌کند که عبارتست از فضایی کوچک برای ذخیره‌سازی اطلاعات کلید-مقدار^۱. این اطلاعات با باز و بسته کردن برنامه و حتی پاک

^۱ Key-Value

کردن حافظه موقت نهان^۱، از بین نمی‌روند. البته برای افزایش سرعت برنامه با هربار باز کردن برنامه، این اطلاعات به حافظه موقت برنامه منتقل می‌شوند تا دسترسی راحت‌تر شود.

مثال دیگری از توابع مهم، تابع `dateConverter` است که در چند ویجت استفاده شده و برای تبدیل تاریخ‌های میلادی^۲ به جلالی استفاده می‌شود. برای انجام این کار از کتابخانه‌ی `shamsi_date` استفاده کردم.

برای ایجاد چهار بخش در نوار پایین برنامه (خانه، پروژه، اشیا و سبد خرید)، از ویجت آماده‌ی `BottomNavigationBar` و برای منوی دراور از ویجت آماده‌ی `Drawer` استفاده کرده‌ام.

۸.۴ جمع‌بندی

معماری MVC در این پروژه به صورت دولایه اجرا شده است؛ معماری لایه اول که در آن مدل و کنترلر در لایه اول پیاده‌سازی شده‌اند و `view` آن همان اپلیکیشن موبایل است که با تماس با `API` آن در ارتباط است. لایه دوم اپلیکیشن موبایل است که مدل‌های آن از مدل‌های لایه بالاتر تبعیت کرده و بر آن اساس ساخته شده‌اند و کنترلر و `view` آن‌ها نیز در همان کلاس‌های اشاره‌شده موجودند.

^۱ cache

^۲ Gregorian

۵

فصل پنجم

جمع‌بندی و نتیجه‌گیری

جمع‌بندی و نتیجه‌گیری

در دنیای کنونی و شرایط حاضر که تمامی ابزارها و موجودیت‌های اطراف ما از طریق شبکه‌های کامپیوتری و به ویژه اینترنت به هم متصل هستند و به نحوی با یکدیگر تعامل دارند، وجود نرم‌افزارهایی یکپارچه برای کنترل وضعیت و مشاهده فعالیت‌ها و داده‌های تولیدی آنها امری ضروری است.

پروژه حاضر تلاشی برای ساخت چنین نرم‌افزاری در بستر موبایل است تا بتواند هر مکان و هر زمان پاسخگوی نیاز کاربران باشد و بهره‌گیری از اشیا هوشمند را در جامعه‌ی الکترونیک راحت‌تر کند. خروجی این اپلیکیشن مدیریتی در هردو سیستم عامل اندروید و iOS قابل نصب و استفاده می‌باشد.

منابع و مراجع

- [۱] ITU, "Internet of Things Global Standards Initiative." [Online]. Available: <http://www.itu.int/en/ITU-T/gsi/iot/Pages/default.aspx> . [Accessed: ۱۱-April-۲۰۱۹].
- [۲] B. Edson, "Get started with the Internet of Things in your organization Introducing the Microsoft Azure Internet of Things Suite." Microsoft.
- [۳] J. Stansberry, "The IoT communication protocols," *LinkedIn*, ۱۶-Oct-۲۰۱۵. [Online]. Available: <https://www.linkedin.com/pulse/iot-communication-protocols-james-stansberry> . [Accessed: ۱۰-April-۲۰۱۹].
- [۴] همایش «شهر هوشمند زیرساخت و فرصت‌های سرمایه‌گذاری»، تاریخ ۱۵ و ۱۶ شهریور ۱۳۹۴، برج میلاد، تهران
- [۵] DHL Trend Research and Cisco Consulting Services, *Internet of Things in Logistics*, ۲۰۱۵, Page ۴
- [۶] "LoraWAN", [Online]. Available: <https://linkap.net/blog/%D8%B4%D8%A8%DA%A9%D9%87-%D9%84%D9%88%D8%B1%D8%A7-lorawan/> . [Accessed: ۱۱-April-۲۰۱۹]
- [۷] Model View Controller. [online] Available at: <https://www.martinfowler.com/eaDev/uiArchs.html>
- [۸] Dart. [online] Available at: <https://www.dartlang.org/> [Accessed ۶ April ۲۰۱۹].
- [۹] Flutter. [online] Available at: <https://flutter.dev/> [Accessed ۶ April ۲۰۱۹].

