



گزارش پایانی پروژه برنامه نویسی دستگاه های موبایل

علی رضا صالحی دهنوی
علی رضایی
محمد راد

نام پروژه: Handy Home

۱۵ دی ۱۴۰۰

فهرست

- ۱- نام..... 3
- ۲- معرفی خلاصه برنامه..... 4
- ۳- انگیزه..... 5
- ۴- پیش زمینه و معرفی تیم..... 6
- ۵- هدف..... 7
- ۶- راهنمای استفاده..... 8
- ۷- تقسیم کار..... 13
- ۸- برنامه ریزی..... 14
- ۹- یادگیری..... 15

۱- نام

Handy Home را برگزیدیم. زیرا هدف پروژه آسان کردن مدیریت خانه (Home) با فراهم کردن یک GUI¹ کاربر پسند و کمینی² بود. به طوری که کاربری آن مانند ویژگی دم دستی و سریع (Handy) باشد. نیز قافیه میان دو واژه و تقارن نوشتاری نسبی که در طراحی GUI مناسب است، باعث این نامگذاری بود.

1رابط کاربری گرافیکی

2Minimal

۲- معرفی خلاصه برنامه

آیا شما نیز به ساخت اشیا هوشمند با پلتفرم آردوینو علاقه دارید؟ یاسیستم خانه هوشمند توسعه می‌دهید و دغدغه GUI آسان و کاربرپسند دارید؟ پس این برنامه برای شماست.

۳- انگیزه

بسیاری از افراد تمایل دارند وسایل را خودشان بسازند³. امروزه به ویژه با پدیدار شدن بسترهای ساده کننده ای مانند Arduino کار توسعه در زمینه برنامه نویسی سیستم های نهفته بسیار ساده تر از گذشته شده است. اینکار نیاز به تخصص چندانی ندارد از این رو فرد علاقه مند به این زمینه می تواند با کمی تلاش به سادگی شبکه ای از اشیا به هم متصل خود را بسازد. ولی این فرد علاقه مند احتمالاً به اندازه کافی در زمینه مهندسی کامپیوتر تخصص ندارد که یک برنامه رابط گرافیکی کارآمد هم بسازد.

یا شرکتی فعال در زمینه الکترونیک و سیستم های نهفته که به صورت تجاری در حال ساخت سخت افزار شبکه اشیا خانگی هوشمند می باشد ولی تخصص کمی در زمینه توسعه برنامه گرافیکی رابط کاربری دارد و نگران هزینه تمام شده محصول نهایی میباشد. برای دو مورد بالا برنامه موبایلی با پروتکل متن باز ارتباط با شبکه اشیا که در اندروید توسعه داده شده باشد، جذاب خواهد بود. زیرا تعامل کاربر با شبکه اشیا زیاد می باشد. پس رابط کاربری باید در دسترس باشد و موبایل بهترین گزینه است. بیشتر موبایل ها سیستم عامل اندروید دارند و سرانجام با متن باز کردن پروتکل، کاربران هدف را به محصول جذب و با پرداخت درون برنامه ای درآمد کسب میکنیم.

```
1 {
2   "devices": [
3     {
4       "type": "Temperature Sensor",
5       "location": "Dinning Room",
6       "custom_description": "",
7       "value": 35,
8       "unit": "Celcius",
9       "url": "ts1"
10    },
11    {
12      "type": "Lamp",
13      "location": "Living Room",
14      "custom_description": "",
15      "value": "off",
16      "unit": "enum(on,off)",
17      "url": "lp1"
18    },
19    {
20      "type": "Light Sensor",
21      "location": "Bed Room",
22      "custom_description": "",
23      "value": 35,
24      "unit": "Celcius",
25      "url": "/ls1"
26    }
27  ]
28 }
```

3 به اصطلاح با مخفف Do It Yourself (DIY) به آن اشاره می شود.

۴- پیش زمینه و معرفی تیم

علی رضا صالحی ورودی ۹۶ دانشکده است. در تابستان پس از ترم ۴ به دلیل علاقه به توسعه برنامه های موبایلی اندکی به صورت غیر ساختاری و نامنظم آشنایی پیدا کرد. ولی با گذشت دو سال و نیم، برتری کاربردی خاصی به شمار نمی رود. اما در توسعه برنامه های گرافیکی دسکتاپی در به زبان #C در فریم ورک WPF و به زبان Java و فریم ورک JavaFX آشنایی دارد.

علی رضایی ورودی ۹۶ دانشکده مهندسی پزشکی است. با برنامه نویسی Java و C آشنایی دارد و برنامه نویسی پیشرفته را گذارنده است.

محمد راد ورودی ۹۶ دانشکده است. وی نیز در زمینه های مرتبط پیشینه ای همانند با علی رضا صالحی دارد.

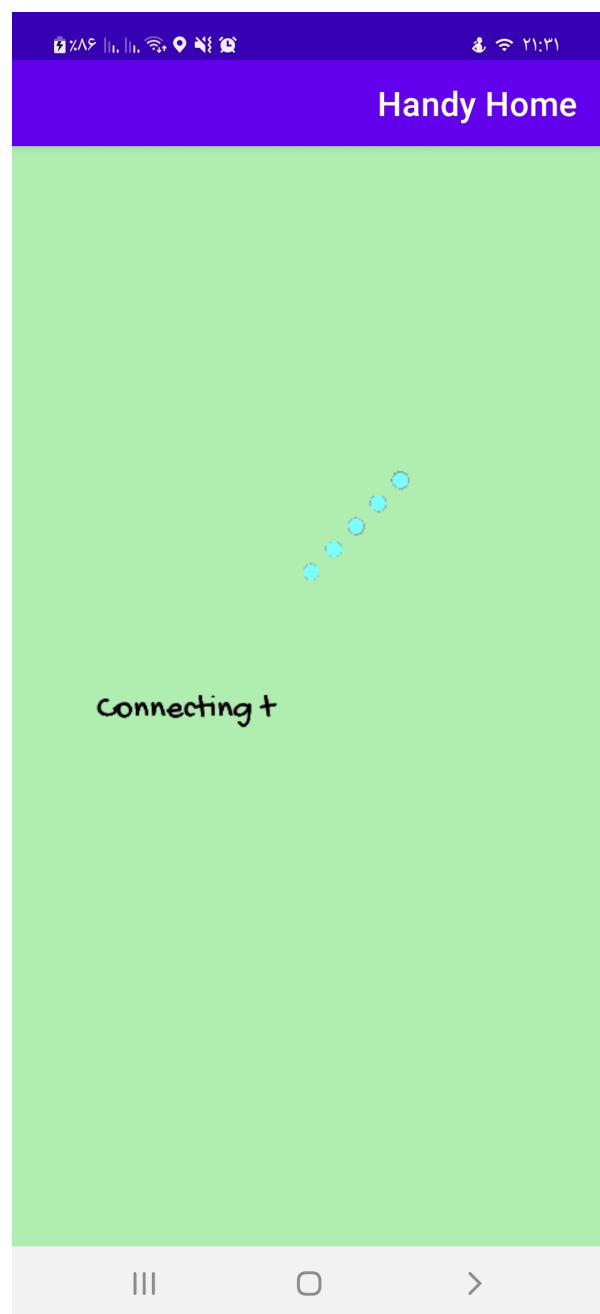
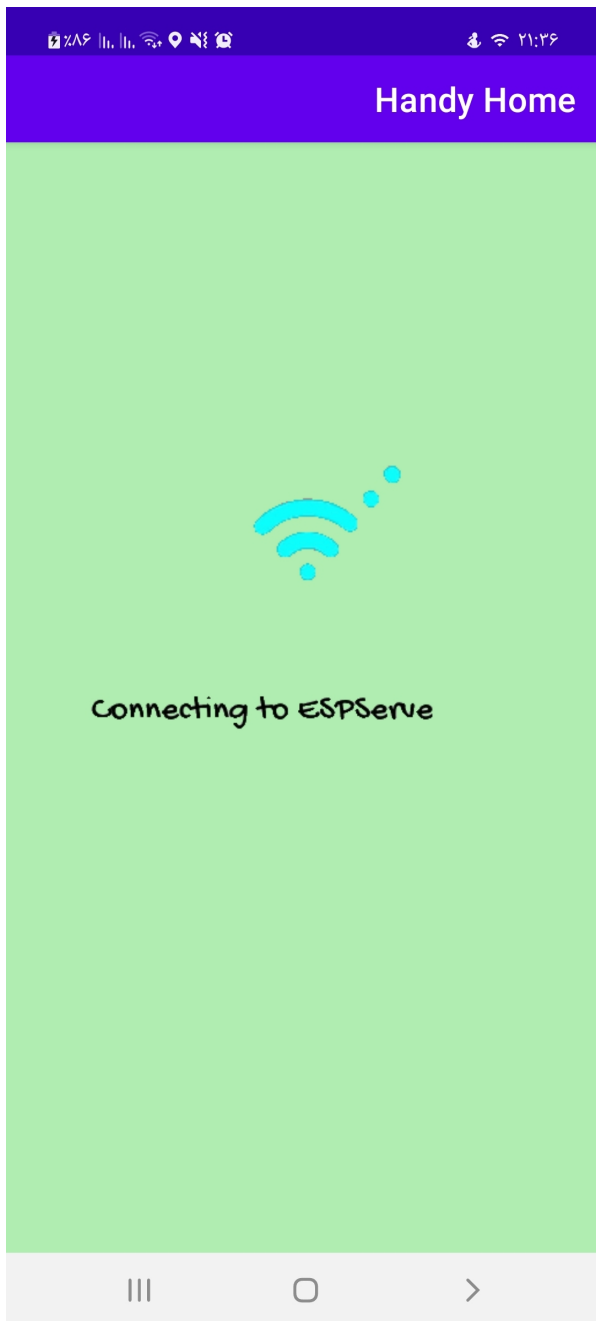
۵- هدف

همان طور که در بخش انگیزه توضیح داده شد. تلاش داریم یک رابط کاربری جذاب برای شبکه اینترنت اشیا توسعه دهیم و با جذب جامعه هدف از راه تبلیغات درون برنامه ای، کسب درآمد کنیم. از این رو باید اهداف ما در راستای:

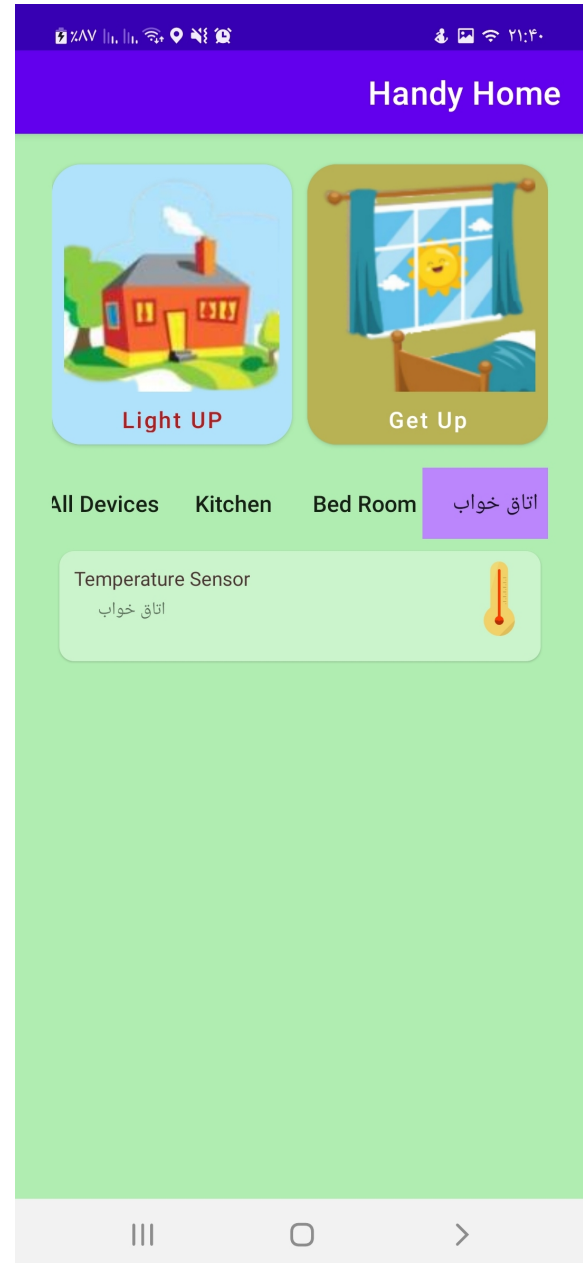
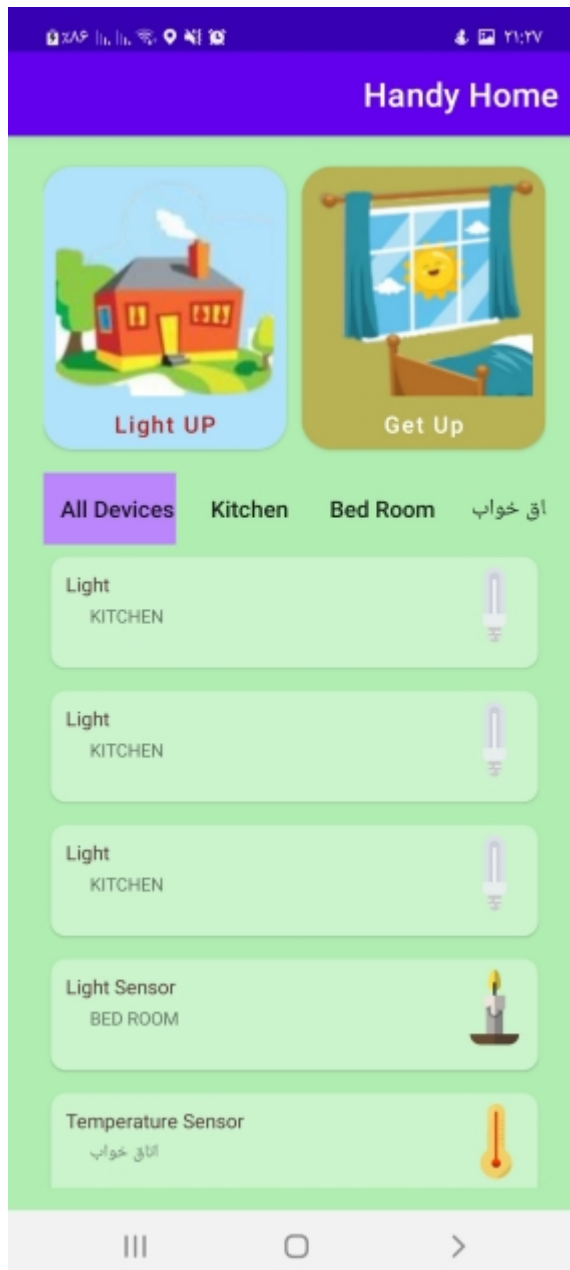
- طراحی یا استفاده از یک پروتکل ارتباطی که: ۱- کاربری آن آسان است ۲- متن باز باشد. ۳- بتواند واحدها در شبکه اینترنت اشیا خانگی را به طور کامل بازنمایی کند. مانند پشتیبانی از لامپ، سنسور دما، سنسور نور و... به همراه دستورهای مورد استفاده در هر یک مانند خاموش و روشن کردن لامپ، کاهش روشنایی لامپ، چشمک زدن و... یا خواندن سنسور دما
- رابط کاربری گرافیکی که: ۱- طراحی کمینه ای داشته باشد به طوری که استفاده از آن آسان باشد. ۲- جذاب باشد به طوری که جامعه هدف را جذب کند. ۳- پشتیبانی کامل از پروتکل طراحی یا استفاده شده مانند نمایش دستگاه های لامپ، سنسور دما و... و دستورهای مرتبط با آن ها.

۶- راهنمای استفاده

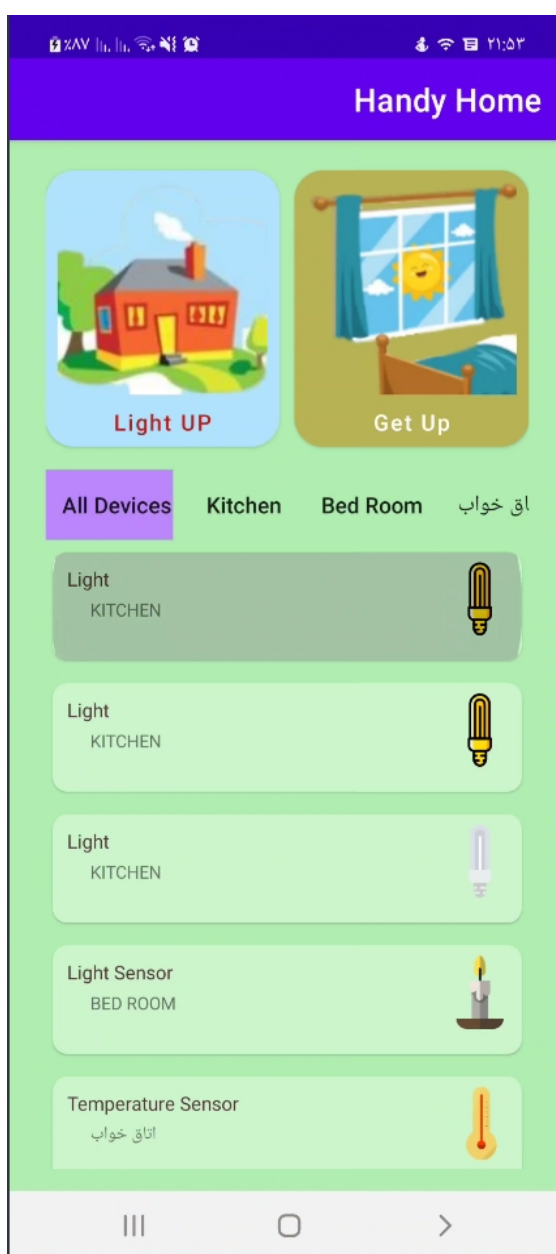
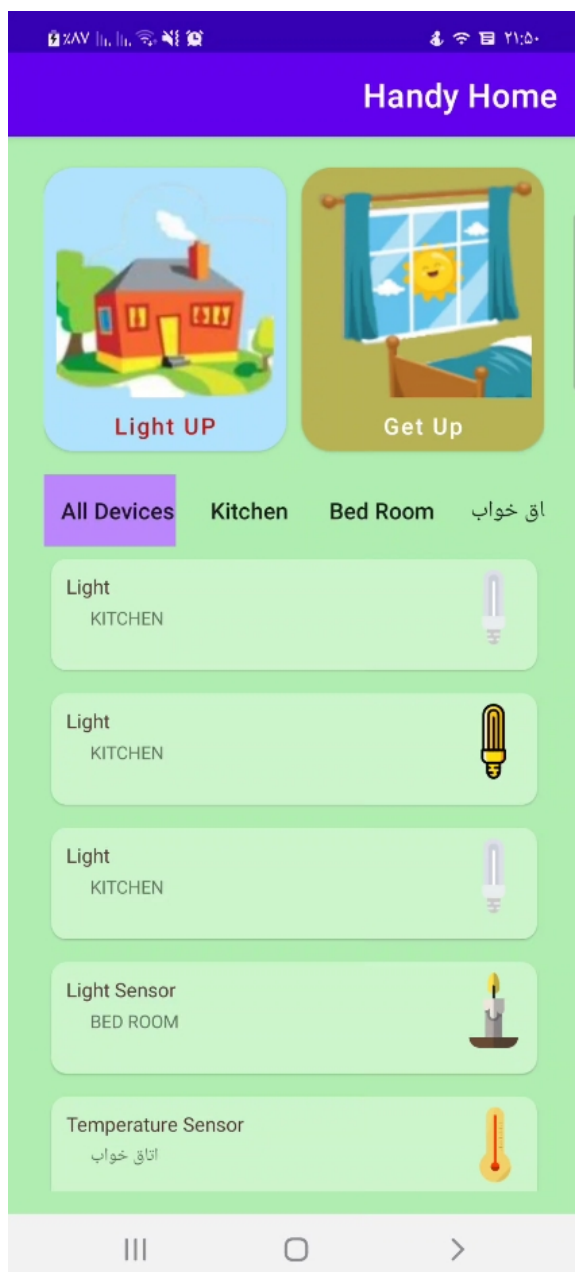
پس از باز شدن برنامه منتظر مانده تا برنامه اطلاعات دستگاه های درون شبکه را از سرور ESPServer دریافت نماید.



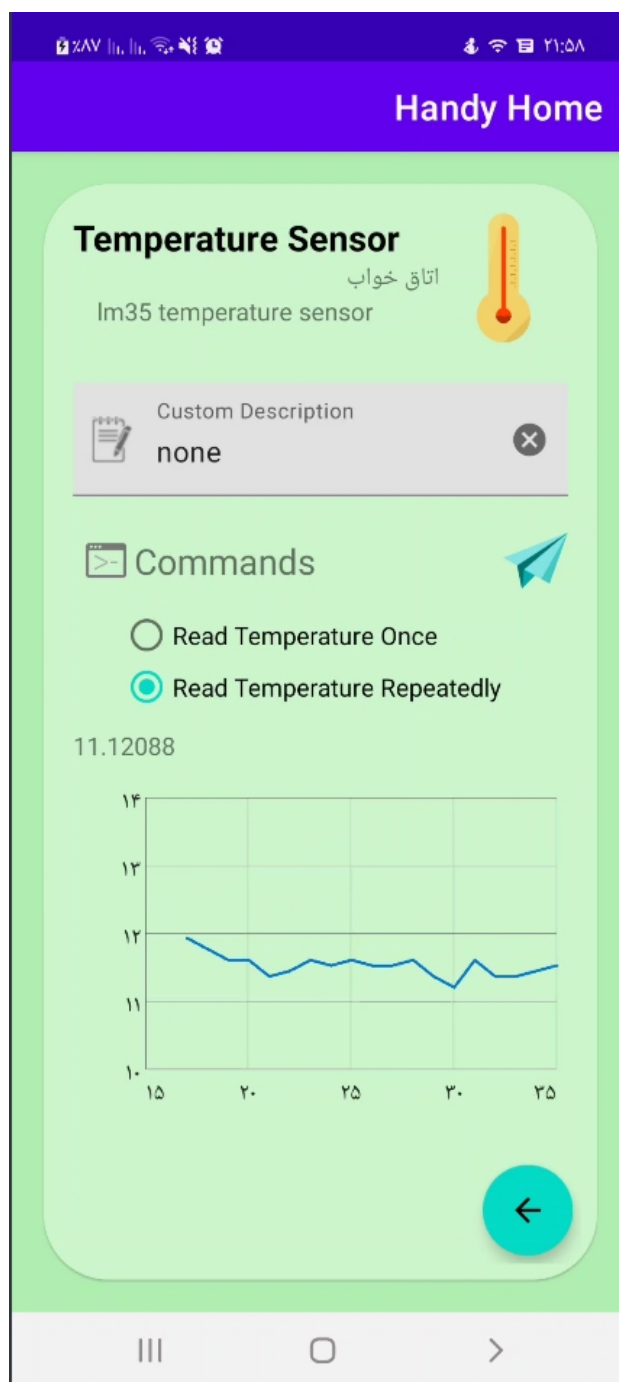
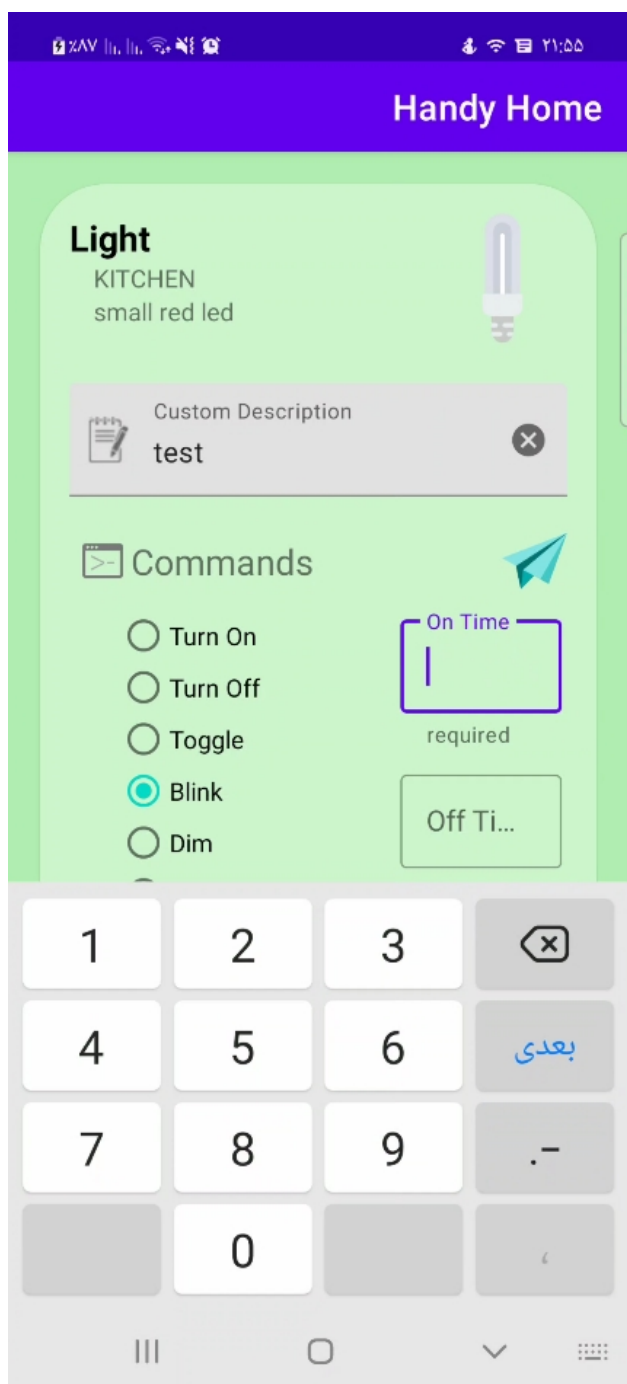
پس از گرفتن اطلاعات شبکه، رابط کاربری ساخته شده (Render) و صفحه ای همانند زیر نمایش داده می شود.



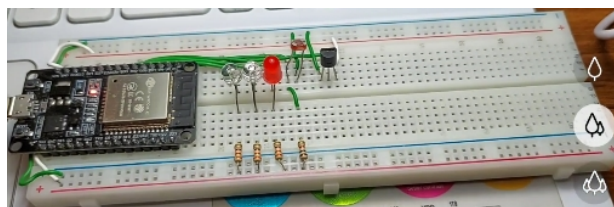
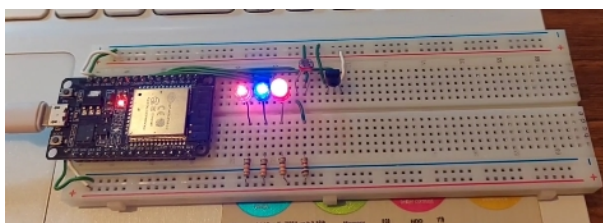
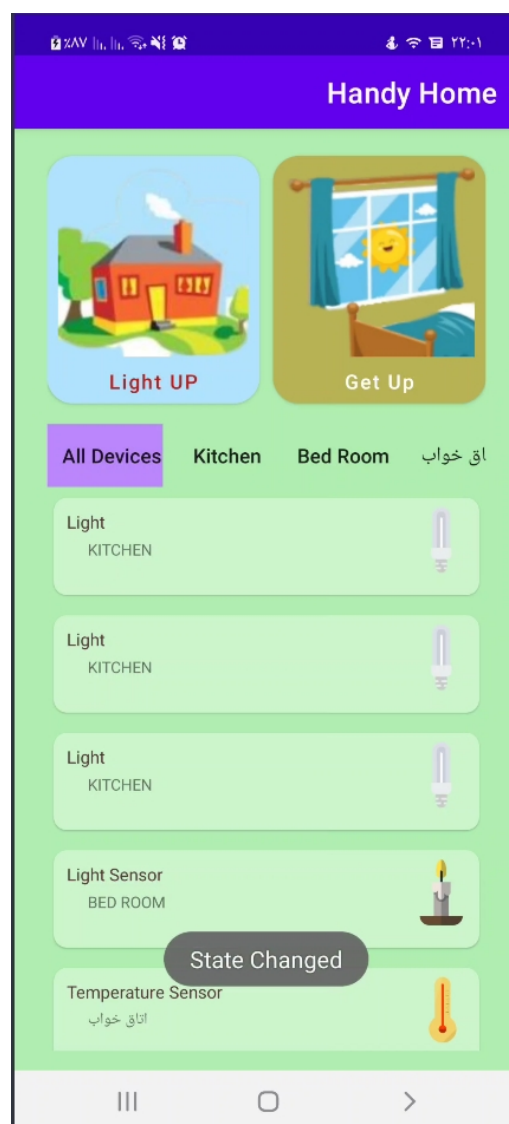
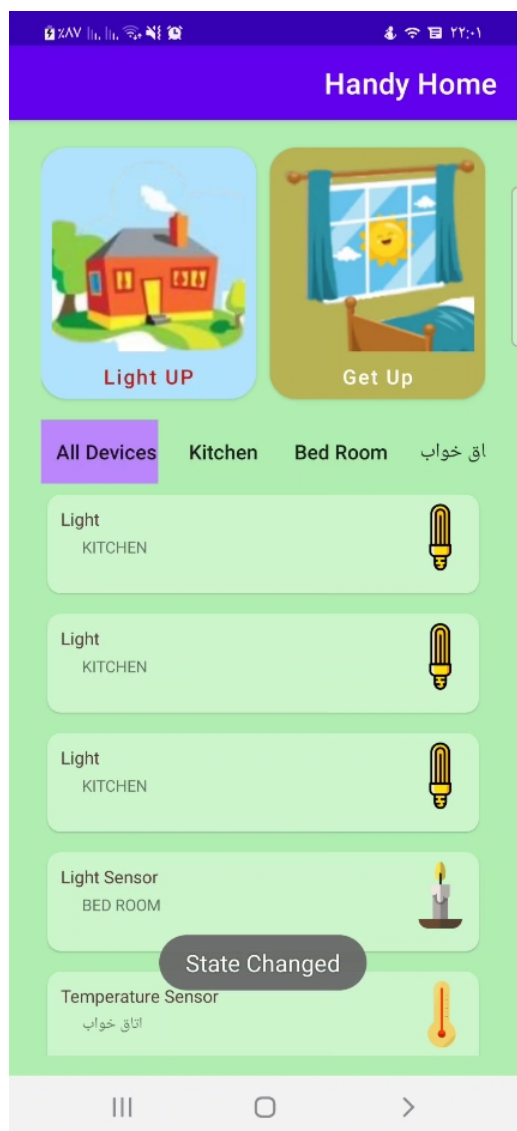
سپس می توان با زدن بر روی آیکون چراغ ها، وضعیت آن ها را تغییر داد و آیکون آن ها وضعیت کنونی را نمایش می دهد.



یا میتوان با کلیک بر روی نام هر دستگاه وارد صفحه مربوط به آن شد و همه دستورات را دید و یا توضیحاتی به دستگاه افزود: (Custom Description)



همچنین در صفحه خانه (اول) میتوان کنش های شرطی را نیز فعال کرد. برای نمونه دکمه Light Up در صورتی که روشنایی اندازه گیری شده با سنسور نور ۱ کمتر از ۵۰ درصد بود چراغ ها را روشن و در غیر این صورت خاموش میکند.



۷- تقسیم کار

طراحی بخش پروتکل ارتباطی با همکاری علی رضا صالحی و محمد راد انجام شد. همچنین وب سرور و تبادل JSON و ارتباط با شبکه (المان های مختلف ESPServer) به صورت همکاری این دو صورت گرفت.

بخش طراحی صفحه انتظار برای ارتباط با ESPServer و آیکون برنامه توسط علی رضایی انجام شد. و بخش خانه و نمایش ListView ای از المان های یک ArrayList - که از تجزیه کردن سند JSON فرستاده شده از ESPServer ساخته می شود یا در صورت ناموفق بودن ارتباط، از داده های درون پایگاه داده که به وسیله کتابخانه Room خوانده می شد، ساخته می گردد - با به کار بردن ArrayListAdapter که هر View این ListView یک CardView از Material است. (البته در آغاز یک CardView اولیه بود که خود تغییر و بهبود داده بودیم) و نمایش ImageButton های مربوط به کنش های شرطی و پیاده سازی OnClickListener های المان ها در فرگمنت خانه و طراحی مدل پایگاه داده با استفاده از کتابخانه room توسط علی رضا صالحی انجام شد.

پیاده سازی منطق انجام کنش شرطی پس از کلیک بر روی دکمه کنش شرطی LightUp و نمایش بخش مربوط به مشخصات دستگاه در فرگمنت مربوط به توضیحات توسط محمد راد انجام شد. همچنین مدیریت ارسال دستور ها در این فرگمنت بر توسط علی رضایی بود.

۸- برنامه ریزی

به عنوان فاز اول، برنامه داشتیم ابتدا یک رابط گرافیکی ایستا اولیه بر اساس موجودیت های فرضی و بلقوه که در دسترس داشتیم طراحی کنیم. که فقط یک پوسته باشد تا برآورد کنیم که با توجه به تلاش مورد نیاز برای این بخش نهایتا تا پایان ترم چه اندازه میتوانیم انتظار داشته باشیم که به پروژه بیاوریم. و این بخش در ارائه اول تقریبا آماده شد. هر چند آرایش و چینش جعبه متن ها و مقدار پدینگ و موارد از این دست به خوبی تنظیم نشده بود.

پس از آن تلاش کردیم بر اساس برآورد فاز یک پروتکل را طراحی کنیم و ESPServer (برنامه پیاده شده بر روی ESP) را پیاده سازی کنیم و کمی چینش رابط گرافیکی را بهبود بخشیم. که پروتکل تقریبا آماده شد و در رابط گرافیکی برای مثال پدینگ و چپ-چینی/راست-چینی مناسبی پیدا کرد و منظم تر شد ولی ESPServer به برنامه مدنظر نرسید و ارتباط میان رابط گرافیکی و ESPServer تا ارائه دوم هنوز برقرار نشده بود.

در ارائه نهایی برنامه ESPServer کامل شد و ارتباط میان رابط گرافیکی انجام شد و رابط گرافیکی بخش های پویا به آن افزوده شد. و مدل پایگاه داده افزوده شد. به طور کلی ارزیابی بخش های مختلف را می توان به صورت زیر داشت:

بخش های پیاده سازی سرور ESP که بر اساس پروتکل ارتباطی عملیات بر روی دستگاه ها را انجام میداد و وضعیت شبکه را با رابط کاربری تبادل میکرد. به خوبی انجام شد و نیاز های مورد انتظار که وجود یک سرور HTTP که دسترسی به منابع را با استفاده از پروتکل طراحی شده بر اساس روش REST فراهم میکرد، برآورده ساخت.

همچنین پروتکل ارتباطی با توجه به محدود بودن سخت افزار و بودجه به اندازه کافی طراحی شد. (هر سه دستگاه موجود یعنی LED و Photocell و LM35) را در بر گرفت.

همچنین در بخش رابط کاربری گرافیکی بازنمایی دستگاه ها و دستورهای موجود برای هر دستور پیاده شد. به طوری که هر یک از دستگاه ها با آیکون مناسب و با توجه به حالت آن (در صورتی که لامپ باشد) در صفحه اول بر اساس نام و مکان فهرست می شدند. و با کلید بر روی هر یک، فرگمنت دستورات و توضیح سلیقه ای باز می شد و می توانستیم دستورات مختلف را اجرا کنیم.

ولی بخش مربوط به **کنش های شرطی** برنامه داشتیم که **بتواند کلی تر** باشد و موارد قابل ویرایش سلیقه ای داشته باشد. مثلا اگر درصد نور/دما [و نور/دما] از مقدار x بیشتر/کمتر [و از مقدار y بیشتر/کمتر] شد آنگاه دستور چراغ های [یا مجموعه ای از چراغ ها] خاموش/روشن/میزان روشنایی به درصد/چشمک با پارامتر های X انجام شود.

و این کنش های شرطی به صورت سطرهای جدول در پایگاه داده نگه داری شوند و قابلیت افزایش و کاهش کنش های شرطی وجود داشته باشد. همچنین اضافه نمودن توضیح و تصویر برای آن ها و نمایش آن ها به صورت یک فهرست لغزان افقی که مانند فهرست لغزان مکان دستگاه ها باشد.

همچنین بخش دیگری که البته در تعریف اولیه پروژه در نظر گرفته نشده بود ولی برای محصول نهایی باید پیاده سازی شود. دریافت آدرس آبیی ESPServer توسط رابط گرافیکی به صورت پویا بود. که ما روش های مختلفی را برای اینکار در نظر داشتیم مانند اسکن کردن یک کد QR که آدرس ssid و password شبکه ESPServer که بر روی حالت Access Point گذاشته شده بود را میداد و پس از متصل شدن به آن با گرفتن ssid و password مودم وای فای خانه از حالت Access Point به حالت Station در آمده و به آن متصل میشد. که به دلیل کمبود زمانی و نیروی کار نافرجام ماند.

۹- یادگیری

تجربه گروهی ما در این پروژه یادگیری بهتر شیوه کار با نرم افزار مدیریت نسخه Git بود. همچنین با انجام یک نمونه پروژه در زمینه ارتباط اشیا در یک شبکه، تجربه بیشتری در زمینه مفاهیم شبکه در لایه های مختلف به دست آوردیم مانند دو حالت Access point و Station برای ند موجود در شبکه محلی وای-فای یا آشنایی محدود طراحی وب سرور که دارای آی پی آی RESTful می باشد.

همچنین با مفاهیم اصلی برنامه رابط گرافیکی در برنامه نویسی اندروید مانند Activity، Fragment، Toast، ListView و Adapter ها مانند ArrayListAdapter <> به خوبی آشنا شدیم.

مهم ترین چالش در این پروژه ناهمگونی آن بود به این صورت که بخشی از آن بر روی دستگاه اندرویدی با زبان Kotlin انجام می شد و بخشی دیگر بر روی دستگاه ESP32 یا ESP8266 و با زبان ++C/C++ که دو دستگاه بر روی بستر Wi-Fi با یکدیگر ارتباط برقرار میکردند. از این رو توسعه پرحجم تر و ایرادیابی به دلیل نبود یک بستر یکپارچه سخت تر می شد.

مهم ترین دست آورد نیز موفقیت نسبی در مدیریت این ناهمگونی بود. زیرا با آگاهی از احتمال رخداد باگ و ایراد یابی سخت به دلیل تبادل داده در میان چند دستگاه (شبکه) و ایرادیابی سخت برنامه های به زبان C، تلاش شد از روش های مطمئن و آزموده شده استفاده شود. مانند استفاده از سند JSON برای تبادل اطلاعات به جای پیاده سازی یک روش باینری خودساخته و ساختاریافتگی مناسب در کد C که مدیریت و ایراد یابی آن را ساده تر میکرد. به طوری که با وجود این نگرانی که ایراد در تبادل داده و انجام دستور توسط ESP وجود خواهد داشت و ایرادیابی آن زمان زیادی خواهد گرفت و بر روی کل پروژه و به ویژه بخش رابط گرافیکی اثر منفی میگذارد. اما اینگونه نشد.