



نام و نام خانوادگی اعضای گروه:

امیررضا جعفری (۴۰۲۱۰۵۸۳۵)

سیداحمد موسوی اول (۴۰۲۱۰۶۶۴۸)

عرفان تیموری (۴۰۲۱۰۵۸۱۳)

نام گروه: عاج

استاد راهنما: دکتر صفایی – مهندس جوادی

### ❖ گزارش پروژه

#### ○ عنوان پروژه

- مدیریت هوشمند داروها برای سالمندان

#### ○ چکیده

▪ عدم مصرف به موقع و منظم داروها یکی از چالش‌های اصلی در فرآیند درمان بسیاری از بیماران است. این پروژه با هدف حل این مشکل، به طراحی و ساخت یک «جعبه هوشمند دارو» می‌پردازد. این سیستم شامل یک جعبه فیزیکی با چهار محفظه مجزا است که هر کدام به سنسور وزن مجهز شده‌اند و از طریق میکروکنترلر ESP32 کنترل می‌شوند. یک اپلیکیشن اندروید نیز برای مدیریت جعبه طراحی شده است که به کاربر اجازه می‌دهد نام دارو، دوره زمانی مصرف و صدای آلارم مخصوص هر دارو را تنظیم کند. در زمان مقرر، اپلیکیشن با پخش آلارم و ارسال نوتیفیکیشن به کاربر یادآوری کرده و همزمان دستوری برای روشن شدن LED مربوط به داروی مشخص به جعبه ارسال می‌کند. پس از برداشتن دارو توسط کاربر، سنسور وزن این تغییر را تشخیص داده و پیامی به اپلیکیشن ارسال می‌کند که باعث قطع شدن آلارم و ریست شدن تایمر برای دوره بعدی می‌شود. این سیستم یک راحل یکپارچه و هوشمند برای بهبود پایداری بیماران به برنامه درمانی خود ارائه می‌دهد.

#### ○ حوزه کاربردی پروژه

- این پروژه در حوزه سلامت دیجیتال و اینترنت اشیا (IoT) کاربرد دارد. سیستم طراحی شده، برای نظارت بر مصرف داروهای افراد سالمند استفاده می‌شود. با استفاده از این دستگاه می‌توان از فراموشی در مصرف دارو جلوگیری کرد و از راه دور نیز وضعیت مصرف را کنترل نمود. کاربرد آن در خانه، مراکز نگهداری سالمندان یا حتی در محیط‌های درمانی امکان‌پذیر است.

## ○ اهداف پروژه

- اهداف اصلی این پروژه به شرح زیر تعریف شده بودند:
- طراحی و ساخت یک جعبه فیزیکی با چهار محفظه مجزا برای نگهداری داروها.
- یادآوری زمان مصرف دارو به کاربر از طریق آلارم صوتی قابل تنظیم و نوتیفیکیشن روی گوشی هوشمند.
- تشخیص خودکار برداشتن دارو توسط کاربر با استفاده از سنسورهای وزن (لودسل) برای هر محفظه.
- ارائه یک اپلیکیشن اندروید کاربرپسند برای مدیریت کامل جعبه، شامل تنظیم نام دارو، دوره زمانی مصرف (بر حسب ساعت، دقیقه یا ثانیه) و انتخاب صدای آلارم.
- برقراری ارتباط بی‌سیم و پایدار بین جعبه و اپلیکیشن از طریق شبکه وای‌فای و پروتکل MQTT.
- امکان مصرف دارو زودتر از موعد و ریست شدن خودکار تایمر.

## ○ معماری سیستم

- این سیستم از دو بخش اصلی سخت‌افزاری (جعبه فیزیکی) و نرم‌افزاری (اپلیکیشن اندروید) تشکیل شده است که از طریق یک سرور MQTT با یکدیگر ارتباط برقرار می‌کنند.

- اجزای سخت‌افزاری
  - میکروکنترلر ESP32: به عنوان مغز متفکر سیستم، وظیفه کنترل سنسورها، LED ها و برقراری ارتباط با شبکه را بر عهده دارد.
  - ماژول سنسور وزن HX711 (لودسل) - ۴ عدد: برای اندازه‌گیری دقیق وزن دارو در هر محفظه و تشخیص تغییرات آن.
  - LED - ۴ عدد: برای راهنمایی بصری کاربر و مشخص کردن محفظه‌ای که باید دارو از آن برداشته شود.
  - منبع تغذیه: برای تأمین انرژی مورد نیاز کل سیستم.
- اجزای نرم‌افزاری
  - سمت میکروکنترلر: کدنویسی با فریم‌ورک آردوینو و استفاده از کتابخانه‌های WiFiManager.h برای مدیریت اتصال به شبکه، PubSubClient.h برای ارتباط MQTT و HX711.h برای کار با سنسورهای وزن.
  - سمت اپلیکیشن: توسعه اپلیکیشن نیتیو اندروید با زبان کاتلین (Kotlin) و استفاده از Jetpack Compose برای طراحی رابط کاربری مدرن و واکنش‌گرا.
  - سرور: استفاده از یک بروکر عمومی MQTT به عنوان واسطه برای تبادل پیام‌های سبک بین جعبه و اپلیکیشن.
- پروتکل ارتباطی
  - پروتکل MQTT به دلیل سبک بودن، مصرف انرژی پایین و مدل کارآمد Publish/Subscribe برای این پروژه انتخاب شد. تایپیک‌های اصلی تعریف شده عبارتند از:
  - esp32/commands: اپلیکیشن دستورات خود (مانند روشن/خاموش کردن LED یا ریست کردن بخش‌ها) را روی این تایپیک منتشر می‌کند و ESP32 به آن گوش می‌دهد.

○ `esp32/medication/taken`: هرگاه ESP32 تشخیص دهد که دارویی مصرف شده، پیامی حاوی شماره بخش مربوطه را روی این تاپیک منتشر می‌کند و اپلیکیشن به آن گوش می‌دهد.

## ○ جزئیات پیاده‌سازی

- بخش سخت‌افزار و جعبه فیزیکی  
جعبه فیزیکی به گونه‌ای طراحی شد که چهار محفظه مجزا داشته باشد و در زیر هر محفظه یک سنسور لودسل قرار گیرد. میکروکنترلر ESP32 در مرکز مدار قرار گرفته و اتصالات مربوط به سنسورهای HX711 و LEDها به پین‌های GPIO آن متصل شده‌اند.
- برنامه سمت میکروکنترلر  
اتصال به شبکه: با استفاده از کتابخانه WiFiManager، دستگاه در اولین راه‌اندازی یک شبکه وای‌فای ایجاد می‌کند که کاربر می‌تواند با اتصال به آن، اطلاعات شبکه وای‌فای خانگی خود را وارد کند. این اطلاعات در حافظه ESP32 ذخیره شده و در راه‌اندازی‌های بعدی به صورت خودکار استفاده می‌شود.

اتصال به MQTT: پس از اتصال به شبکه، دستگاه به بروکر MQTT متصل شده و تاپیک `esp32/commands` را برای دریافت دستورات باز می‌کند.

منطق خواندن وزن: در حلقه اصلی برنامه (loop)، وزن هر چهار سنسور به صورت مداوم خوانده می‌شود. برای جلوگیری از خطاهای ناشی از نویز، از میانگین‌گیری چندین نمونه استفاده می‌شود.

منطق تشخیص مصرف دارو: کد به دقت تغییرات وزن را زیر نظر دارد. با استفاده از یک متغیر **THRESHOLD**، تنها تغییرات وزنی قابل توجه به عنوان یک رویداد (برداشتن یا گذاشتن) در نظر گرفته می‌شود. منطق `if (med_w1 - sstable_w1 > THRESHOLD && ret1)` دقیقاً لحظه‌ای را تشخیص می‌دهد که پس از گذاشتن جعبه دارو، مقداری از آن کم شده است (مصرف شده).

ارسال پیام: پس از تشخیص مصرف، یک پیام با فرمت JSON (مثلاً `{"part": 1}`) ساخته و به تاپیک `esp32/medication/taken` ارسال می‌شود.

دریافت دستورات: تابع `mqttCallback` پیام‌های دریافتی از اپلیکیشن را پردازش کرده و توابع مربوطه مانند `turn_LED_on` یا `reset_part` را فراخوانی می‌کند.

- اپلیکیشن اندروید  
رابط کاربری (UI): با استفاده از Jetpack Compose، یک رابط کاربری ساده و مدرن طراحی شد. صفحه اصلی شامل چهار کارت است که هر کدام اطلاعات یک دارو (نام، زمان مصرف بعدی و وضعیت) را نمایش می‌دهند. دکمه تنظیمات امکان تغییر آدرس بروکر MQTT را فراهم می‌کند و دیالوگ تنظیم دوره زمانی به کاربر اجازه می‌دهد زمان را بر حسب ساعت، دقیقه یا ثانیه وارد کند.

سرویس پس‌زمینه (**MqttForegroundService**): این سرویس قلب تپنده اپلیکیشن است. با اجرای آن به عنوان یک سرویس پیش‌زمینه (**Foreground Service**)، اندروید اجازه نمی‌دهد که با بسته شدن اپلیکیشن، فرایندهای حیاتی آن (مانند اتصال به MQTT و چک کردن زمان) متوقف شوند. این امر پایداری عملکرد سیستم را تضمین می‌کند.

سیستم هشدار و آلام:

سرویس در یک حلقه بی‌نهایت، هر ثانیه زمان فعلی را با زمان آلام بعدی هر دارو مقایسه می‌کند. در زمان مقرر، نوتیفیکیشن با دکمه "**Dismiss**" نمایش داده شده و صدای آلام مخصوص آن دارو به صورت مداوم پخش می‌شود.

دستور `{command:"alarm_on", "part":X"}` به **ESP32** ارسال می‌شود تا **LED** مربوطه روشن شود.

با دریافت پیام "**taken**" از **ESP32**، تابع **handleMedicationTaken** فراخوانی می‌شود که آلام و نوتیفیکیشن را متوقف کرده، دستور خاموش شدن **LED** را به **ESP32** ارسال کرده و تایمر را برای دوره بعدی ریست می‌کند.

ذخیره‌سازی داده‌ها: تمام تنظیمات مربوط به هر دارو (نام، دوره زمانی، صدای آلام، وضعیت ردیابی و...) با استفاده از **SharedPreferences** در حافظه داخلی گوشی ذخیره می‌شود تا با بستن و باز کردن مجدد اپلیکیشن، اطلاعات از بین نرود.

## ○ نتایج و تست سیستم

- سیستم در شرایط مختلف مورد آزمایش قرار گرفت و نتایج زیر به دست آمد:
  - تست اتصال: دستگاه **ESP32** با موفقیت به شبکه‌های وای‌فای مختلف متصل شد و ارتباط پایداری با بروکر **MQTT** برقرار کرد. اپلیکیشن نیز در حالت‌های آنلاین و آفلاین، اتصال خود را به درستی مدیریت کرد.
  - تست دقت سنسور وزن: سنسورها توانستند تغییرات وزنی بالای ۱ گرم را به صورت پایدار و قابل اطمینان تشخیص دهند که برای تشخیص برداشتن یک قرص یا کپسول کاملاً مناسب است.
  - تست اپلیکیشن: آلام‌ها و نوتیفیکیشن‌ها دقیقاً در زمان تنظیم شده فعال شدند. دکمه "**Dismiss**" و همچنین منطق قطع خودکار آلام پس از دریافت پیام از **ESP32** به درستی کار کردند. قابلیت مصرف زودهنگام دارو نیز با موفقیت تست شد.
- نتایج تست را می‌توانید در [این لینک](#) ببینید
- چالش‌های پروژه:
  - اتصال در پس‌زمینه: یکی از بزرگترین چالش‌ها، اطمینان از فعال ماندن سرویس در نسخه‌های جدید اندروید بود که محدودیت‌های زیادی برای مصرف باتری دارند. این مشکل با استفاده صحیح از **Foreground Service** و درخواست از کاربر برای غیرفعال کردن بهینه‌سازی باتری (**Battery Optimization**) برای اپلیکیشن حل شد.
  - ناپایداری سنسور وزن: در ابتدا، سنسورهای وزن نویز زیادی داشتند. این مشکل با میانگین‌گیری از تعداد بیشتری نمونه در کد **ESP32** و همچنین طراحی مناسب پایه فیزیکی برای لودسل‌ها برطرف گردید.