# Deliverable 3

# Quantum Quants

# Housefy

# Team 04

## Team members

Member 1: Wenyuan Yu, Student ID: n01403697

Member 2: Kyrylo Lvov, Student ID: n01414058

Member 3: Artem Tsurkan, Student ID: n01414146

# Table of Contents

# 1. Brief Description of the Project

Housefy project consists of two parts - a mobile application and an IoT (hardware) device. Both are designed to work together and provide a Smart Home solution.

# 2. Member's Info and Participation Table

| Name | ID | Signature | Effort |
|---|---|---|---|
| Kyrylo Lvov | n01414058 | Kyrylo Lvov | 100% |
| Wenyuan Yu | n01403697 | Wenyuan Yu | 100% |
| Artem Tsurkan | n01414146 | Artem Tsurkan | 100% |
| | | | |

# 3. GitHub Repo Link

Android app: https://github.com/WenyuanYu3697/housefy
Backend: https://github.com/WenyuanYu3697/housefy-asp.net-core-web-app

## 4. Sprint Goals

In Sprint 3, we're focusing on covering all important areas of the software, making sure it works well and is easy to use.
First, we're aiming to create a user-friendly interface. This is all about how the app looks and feels to our users. Every button, swipe, and action needs to make sense and be easy to use.

Next, we're focusing on the 'behind the scenes' part of our app - the backend. This is really important because it handles all the heavy lifting for our app.

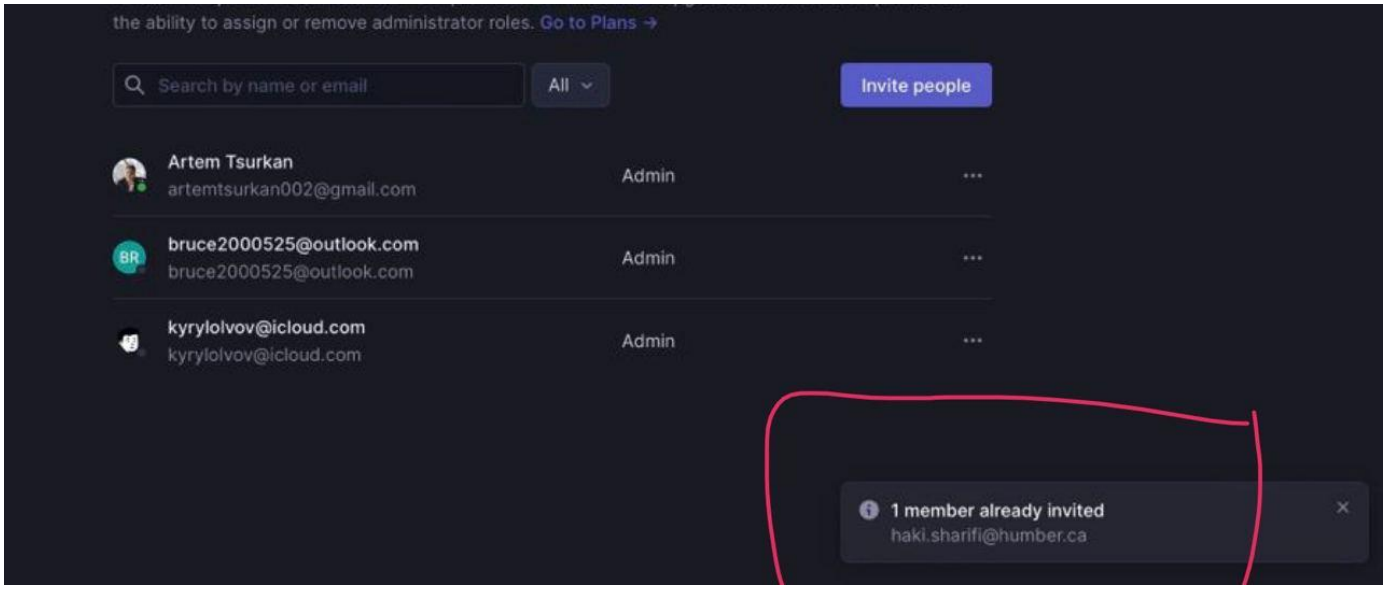
But it's not just about making the frontend (how the app looks) and backend (how the app works) better on their own. We also want them to work well together. When the frontend and backend work well together, users have a smooth and enjoyable experience with our app.

To sum up, our sprint goal is about improving three main things: how the app looks, how the app works, and how these two parts work together.

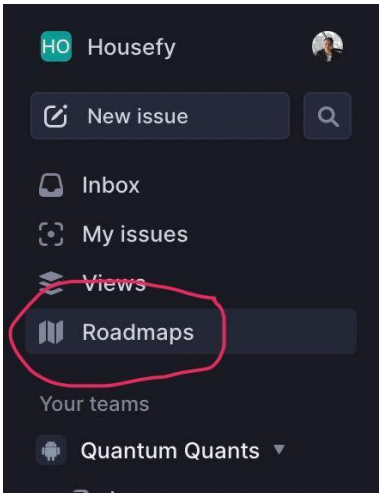
# 5. Sprint Dashboard w./ stories, tasks, owner, status, start/end date, size

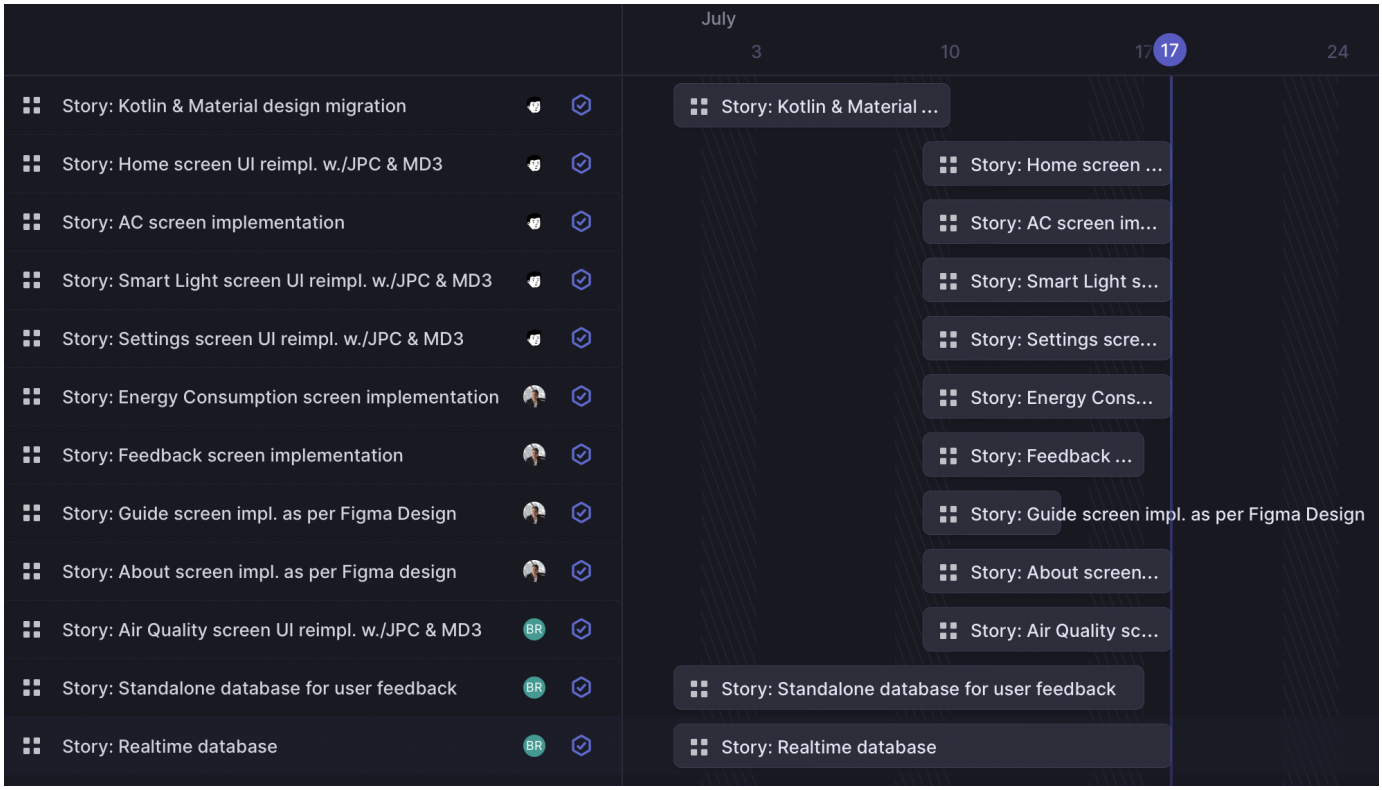
https://linear.app/housefy/roadmap/all

Screenshot of invitation:

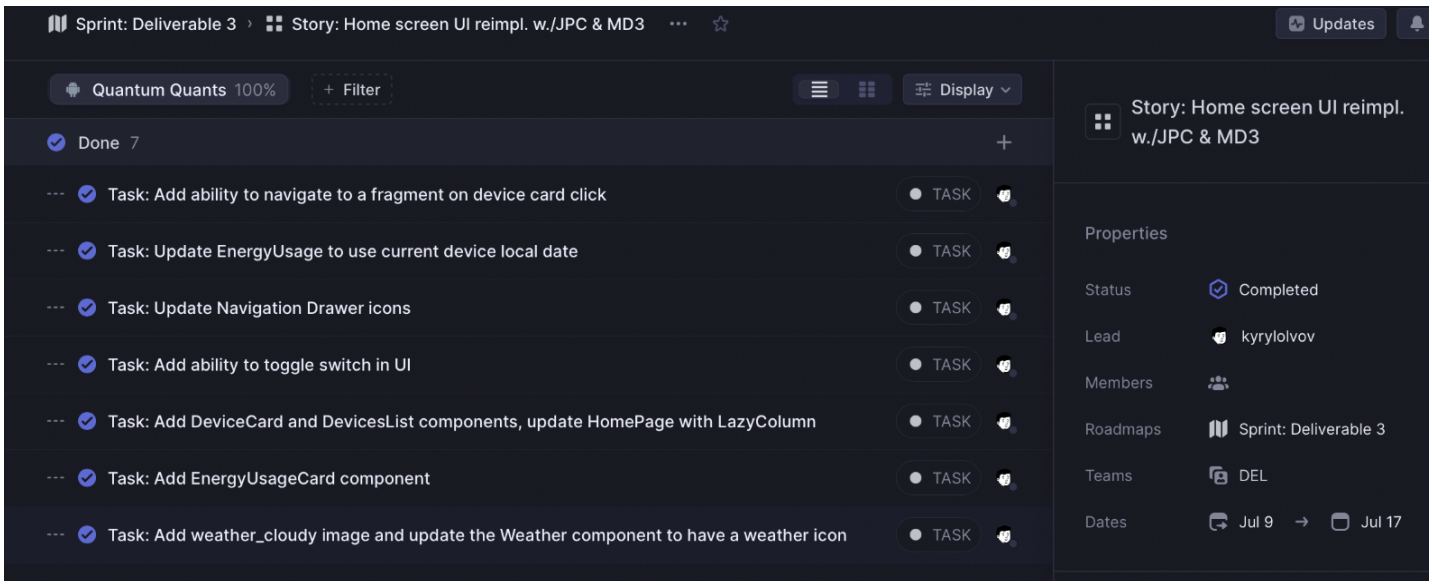



The stories are located in the “Roadmaps” section of the side menu on the left:

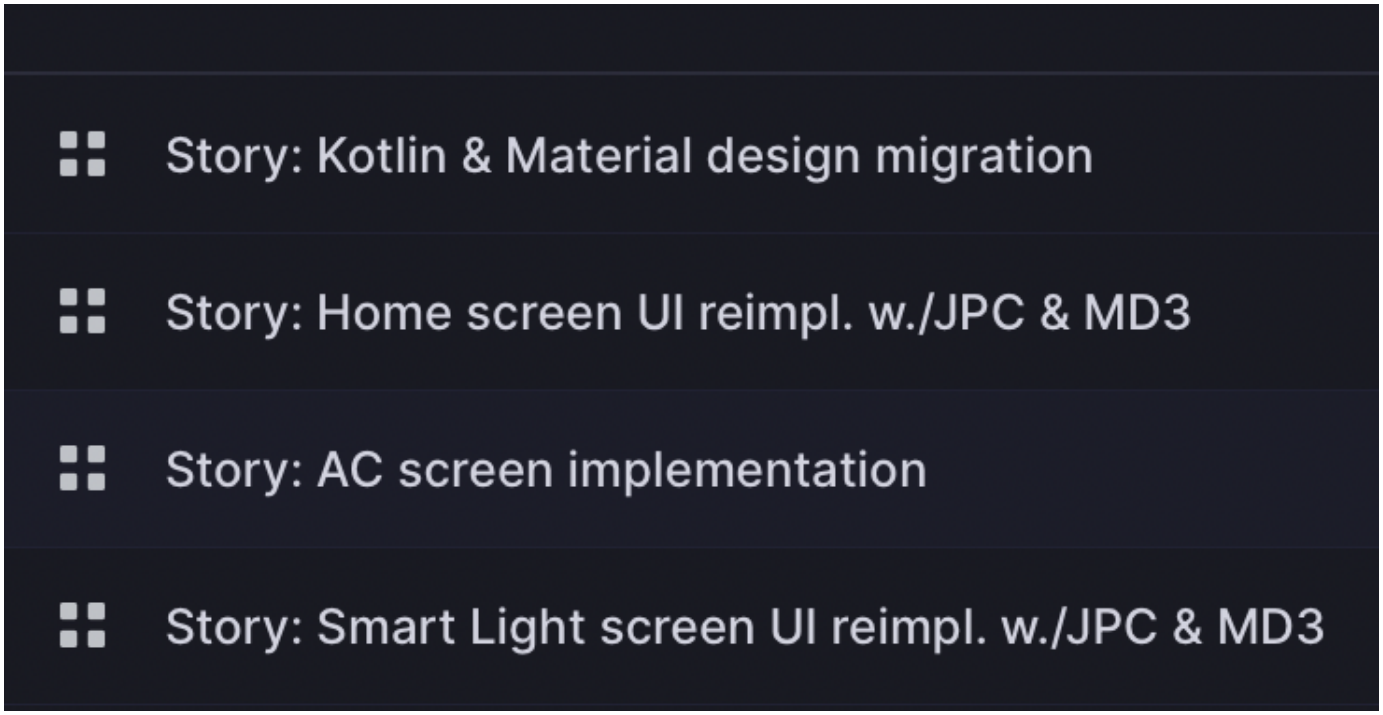





You can click on each story to view the tasks and details. For example:



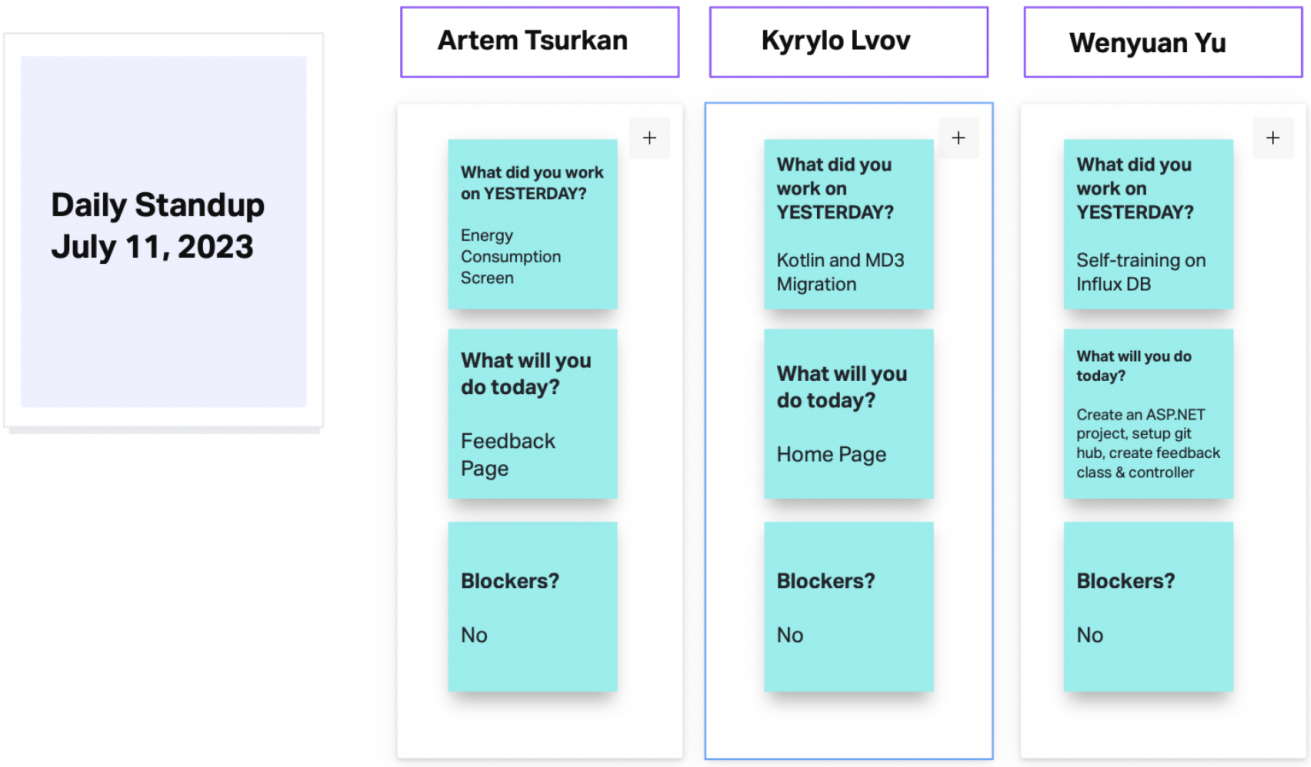
The first 4 stories already meet the requirement of 4 stories x 5 tasks. You can click on each of them to confirm. Many other stories have 5+ tasks in them (some don’t, but there aren’t many of them).

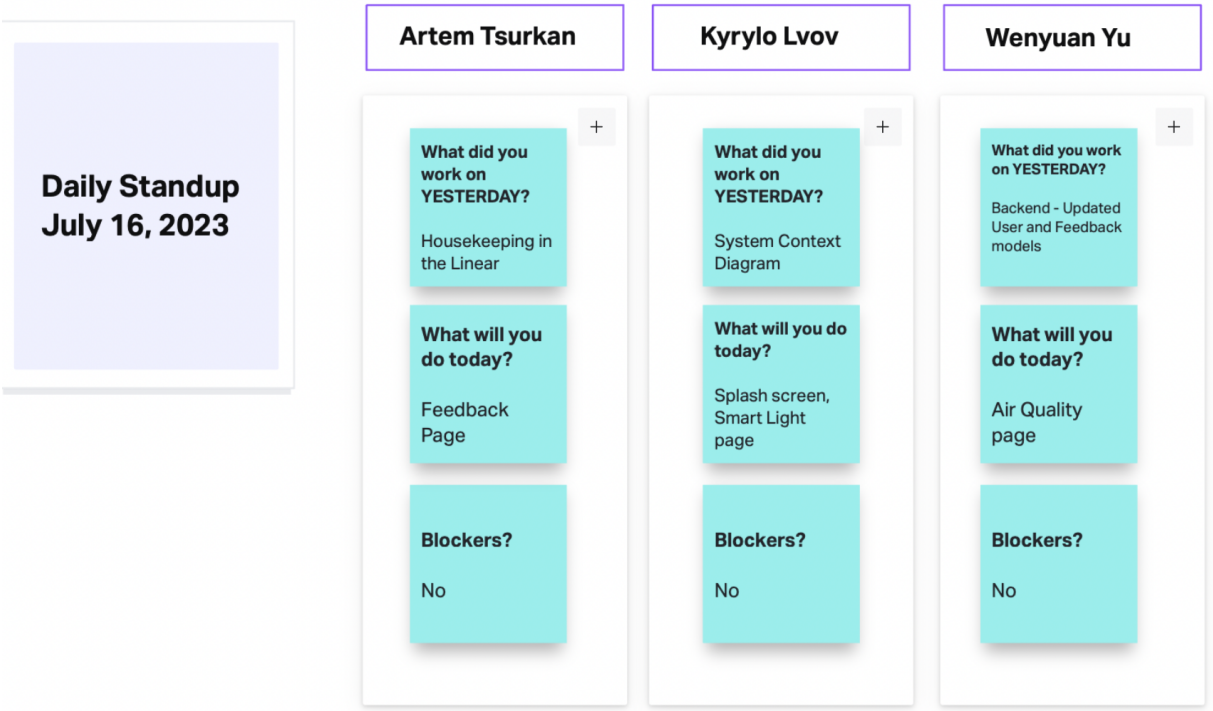


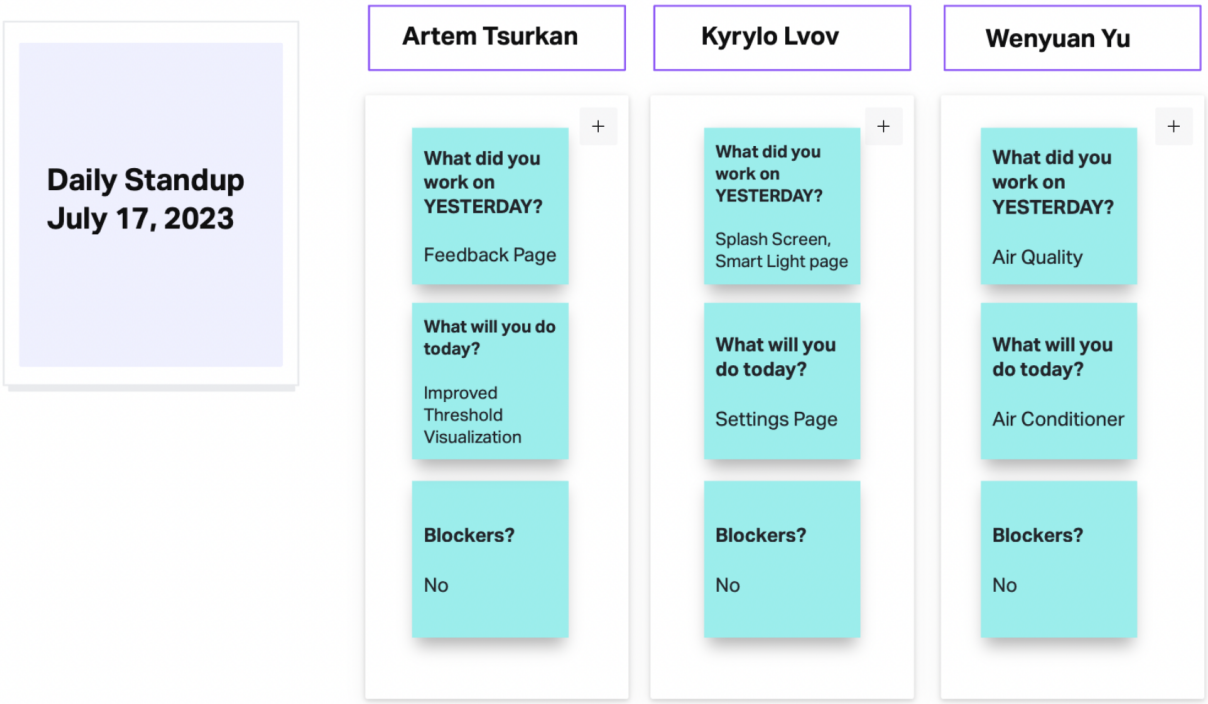
## 6. Gantt Chart with main milestones and work progress

The app that we are using for Sprint Dashboard, Linear, automatically creates a Gantt Chart (screenshots above already illustrate that).

## 7. Daily Standups Table





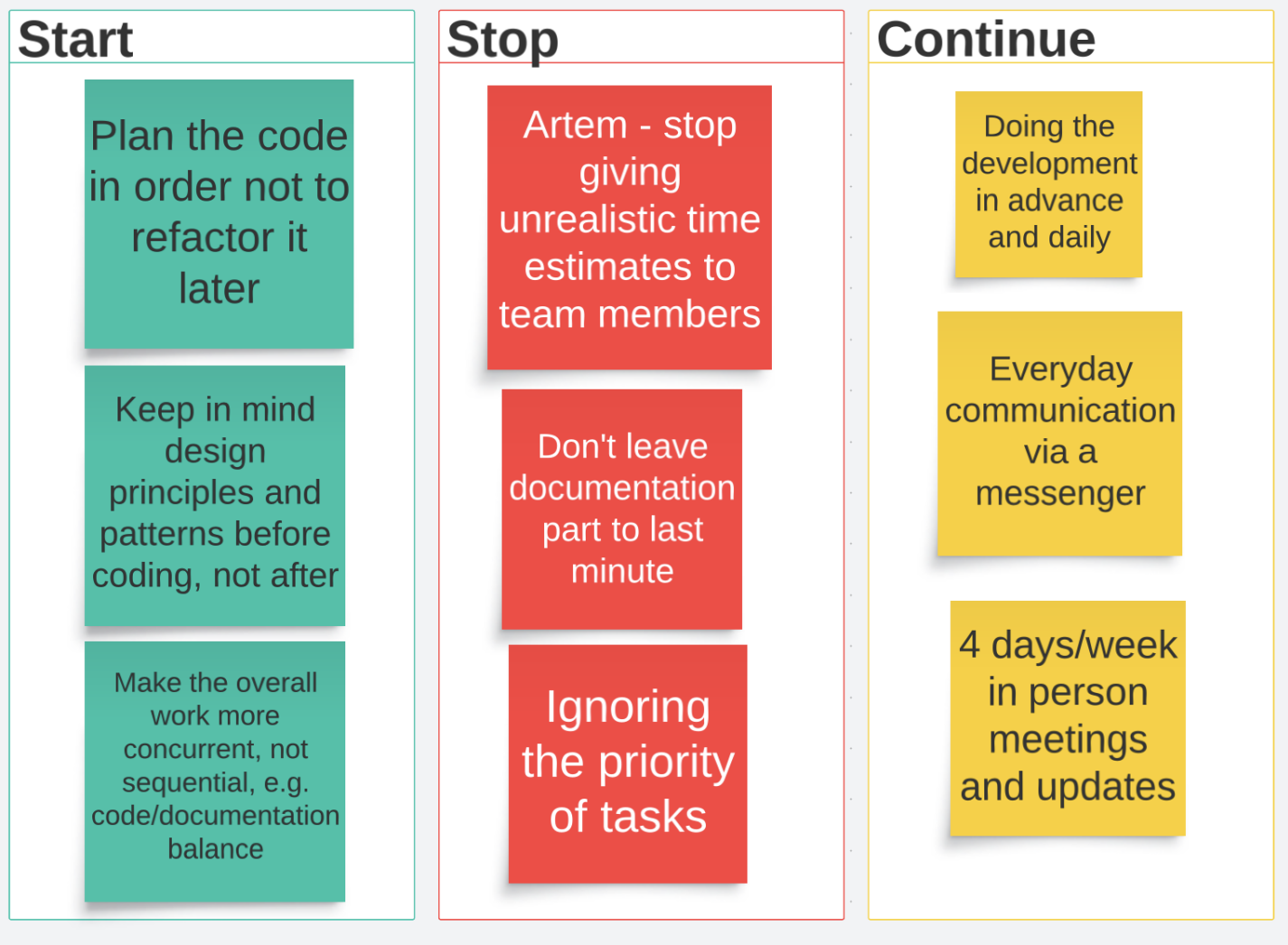




**Link:**

https://artemtsurkan32268.invisionapp.com/freehand/Daily-Standups-hTtty7cR8

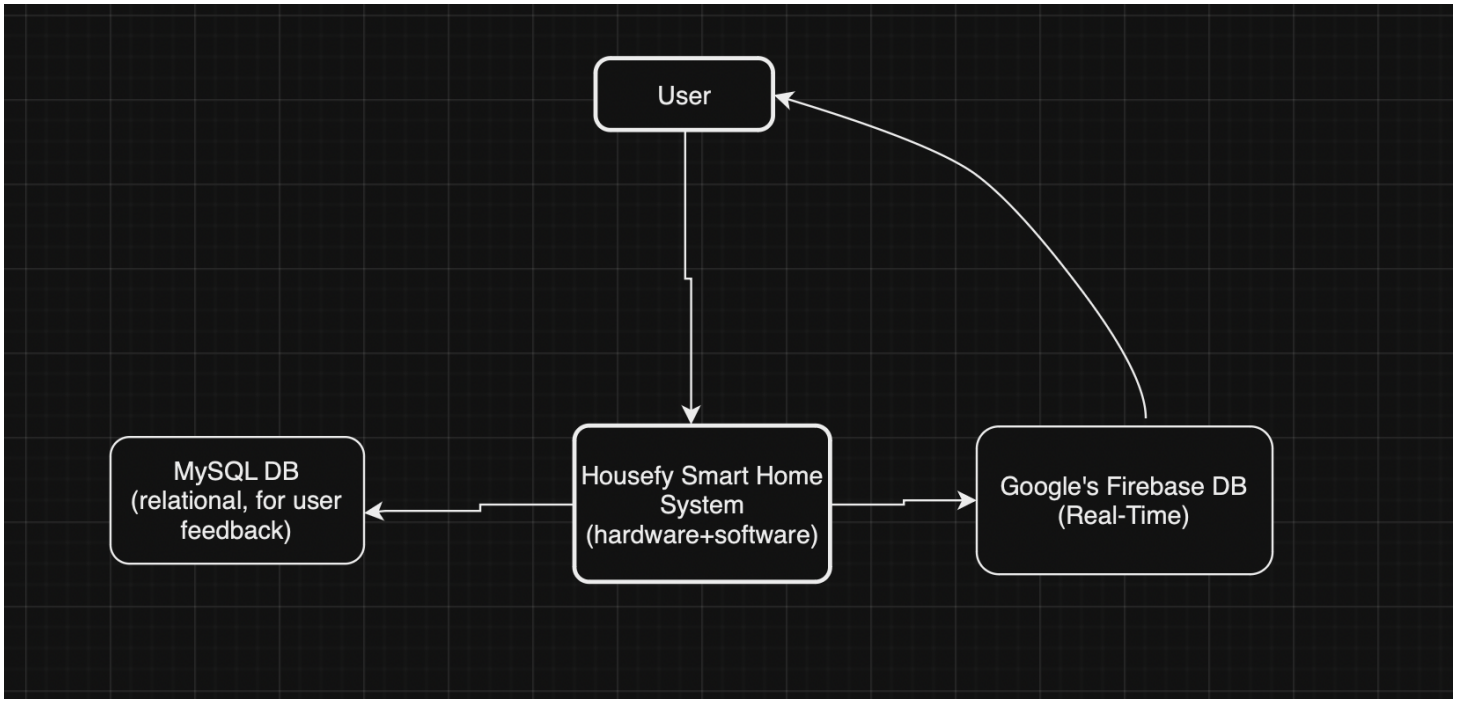
## 8. Sprint Retrospective

https://lucid.app/lucidspark/53fa8641-3694-4821-9f08-e03faf58cab5/edit?viewport_loc=-3259%2C-4466%2C7631%2C4218%2C0_0&invitationId=inv_e7d40b6d-7246-4a1b-87fa-b18e2d9422ec



# 9. System Context Diagram

**Link:** https://bit.ly/draw_io_diagram




Justification:

https://www.youtube.com/watch?v=x2-rSnhpw0g

As per the video fragment from 11:46 to 11:58 from the video above that was shown in class, quote: *"A System Context Diagram basically shows the thing you are working on or building and stuff around it in terms of other systems it talks to and the people who use your system"*. Based on this definition, the diagram above was built. Our hardware device is useless without the app, so they are unified to a single entity representing Housefy Smart Home System. Separation of hardware and software into separate entities is to be done in the next deliverable in the Container Diagram.

## 10. Two design principles

1. Let’s take a look at the Home Page code. Each function (Composable) has a single responsibility. For example, TemperatureCard(), EnergyUsageCard(), and DevicesList() are responsible for their own UI representation and related logic. ***So, the code follows SRP (Single Responsibility) Design Principle.***

```
@Composable
fun TemperatureCard() {
  // All logic and UI related to the TemperatureCard lives here.
  // It doesn't concern itself with anything outside of its scope.
  // This function has a single responsibility - manage the TemperatureCard.
}

@Composable
fun EnergyUsageCard() {
  // This function has a single responsibility - manage the EnergyUsageCard.
  // It doesn't concern itself with anything else.
}

@Composable
```

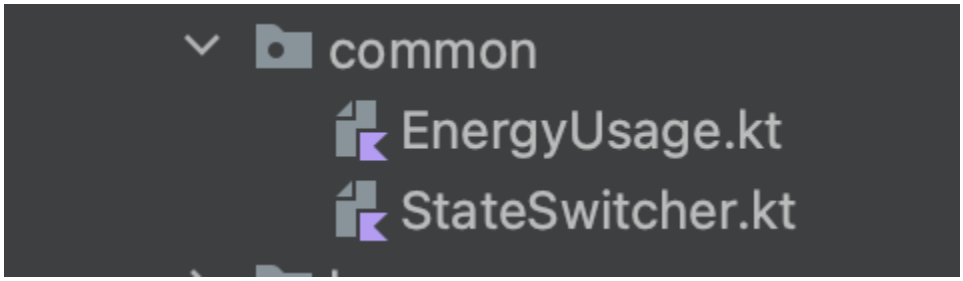
```
fun DevicesList(navController: NavHostController) {

  // The DevicesList function is only responsible for managing the
DevicesList component.

  // It doesn't concern itself with other parts of the UI.

}
```

2. Now let’s take a look at the StateSwitcher and EnergyUsage components inside the common package.




These components are used in the Smart Light and Air Conditioner pages. Thus, DRY (Don’t Repeat Yourself principle is used).

*Excerpt from Smart Light Page:*

```
        StateSwitcher(

            text = "Toggle on/off",

            checked = isLightOn.value,

            onCheckedChange = { isChecked -> isLightOn.value = isChecked },

            modifier = Modifier.weight(1f)

        )
```

*Excerpt from Air Conditioner Page:*

```
        EnergyUsage(
            text = energyUsageText.value,
            modifier = Modifier.weight(1f)
        )
```

## 11. Two design patterns

In the Home Page, a **Factory Pattern** is used: The functions **TemperatureCard()**, **EnergyUsageCard()**, and **DevicesList()** are used like factories. They return a UI component, but the implementation details of these components are hidden.

```
@Composable
fun TemperatureCard(): TemperatureCard {
  // Here we would define the properties and behavior of a TemperatureCard
  // The details are encapsulated within this factory function, and we simply
  // return a new instance of the component.
  return TemperatureCard(/* ... */)
}
@Composable
fun EnergyUsageCard(): EnergyUsageCard {
  // Similarly, this function defines and returns an EnergyUsageCard.
  return EnergyUsageCard(/* ... */)
```

The Smart Light page code code uses the Ambient design pattern with **LightOnAmbient.current**, which propagates the state of the light through the composition tree:

```
val isLightOn = LightOnAmbient.current
```

This ambient value can be accessed and modified directly from any Composable function that has access to it, allowing for a state change like toggling the light on and off.

```
StateSwitcher(text = "Toggle on/off", checked = isLightOn.value,
onCheckedChange = { isChecked -> isLightOn.value = isChecked })
```

## 12. Coding work progress

- Kotlin, Jet Pack Compose and Material Design 3 migration

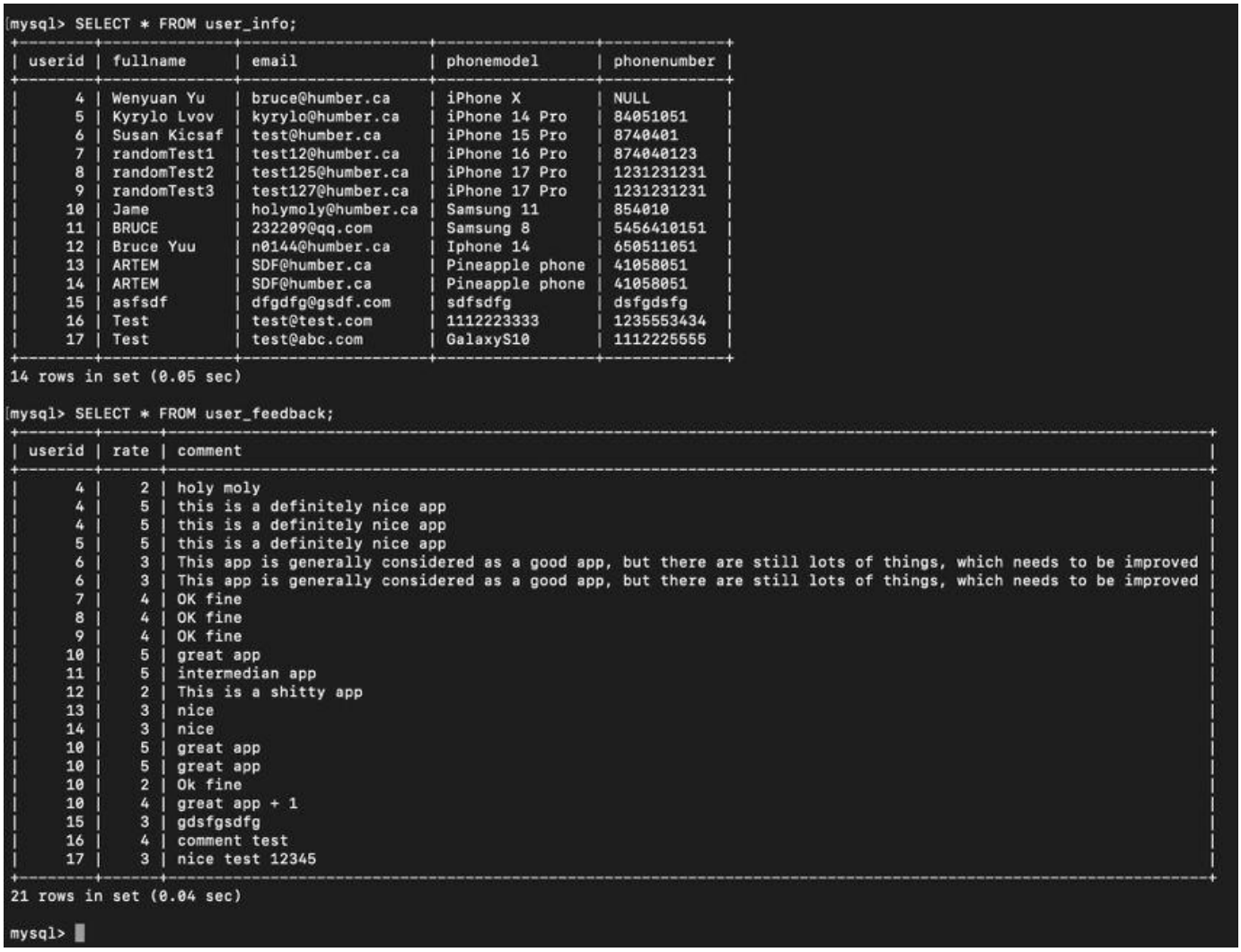
- Reimplementation of the code from Deliverable 1 and 3 in Kotlin, Jet Pack Compose and Material Design 3

- New functionality:
    - Front End:
        - AC page
        - Energy Consumption page
        - About page
        - Feedback Page
        - Guide Page
    - Backend:
        - MySQL hosted on Azure and integrated with front end
        - Real Time Firebase DB set up. An ASP.NET project configured with Firebase so that it can read the data from the db and output to the URL.

## 13. Runtime permission

We use Internet and Location runtime permissions.

## 14. MySQL DB screenshot



```
mysql> SELECT * FROM user_info;
+--------+--------------+--------------------+-----------------+-------------+
| userid | fullname     | email              | phonemodel      | phonenumber |
+--------+--------------+--------------------+-----------------+-------------+
|      4 | Wenyuan Yu   | bruce@humber.ca    | iPhone X        | NULL        |
|      5 | Kyrylo Lvov  | kyrylo@humber.ca   | iPhone 14 Pro   | 84051051    |
|      6 | Susan Kicsaf | test@humber.ca     | iPhone 15 Pro   | 8740401     |
|      7 | randomTest1  | test12@humber.ca   | iPhone 16 Pro   | 874040123   |
|      8 | randomTest2  | test125@humber.ca  | iPhone 17 Pro   | 1231231231  |
|      9 | randomTest3  | test127@humber.ca  | iPhone 17 Pro   | 1231231231  |
|     10 | Jame         | holymoly@humber.ca | Samsung 11      | 854010      |
|     11 | BRUCE        | 232209@qq.com      | Samsung 8       | 5456410151  |
|     12 | Bruce Yuu    | n0144@humber.ca    | Iphone 14       | 650511051   |
|     13 | ARTEM        | SDF@humber.ca      | Pineapple phone | 41058051    |
|     14 | ARTEM        | SDF@humber.ca      | Pineapple phone | 41058051    |
|     15 | asfsdf       | dfgdfg@gsdf.com    | sdfsdfg         | dsfgdsfg    |
|     16 | Test         | test@test.com      | 1112223333      | 1235553434  |
|     17 | Test         | test@abc.com       | GalaxyS10       | 1112225555  |
+--------+--------------+--------------------+-----------------+-------------+
14 rows in set (0.05 sec)

mysql> SELECT * FROM user_feedback;
+--------+------+-----------------------------------------------------------------------------------------------------------------------+
| userid | rate | comment                                                                                                               |
+--------+------+-----------------------------------------------------------------------------------------------------------------------+
|      4 |    2 | holy moly                                                                                                             |
|      4 |    5 | this is a definitely nice app                                                                                         |
|      4 |    5 | this is a definitely nice app                                                                                         |
|      5 |    5 | this is a definitely nice app                                                                                         |
|      6 |    3 | This app is generally considered as a good app, but there are still lots of things, which needs to be improved       |
|      6 |    3 | This app is generally considered as a good app, but there are still lots of things, which needs to be improved       |
|      7 |    4 | OK fine                                                                                                               |
|      8 |    4 | OK fine                                                                                                               |
|      9 |    4 | OK fine                                                                                                               |
|     10 |    5 | great app                                                                                                             |
|     11 |    5 | intermedian app                                                                                                       |
|     12 |    2 | This is a shitty app                                                                                                  |
|     13 |    3 | nice                                                                                                                  |
|     14 |    3 | nice                                                                                                                  |
|     10 |    5 | great app                                                                                                             |
|     10 |    5 | great app                                                                                                             |
|     10 |    2 | Ok fine                                                                                                               |
|     10 |    4 | great app + 1                                                                                                         |
|     15 |    3 | gdsfgsdfg                                                                                                             |
|     16 |    4 | comment test                                                                                                          |
|     17 |    3 | nice test 12345                                                                                                       |
+--------+------+-----------------------------------------------------------------------------------------------------------------------+
21 rows in set (0.04 sec)

mysql>
```

## 15. Additional Notes (Important)

- **Regarding pictures (6 pictures are required):**
  We are using dynamic pictures in our application. The weather component on

Home fragment contains 4 pictures which switch depending on OpenWeatherApi response. The other 2 pictures are on Smart Light fragment, when you toggle the switch, image changes.

- **Regarding menu:**
  As of now, Jetpack Compose (the new way to style Android applications which are build using Kotlin) doesn't come with always, ifroom, never menu items, that's why we needed to create them ourselves, the functionality is the same, you can find the code in /components/navigation/Menu.kt

- **Regarding landscape mode:**
  In the new paradigm (Jetpack Compose), rather than having separate layout files for each orientation, Jetbrains suggest having one, which adapts to any landscape. We still implemented explicit change of layout to landscape mode using if - else statement on Splash and Home screens

```
fun HomePage(navController: NavHostController, snackbarHostState: SnackbarHostState) {
    val configuration = LocalConfiguration.current

    if (configuration.orientation == Configuration.ORIENTATION_LANDSCAPE) {
        HomePageLayout(navController, 16.dp, snackbarHostState)
    } else {
        HomePageLayout(navController, 24.dp, snackbarHostState)
    }
}
```

- **Regarding Gantt chart:**

  A vast majority of tasks in the Gantt Chart are parallel. If you look at the history of commits in out repo, almost every day our team worked on a variety of features and screens, thus working on a wide set of tasks from different stories concurrently, with progress being made everyday. We considered it necessary to leave a separate note regarding this matter, as the Gantt Charts demonstrated in class seemed to follow a more consequent approach.