



- Développement Android -

UNIVERSITÉ
PARIS8
VINCENNES-SAINT-DENIS

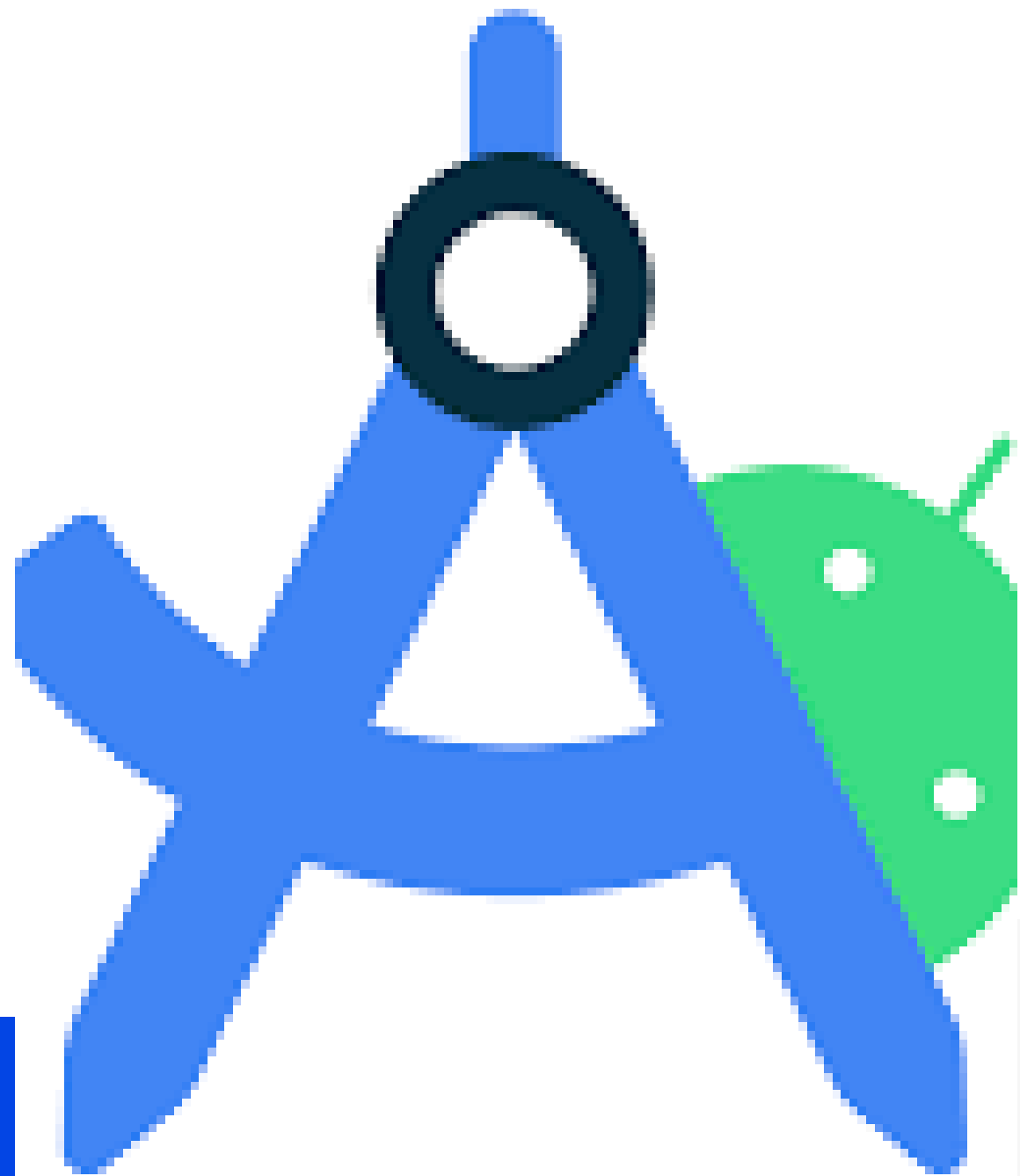
SOUTENANCE DE PROJET

Par ANTON NELCON Steve

Licence 3 Informatique
des Systèmes embarqués et interactif

18 décembre 2024

SOMMAIRE :



01 INTRODUCTION : *Le projet*

02 PRESENTATION DU PROJET : *En quoi consiste mon projet ?*

03 LE FONCTIONNEMENT : *Comment ça fonctionne?*

04 LES FONCTIONNALITÉS : *Les codes*

05 LA DEMONSTRATION : *Comment ça fonctionne?*

06 MON RESSENTIE : *Ce que j'ai apprécié*

07 CONCLUSION : *Pour conclure*



01

INTRODUCTION

1. Le Projet

Pour ce projet, vous développerez une application Android sur Android Studio en utilisant Kotlin



02

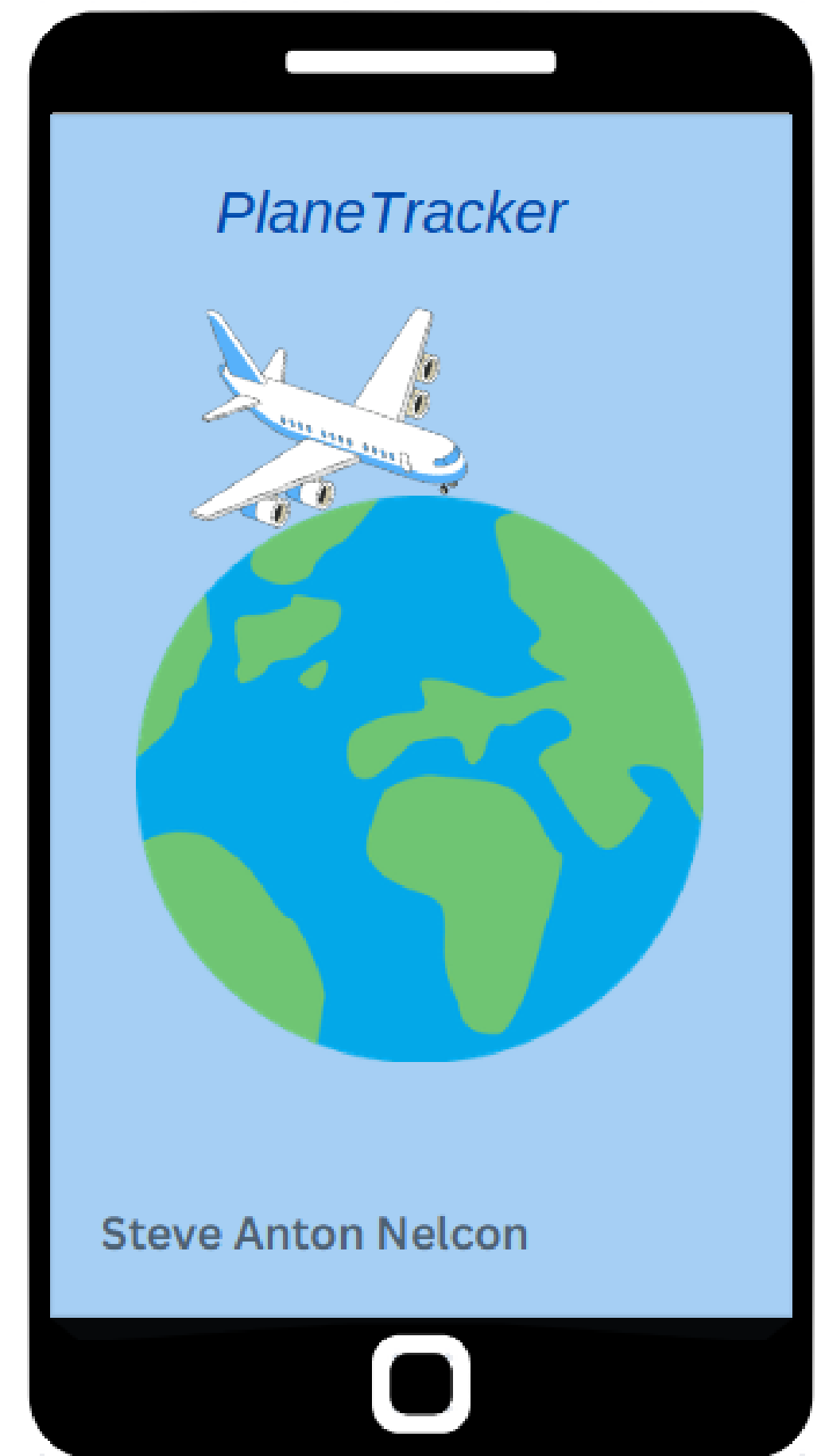
PRÉSENTATION DU PROJET :



2. PRÉSENTATION DU PROJET :

En quoi consiste mon projet ?

- **Nom de l'application:** PlaneTracker
- **Objectif principal de l'application:**
plateforme pour la réservation de vols d'avion.
- **Fonctionnalité de l'application:**
Réservation des vols d'application sur téléphone



03

LE FONCTIONNEMENT



3. LE FONCTIONNEMENT :

LES PAGES

La page d'accueil

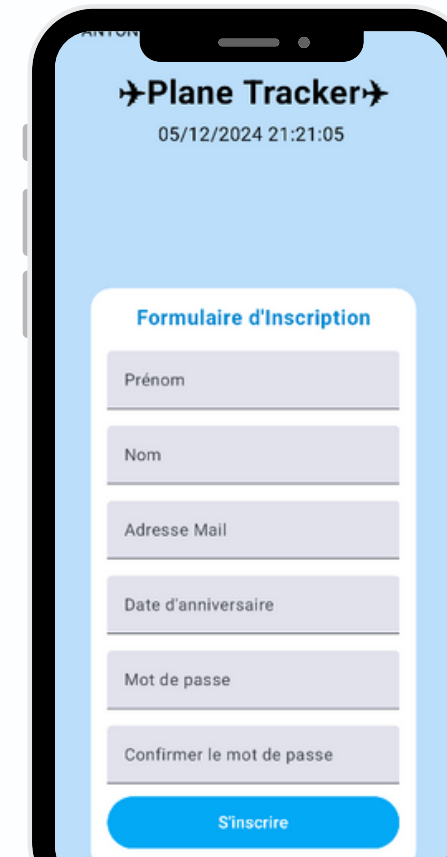


page d'accueil

La page de connexion



La page d'inscription



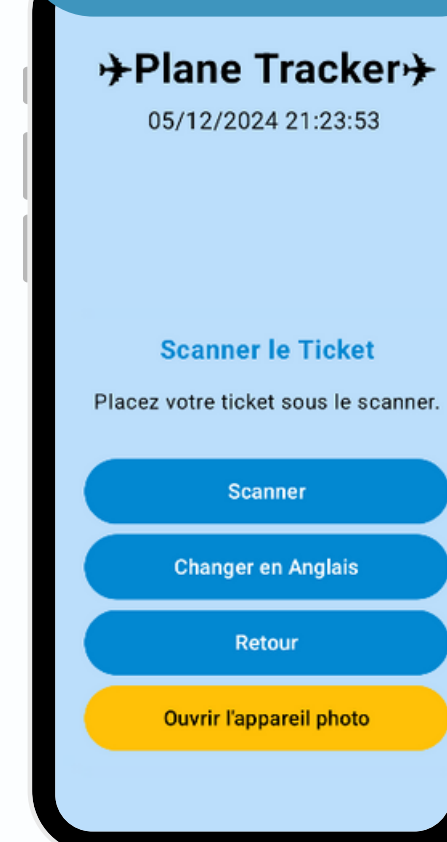
La page de recherche



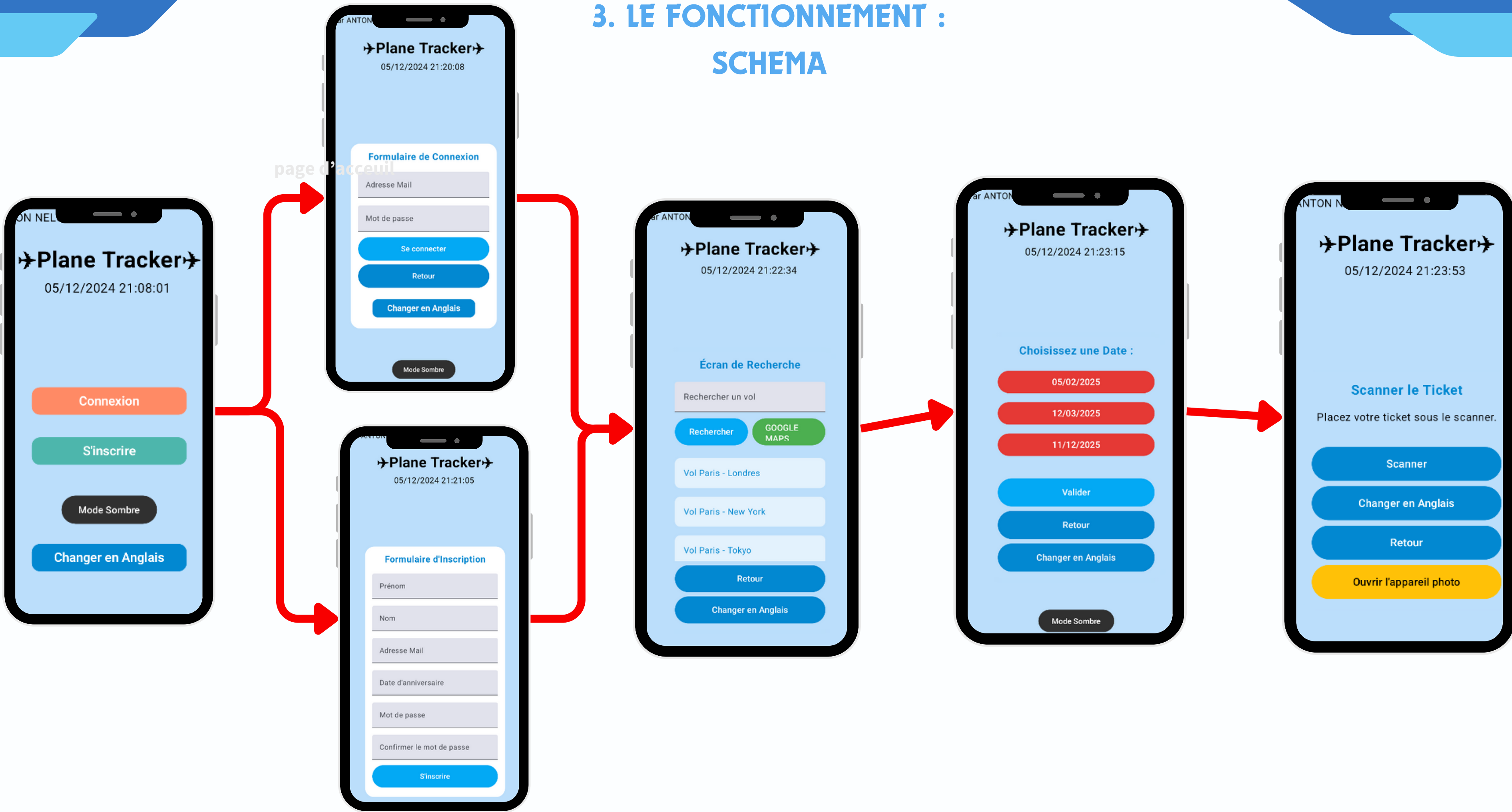
La page des dates



La page scanner



3. LE FONCTIONNEMENT : SCHEMA



04

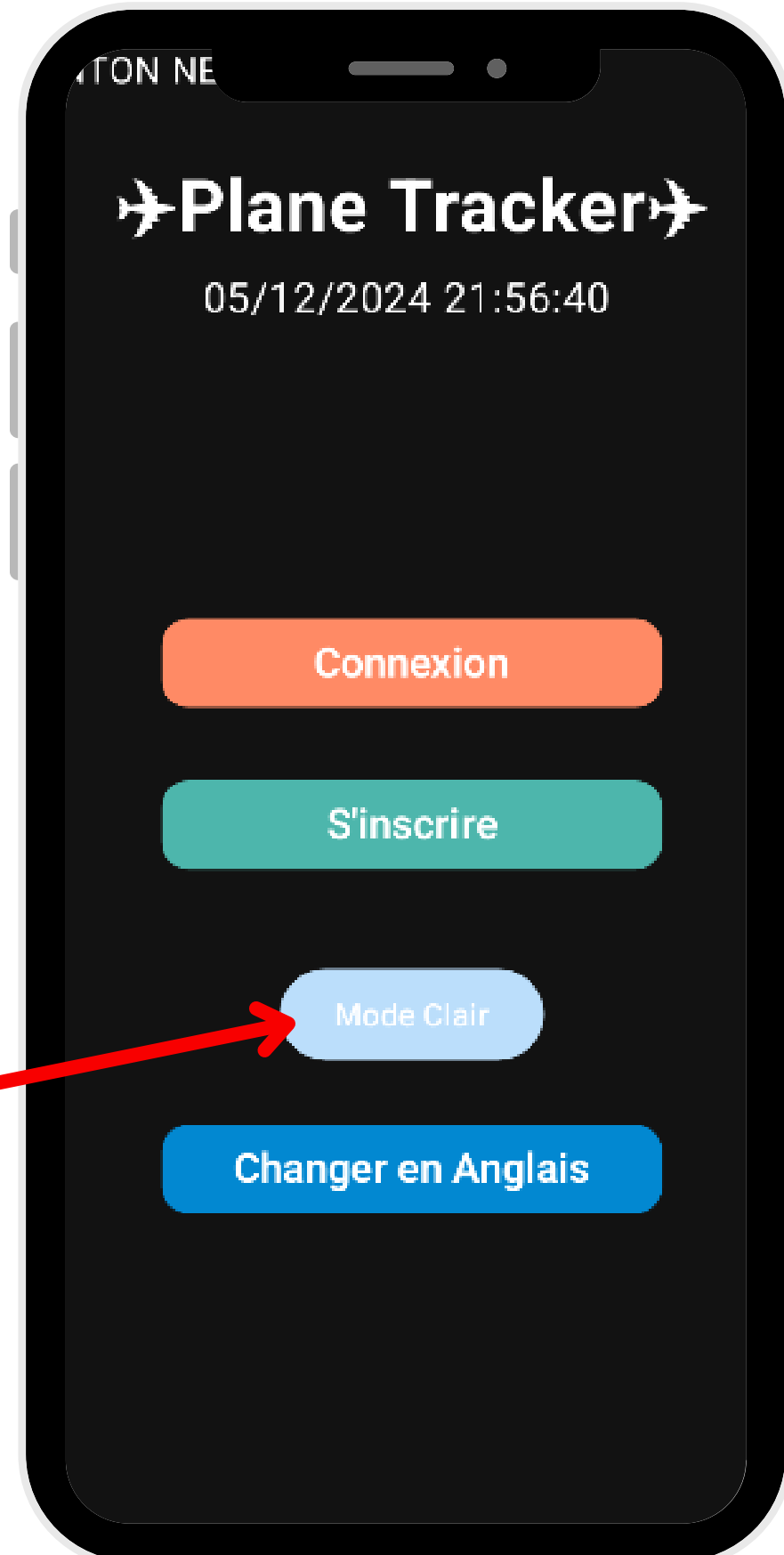
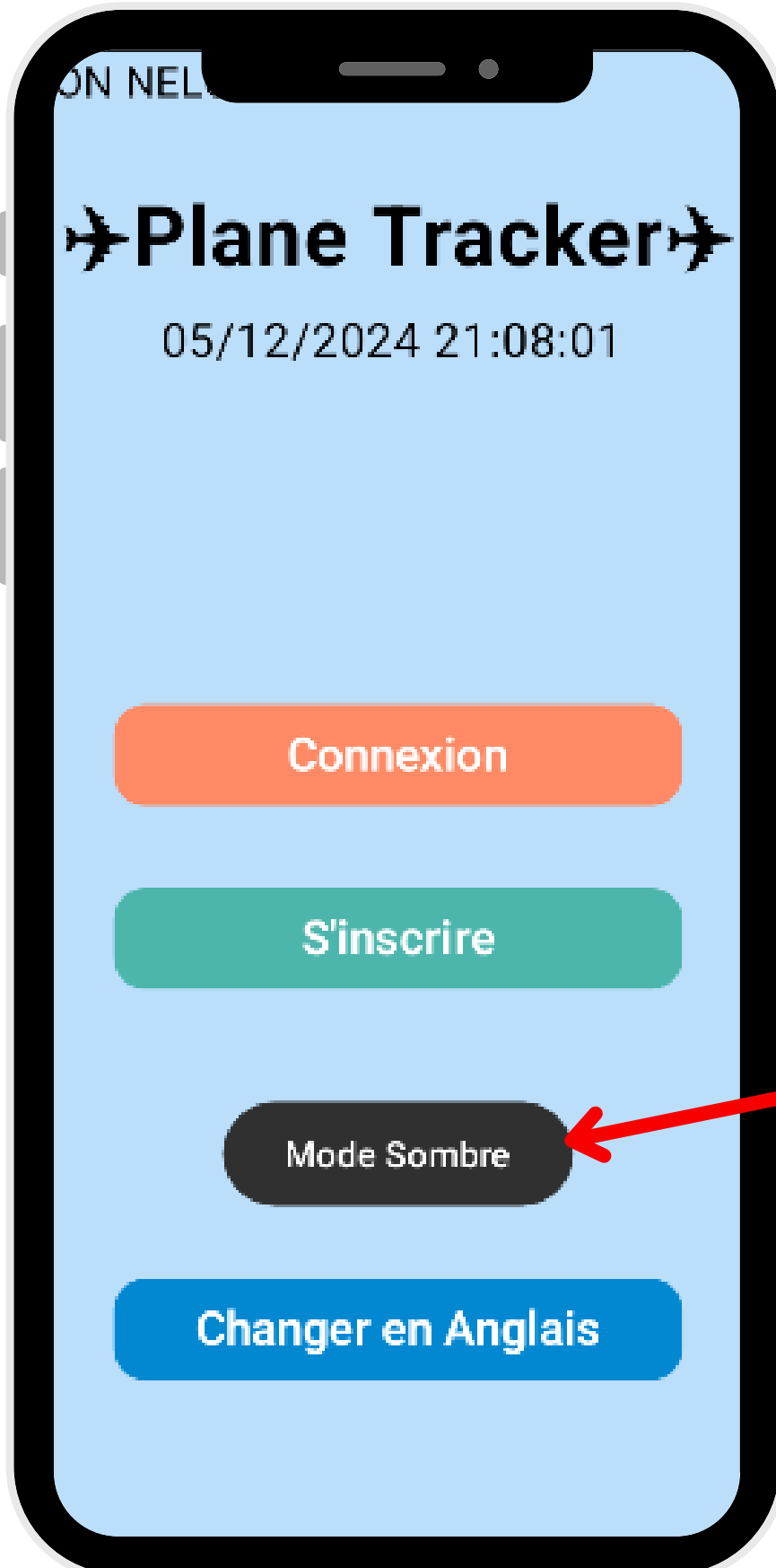
LES FONCTIONNALITES



LES FONCTIONNALITES

MODE CLAIR /SOMBRE

4



LE CODE:

```
// Variable pour le mode clair/sombre
var isDarkMode by remember { mutableStateOf(false) }

// Bouton de bascule clair/sombre
Button(
    onClick = { isDarkMode = !isDarkMode }, // Change l'état du mode clair/sombre lors du
    colors = ButtonDefaults.buttonColors(
        containerColor = if (isDarkMode) Color(0xFFBBDEFB) else Color(0xFF303030) // Coule
    )
) {
    Text(
        text = if (isDarkMode) "Mode Clair" else "Mode Sombre", // Texte du bouton selon
        color = Color.White // Couleur du texte
    )
}

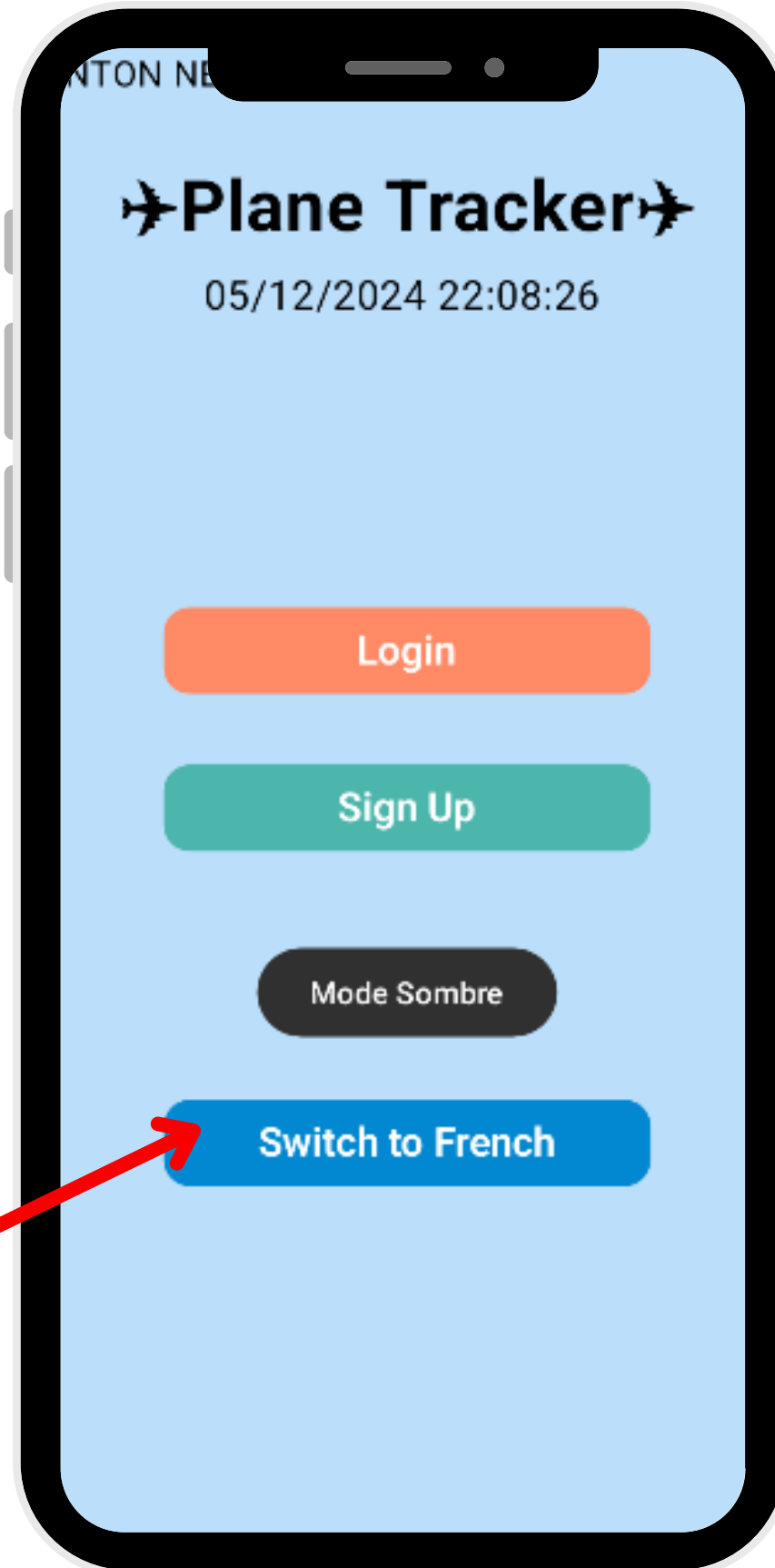
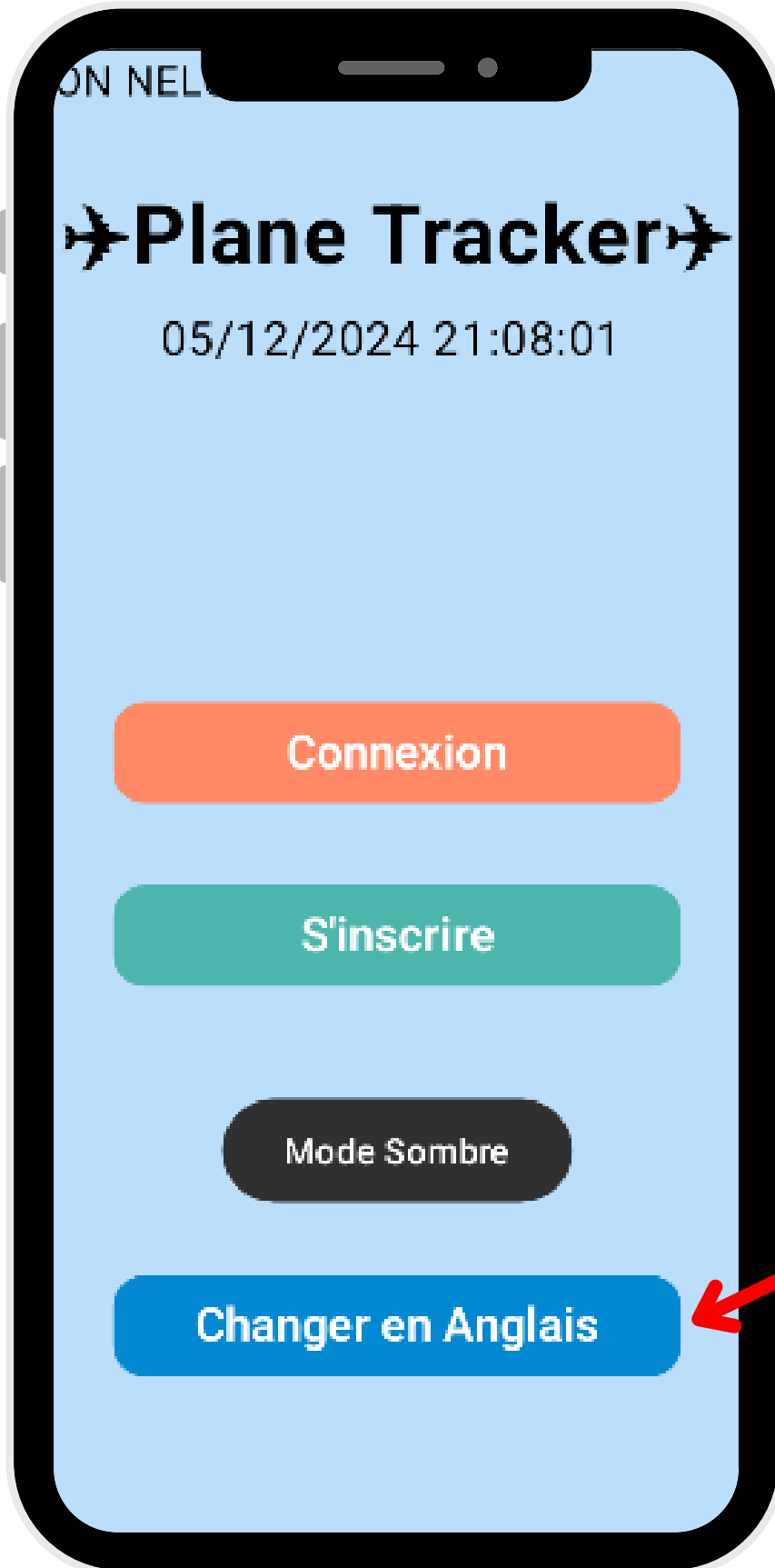
// Application des couleurs selon le mode
val backgroundColor = if (isDarkMode) Color(0xFF121212) else Color(0xFFBBDEFB) // Couleur
val textColor = if (isDarkMode) Color(0xFFFFFFFF) else Color(0xFF000000) // Couleur
```

```
@Composable
fun MainContent(modifier: Modifier = Modifier) {
    le code
}
```

LES FONCTIONNALITES

CHANGEMENT DE LANGUE

4



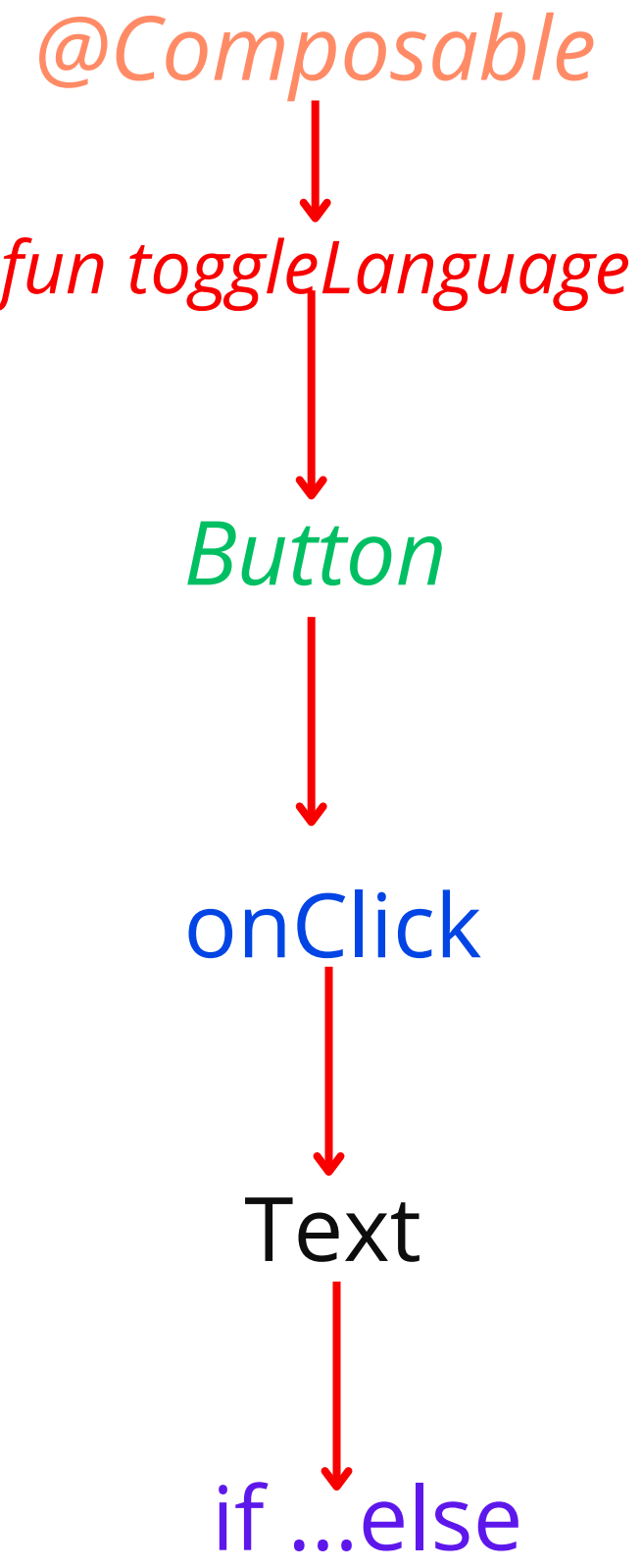
LE CODE:

```
@Composable
fun MainContent() {
    var language by remember { mutableStateOf("fr") } // Variable de langue (français/angl

    // Fonction pour changer la langue
    fun toggleLanguage() {
        language = if (language == "fr") "en" else "fr"
    }

    // Bouton Connexion
    Button(
        onClick = { /* Action Connexion */ },
        colors = ButtonDefaults.buttonColors(containerColor = Color(0xFFFF8A65)),
        shape = RoundedCornerShape(30),
        modifier = Modifier
            .fillMaxWidth()
            .height(55.dp)
            .padding(horizontal = 32.dp, vertical = 8.dp)
    ) {
        Text(
            text = if (language == "fr") "Connexion" else "Login",
            color = Color.White,
            fontSize = 18.sp,
            Spacer(modifier = Modifier.height(16.dp))

    // Bouton pour changer la langue
    Button(
        onClick = { toggleLanguage() },
        colors = ButtonDefaults.buttonColors(containerColor = Color(0xFF0288D1)),
        shape = RoundedCornerShape(30),
        modifier = Modifier
            .fillMaxWidth()
            .height(55.dp)
            .padding(horizontal = 32.dp, vertical = 8.dp)
    ) {
        Text(
            text = if (language == "fr") "Changer en Anglais" else "Switch to French",
            color = Color.White,
            fontSize = 18.sp,
            fontWeight = FontWeight.SemiBold
        )
    }
}
```



LES FONCTIONNALITES

ADRESSE MAIL ERRONEE

4



LE CODE:

MPORT ANDROIDX.COMPOSE.UI.TEXT.INPUT.PASSWORDVISUALTRANSFORMATION
// PERMET DE MASQUER LE MOT DE PASSE DANS UN CHAMP TEXTE.

```
@Composable
fun LoginForm(onBack: () -> Unit, onLoginSuccess: () -> Unit) {
    var email by remember { mutableStateOf("") }
    var password by remember { mutableStateOf("") }
    var message by remember { mutableStateOf("") }
    var errorMessage by remember { mutableStateOf("") }

    Text(
        text = "Formulaire de connexion",
        fontSize = 20.sp,
        fontWeight = FontWeight.Bold,
        color = Color(0xFF0288D1),
        textAlign = TextAlign.Center
    )

    Spacer(modifier = Modifier.height(16.dp))

    TextField(
        value = email,
        onChange = { email = it },
        label = { Text("Adresse Mail") },
        isError = errorMessage.isNotEmpty(),
        modifier = Modifier.fillMaxWidth().padding(vertical = 4.dp)
    )

    TextField(
        value = password,
        onChange = { password = it },
        label = { Text("Mot de passe") },
        visualTransformation = PasswordVisualTransformation(),
        modifier = Modifier.fillMaxWidth().padding(vertical = 4.dp)
    )

    Spacer(modifier = Modifier.height(8.dp))

    Button(
        onClick = {
            errorMessage = ""
            if (!email.contains(".com")) {
                errorMessage = "Veuillez écrire un e-mail valide (.com)"
            } else if (email.isNotBlank() && password.isNotBlank()) {
                message = "Connexion réussie"
                println("Connexion avec $email réussie")
                onLoginSuccess()
            } else {
                errorMessage = "Veuillez remplir tous les champs"
            }
        },
        colors = ButtonDefaults.buttonColors(Color(0xFF03A9F4)),
        shape = RoundedCornerShape(8.dp),
        modifier = Modifier.fillMaxWidth().height(50.dp)
    ) {
        Text(text = "Se connecter", color = Color.White, fontSize = 16)
    }

    if (message.isNotEmpty()) {
        Text(
            text = message,
            color = Color.Green,
            fontSize = 14.sp,
            modifier = Modifier.align(Alignment.Start)
        )
    }
}
```

```
TextField(
    value = password,
    onChange = { password = it },
    label = { Text("Mot de passe") },
    visualTransformation = PasswordVisualTransformation(),
    modifier = Modifier.fillMaxWidth().padding(vertical = 4.dp)
)

Spacer(modifier = Modifier.height(8.dp))

Button(
    onClick = {
        errorMessage = ""
        if (!email.contains(".com")) {
            errorMessage = "Veuillez écrire un e-mail valide (.com)"
        } else if (email.isNotBlank() && password.isNotBlank()) {
            message = "Connexion réussie"
            println("Connexion avec $email réussie")
            onLoginSuccess()
        } else {
            errorMessage = "Veuillez remplir tous les champs"
        }
    },
    colors = ButtonDefaults.buttonColors(Color(0xFF03A9F4)),
    shape = RoundedCornerShape(8.dp),
    modifier = Modifier.fillMaxWidth().height(50.dp)
) {
    Text(text = "Se connecter", color = Color.White, fontSize = 16)
}

if (message.isNotEmpty()) {
    Text(
        text = message,
        color = Color.Green,
        fontSize = 14.sp,
    )
}
```

ADRESSE MAIL :

VALUE= EMAIL

MOT DE PASSE

VALUE=PASSWORD

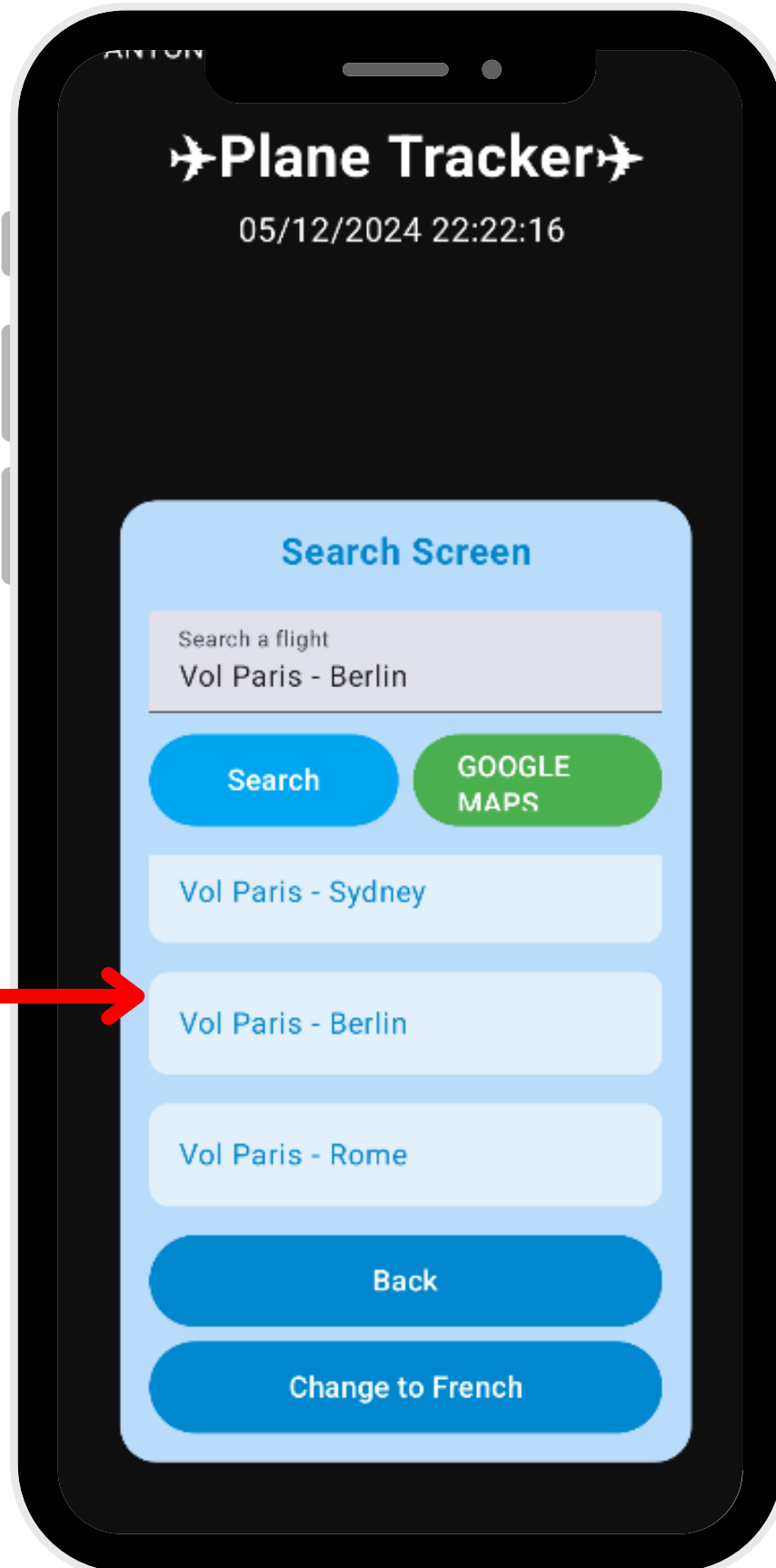
IF (!EMAIL.CONTAINS(".COM")) { ERRORMESSAGE =
"VEUILLEZ ÉCRIRE UN E-MAIL VALIDE (.COM)" }

ELSE IF (EMAIL.ISNOTBLANK() && PASSWORD.ISNOTBLANK())
= IDENTIFIANT ET MOT DE PASSE VALIDE
ONLOGINSUCCESS() ELSE { ERRORMESSAGE =
"VEUILLEZ REMPLIR TOUS LES CHAMPS" }

LES FONCTIONNALITES

LISTE DEROULANTE

4



LE CODE:

```
import androidx.compose.foundation.lazy.LazyColumn // Permet d'afficher une liste déroulante
import androidx.compose.foundation.lazy.items // Permet de créer une liste d'éléments
import androidx.compose.foundation.clickable
```

```
// Variable pour stocker la valeur de la barre de recherche
var searchQuery by remember { mutableStateOf("") }

// Liste avec un seul élément : Vol Paris - Londres
val flights = listOf(
    "Vol Paris - Londres"
)

Column {
    // Barre de recherche située en haut
    TextField(
        value = searchQuery, // La valeur actuelle de la barre de recherche
        onChange = { searchQuery = it }, // Mettre à jour la valeur de recherche
        label = { Text("Rechercher un vol") }, // Texte affiché dans la barre
        modifier = Modifier
            .fillMaxWidth() // La barre occupe toute la largeur de l'écran
            .padding(horizontal = 16.dp, vertical = 8.dp) // Espacement autour de la
    )

    // Liste défilante avec les vols
    LazyColumn(
        modifier = Modifier
            .fillMaxWidth() // La liste défilante occupe toute la largeur de l'écran
            .height(200.dp) // La hauteur de la liste est limitée à 200dp
    ) {
        // Parcourir chaque élément de la liste 'flights' pour l'afficher dans la liste
        items(flights) { flight ->
            Text(
                text = flight, // Le texte du vol
                modifier = Modifier
                    .fillMaxWidth() // Le texte occupe toute la largeur de la liste
                    .padding(vertical = 8.dp) // Un espacement vertical de 8dp autour
                    .background(Color(0xFFE3F2FD), shape = RoundedCornerShape(10.dp))
                    .padding(16.dp) // Un padding interne de 16dp
                    .clickable { // Lorsqu'on clique sur le vol
                        searchQuery = flight // Met à jour la barre de recherche avec
                    },
                color = Color(0xFF0288D1), // La couleur du texte est bleu
                fontSize = 16.sp // La taille de la police est de 16sp
            )
        }
    }
}
```

VAR FLIGHT=LISTOF
AFFICHER LES LIEUX

LABEL:
BARRE DE RECHERCHE

LAZYCOLUMN=
DEFLIMENT
.CLICKABLE
BOUTON CLIQUABLE

SEARCHQUERY
AFFICHER SUR LA BARRE DE RECHERCHE

LES FONCTIONNALITES

CALENDRIER

4



LE CODE: **DROPDOWNNEU** // LISTE DEROULANTE.

```
@Composable
fun DateTimePickerScreen() {
    var selectedDate by remember { mutableStateOf("") }
    var selectedTime by remember { mutableStateOf("") }
    var showTimePicker by remember { mutableStateOf(false) }

    // Liste des dates et heures disponibles
    val availableDates = listOf("05/02/2025", "12/03/2025", "11/12/2025")
    val availableTimes = listOf("08:00", "12:00", "16:00")

    Text(
        text = "Choisissez une Date:",
        fontSize = 18.sp,
        fontWeight = FontWeight.Bold
    )

    DropdownMenu(
        expanded = true,
        onDismissRequest = {},
        modifier = Modifier.fillMaxWidth()
    ) {
        availableDates.forEach { date ->
            DropdownMenuItem(onClick = {
                selectedDate = date
                showTimePicker = true
            }) {
                Text(text = date)
            }
        }
    }
}
```

03 VARAIBLES+ 2 VARIABLES LIST OF
AFFICHE LA DATE ET HEURES

ONCLICK =BOUTON VERT
+DROPDOWNMENU =PASSER A L' HEURE

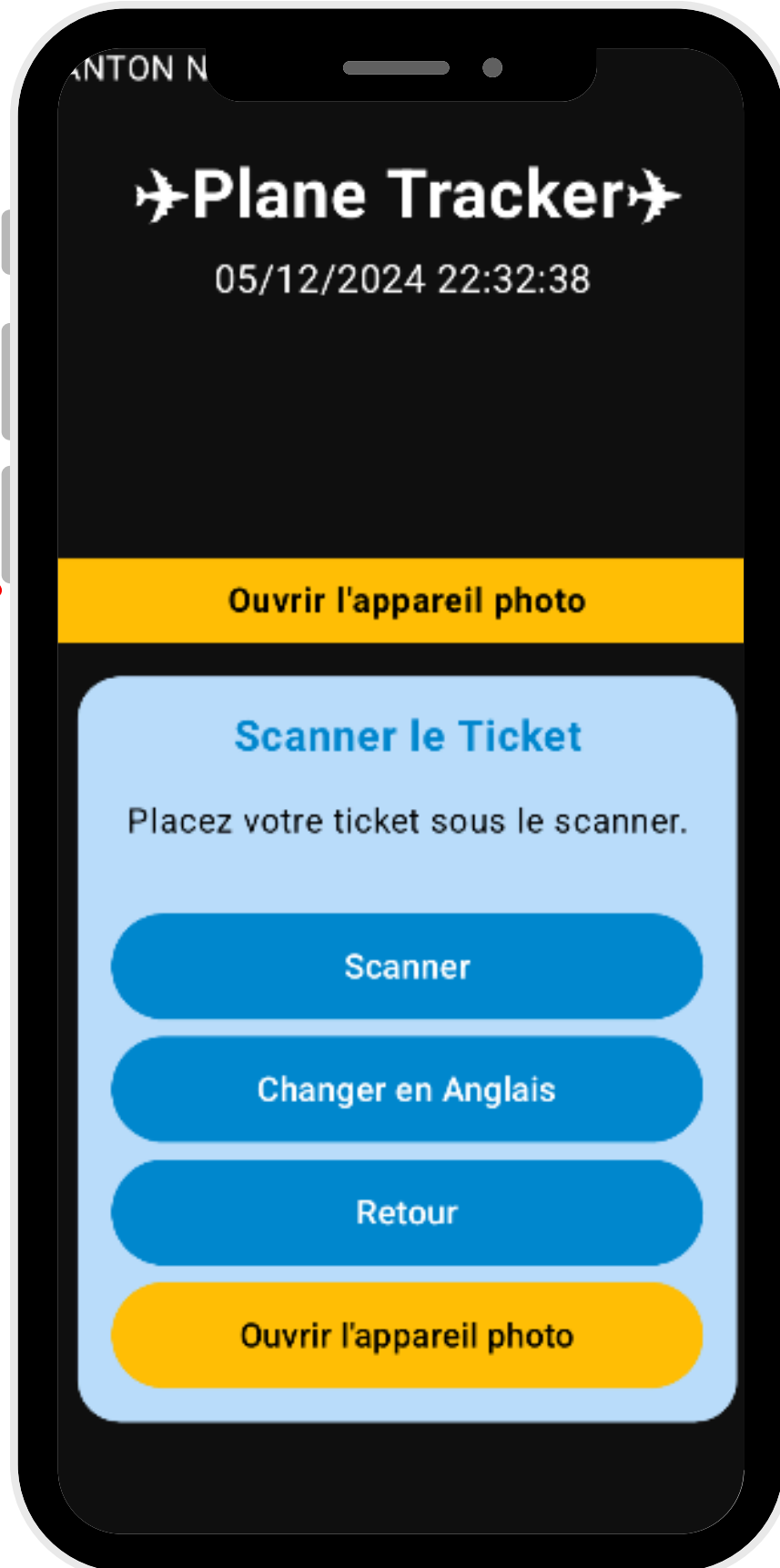
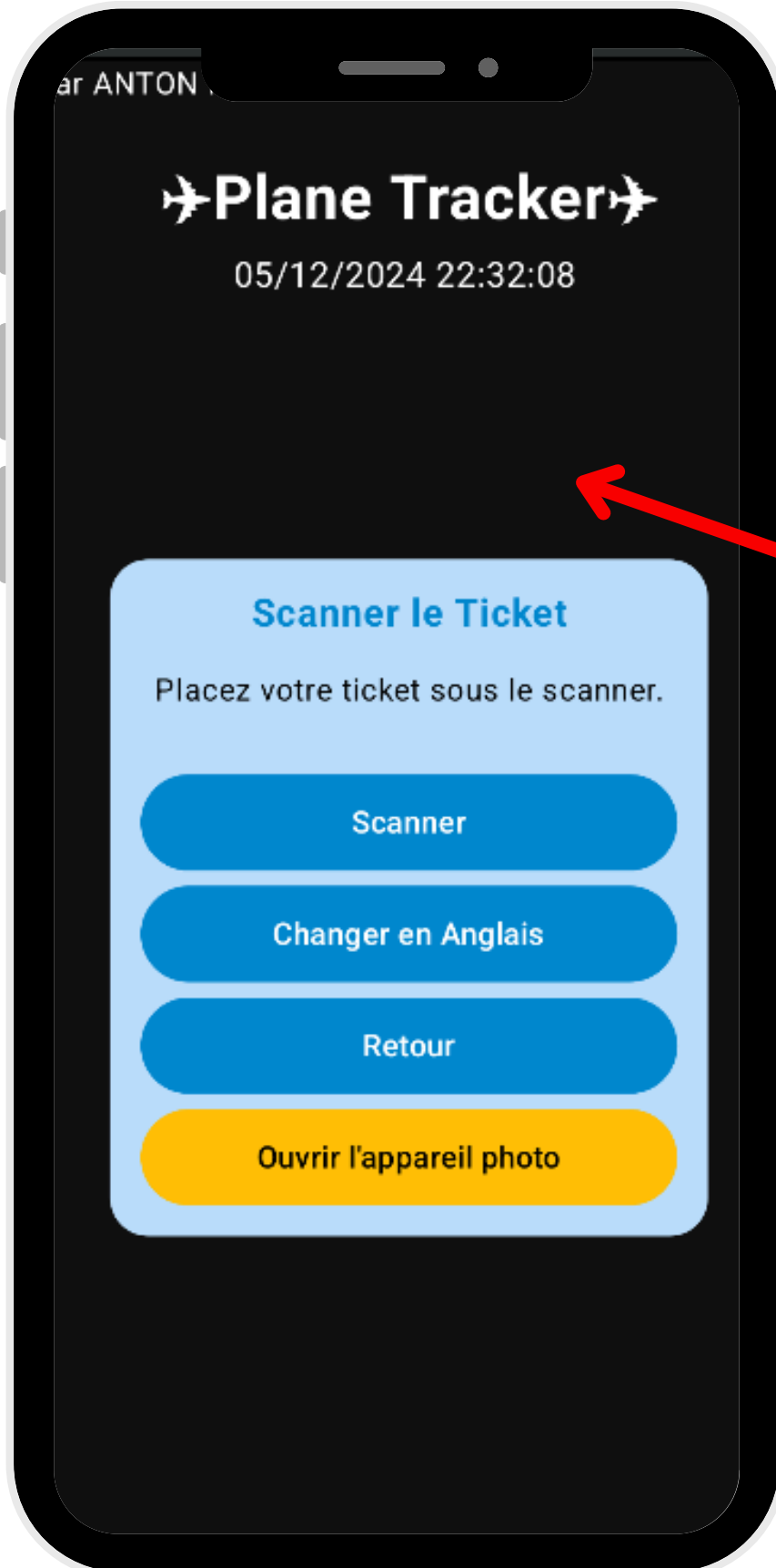
CHOIX DE PLUSIEURS DATE ET HEURE

IF SHOWTIMERPICKER
DROPDOWNMENU= TRUE
AFFICHE LES HEURES

```
if (showTimePicker) {
    Text(
        text = "Choisissez une Heure:",
        fontSize = 18.sp,
        fontWeight = FontWeight.Bold
    )
    DropdownMenu(
        expanded = true,
        onDismissRequest = {},
        modifier = Modifier.fillMaxWidth()
    ) {
        availableTimes.forEach { time ->
            DropdownMenuItem(onClick = { selectedTime = time }) {
                Text(text = time)
            }
        }
    }
}
```

LES FONCTIONNALITES NOTIFICATION

4



LE CODE:

```
var showBanner by remember { mutableStateOf(false) }

// Nouveau bouton pour ouvrir l'appareil photo
Button(
    onClick = {
        showBanner = true // Afficher la bannière de notification
        LaunchedEffect(Unit) {
            delay(3000) // Attendre 3 secondes
            showBanner = false // Cacher la bannière après 3 secondes
        }
    },
    colors = ButtonDefaults.buttonColors(Color(0xFFFFC107)),
    shape = RoundedCornerShape(50),
    modifier = Modifier
        .fillMaxWidth()
        .height(50.dp)
) {
    Text(
        text = if (language == "fr") "Ouvrir l'appareil photo" else "Open Camera",
        color = Color.Black,
        fontSize = 16.sp
    )
}

// Affichage de la bannière
if (showBanner) {
    Box(
        modifier = Modifier
            .fillMaxWidth()
            .background(Color(0xFFFFC107))
            .padding(8.dp),
        contentAlignment = Alignment.Center
    ) {
        Text(
            text = if (language == "fr") "Appareil photo ouvert" else "Camera Open",
            color = Color.Black,
```

DELAY (3000)
DISPARAITION NOTIF

ONCLICK
BOUTON JAUNE

SHOWBANNER
APPARITION DE
LA BANNIERE

05

DEMONSTRATION



5. DÉMONSTRATION

Par ANTON NELCON Steve

✈️ Plane Tracker ✈️

07/12/2024 09:42:48

Formulaire d'Inscription

Prénom
Steve



Nom

Adresse Mail

Date d'anniversaire

Mot de passe

Confirmer le mot de passe

S'inscrire

Retour

EN VIDEO

06

MON RESSENTIE



6. MON RESSENTIE

- Ce que j'ai apprécié
- Ce que je n'ai pas apprécié
- Mes facilités
- Mes difficultés



- FAIRE UNE APPLICATION ANDROID
- Découvrir Kotlin



- LE TEMPS



- TRAVAILLER EN AUTONOMIE
- CHERCHER SUR INTERNET
- LIBRE ARBITRE



- Savoir Kotlin
- Utiliser Android Studio

07

CONCLUSION



7. CONCLUSION

A AMELIORER



- **GOOLE MAPS**

NAVIGUER



- **CALENDRIER**

CALENDRIER INTERACTIF



- **APPAREIL PHOTO**

SCANNER



- **BASE DE DONNE**

CRÉATION DE COMPTE

ERGIE DE L'UNITÉ.

OUR RÉUSSIR : LA SYNERGIE QUI PROPULSERA NOTRE PROJET VERS L'AVANT.



MERCI DE VOTRE ECOUTE

Par ANTON NELCON Steve
L3 ISEI