

```
1 package com.example.bloodbank
2
3 import android.app.DatePickerDialog
4 import android.content.Intent
5 import android.os.Build
6 import android.os.Bundle
7 import android.widget.Toast
8 import androidx.activity.ComponentActivity
9 import androidx.activity.compose.setContent
10 import androidx.annotation.RequiresApi
11 import androidx.compose.foundation.Image
12 import androidx.compose.foundation.clickable
13 import androidx.compose.foundation.layout.*
14 import androidx.compose.material.*
15 import androidx.compose.material.icons.Icons
16 import androidx.compose.material.icons.filled.
    DateRange
17 import androidx.compose.material3.Button
18 import androidx.compose.material3.ButtonDefaults
19 import androidx.compose.material3.
    ExperimentalMaterial3Api
20 import androidx.compose.material3.Icon
21 import androidx.compose.material3.IconButton
22 import androidx.compose.material3.OutlinedTextField
23 import androidx.compose.material3.Surface
24 import androidx.compose.material3.Text
25 import androidx.compose.material3.TextField
26 import androidx.compose.runtime.*
27 import androidx.compose.ui.Alignment
28 import androidx.compose.ui.Modifier
29 import androidx.compose.ui.graphics.Color
30 import androidx.compose.ui.platform.LocalContext
31 import androidx.compose.ui.res.painterResource
32 import androidx.compose.ui.text.font.FontWeight
33 import androidx.compose.ui.text.style.TextAlign
34 import androidx.compose.ui.unit.dp
35 import androidx.compose.ui.unit.sp
36 import androidx.compose.ui.window.Dialog
37 import androidx.navigation.NavHostController
38 import androidx.navigation.NavType
39 import androidx.navigation.compose.NavHost
```

```

40 import androidx.navigation.compose.composable
41 import androidx.navigation.compose.
    rememberNavController
42 import androidx.navigation.navArgument
43 import com.example.bloodbank.ui.theme.BloodBankTheme
44 import java.text.SimpleDateFormat
45 import java.time.LocalDate
46 import java.time.ZoneId
47 import java.time.format.DateTimeFormatter
48 import java.util.*
49
50 class MainActivity : ComponentActivity() {
51     @RequiresApi(Build.VERSION_CODES.O)
52     override fun onCreate(savedInstanceState: Bundle
        ?) {
53         super.onCreate(savedInstanceState)
54         setContent {
55             BloodBankTheme {
56                 // A surface container using the '
background' color from the theme
57                 Surface(
58                     modifier = Modifier.fillMaxSize
59                     (),
60                     color = Color(201, 85, 85, 0)
61                 ) {
62                     BloodBankApp()
63                 }
64             }
65         }
66     }
67     @RequiresApi(Build.VERSION_CODES.O)
68     @Composable
69     fun BloodBankApp() {
70         val navController = rememberNavController()
71
72         NavHost(navController, startDestination = "login"
        ) {
73             composable("login") { Bank(navController) }
74             composable("signIn") { SignIn(navController
        ) }

```

```

75         //composable("signUp") { SignUp(
            navController)}
76     }
77 }
78
79
80
81 @OptIn(ExperimentalMaterial3Api::class)
82 @Composable
83 fun Bank(navController: NavHostController) {
84     var username by remember { mutableStateOf("") }
85     var password by remember { mutableStateOf("") }
86     val context = LocalContext.current
87
88     Column(modifier = Modifier.padding(20.dp)) {
89         Column(
90             modifier = Modifier
91                 .padding(horizontal = 16.dp)
92                 .fillMaxWidth(),
93             horizontalAlignment = Alignment.
CenterHorizontally,
94             verticalArrangement = Arrangement.Center
95         )
96         {
97             Image(
98                 painter = painterResource(id = R.
drawable.logo),
99                 contentDescription = null,
100                 Modifier.height(height = 118.dp)
101             )
102             Spacer(modifier = Modifier.height(16.dp
        ))
103             Text(text = "Welcome Doner", fontSize =
34.sp, fontWeight = FontWeight.Bold)
104         }
105         Column(
106             modifier = Modifier
107                 .padding(horizontal = 10.dp),
108             horizontalAlignment = Alignment.Start,
109             verticalArrangement = Arrangement.Center
110         )

```

```

111         {
112             Spacer(modifier = Modifier.height(20.dp
113             ))
114             Text(text = "Username", fontSize = 24.sp
115             )
116             OutlinedTextField(
117                 value = username,
118                 onChange = { username = it },
119                 modifier = Modifier.fillMaxWidth(),
120             )
121             Spacer(modifier = Modifier.height(16.dp
122             ))
123             Text(text = "Password", fontSize = 24.sp
124             )
125             OutlinedTextField(
126                 value = password,
127                 onChange = { password = it },
128                 modifier = Modifier.fillMaxWidth(),
129             )
130             Spacer(modifier = Modifier.height(16.dp
131             ))
132             Row(
133                 modifier = Modifier.fillMaxWidth(),
134                 verticalAlignment = Alignment.
135                 CenterVertically,
136                 horizontalArrangement = Arrangement.
137                 Center
138             ) {
139                 Text(text = "Donate Blood", fontSize
140                 = 18.sp)
141                 Text(text = "Save Life", fontSize =
142                 18.sp, modifier = Modifier.padding(start = 4.dp))
143             }
144             Spacer(modifier = Modifier.height(24.dp
145             ))
146             Button(
147                 onClick = {
148                     Toast.makeText(
149                         context,
150                         "${username}\n${password}",

```

```
142                Toast.LENGTH_LONG
143                ).show()
144                navController.navigate("signIn")
145            },
146            modifier = Modifier.fillMaxWidth(),
147            colors = ButtonDefaults.buttonColors
148            (Color.DarkGray)
149        ) {
150            Text(
151                text = "Sign In",
152                fontSize = 24.sp,
153                color = Color.White,
154                modifier = Modifier
155                    .padding(start = 4.dp)
156            )
157        }
158        Spacer(modifier = Modifier.height(34.dp
159    ))
160    Row(
161        modifier = Modifier.fillMaxWidth(),
162        verticalAlignment = Alignment.
163        CenterVertically,
164        horizontalArrangement = Arrangement.
165        Center
166    ) {
167        Text(text = "New Member?", fontSize
168            = 18.sp)
169        Text(
170            text = "Sign Up",
171            fontSize = 18.sp,
172            color = Color.Blue,
173            modifier = Modifier
174                .padding(start = 14.dp)
175                .clickable { navController.
176                    navigate("signUp") }
177        )
178    }
179 }
```

```

177 @OptIn(ExperimentalMaterial3Api::class)
178 @RequiresApi(Build.VERSION_CODES.0)
179 @Composable
180 fun SignIn(navController: NavHostController) {
181     var name by remember { mutableStateOf("") }
182     var age by remember { mutableStateOf("") }
183     var group by remember { mutableStateOf("") }
184     var lastDonationDate by remember {
185         mutableStateOf<LocalDate?>(null) }
186     val context = LocalContext.current
187     val selectedDate = remember { mutableStateOf<
188         LocalDate?>(null) }
189     var isDatePickerVisible by remember {
190         mutableStateOf(false) }
191
192     Column(modifier = Modifier
193         .padding(20.dp)
194         .fillMaxSize()) {
195         Column(modifier=Modifier.weight(4f),
196             verticalArrangement = Arrangement.Center
197             ,
198             horizontalAlignment = Alignment.
199             CenterHorizontally) {
200             Text(text = "Schedule your blood
201             Donation Date",fontSize = 32.sp, lineHeight = 35.sp,
202                 textAlign = TextAlign.Center)
203         }
204         Column(modifier= Modifier
205             .weight(7f)
206             .fillMaxWidth(),
207             verticalArrangement =Arrangement.
208             SpaceBetween,
209             ) {
210             Column() {
211                 Text(text = "Name",fontSize = 24.sp
212                 )
213                 OutlinedTextField(
214                     value = name,
215                     onChange = { name = it },
216                     modifier = Modifier.fillMaxWidth
217                     ())
218             }
219         }
220     }

```

```

209         }
210         Column() {
211             Text(text = "Age", fontSize = 24.sp)
212             OutlinedTextField(
213                 value = age,
214                 onChange = { age = it },
215                 modifier = Modifier.fillMaxWidth
216             ),
217         }
218         Column() {
219             Text(text = "Blood Group", fontSize
220                 = 24.sp)
221             OutlinedTextField(
222                 value = group,
223                 onChange = { group = it },
224                 modifier = Modifier.fillMaxWidth
225             ),
226         }
227     }
228     Column(modifier = Modifier.weight(5f)){
229         Column(modifier=Modifier.weight(1f)) {
230
231         }
232         Column(modifier = Modifier.weight(9f)) {
233             Text(text = "Date of Donation",
234                 fontSize = 24.sp)
235             Box(modifier = Modifier.clickable {
236                 isDatePickerVisible = true }) {
237                 Text(
238                     text = selectedDate.value?.
239                     format(DateTimeFormatter.ofPattern("dd-MM-yyyy"
240                     )) ?: "Select Last Donation Date",
241                     modifier = Modifier.padding(
242                         top = 16.dp),
243                     color = if (selectedDate.
244                         value != null) Color.Black else Color.LightGray
245                 )
246             }
247         }
248     }

```

```
241
242         if (isDatePickerVisible) {
243             Dialog(
244                 onDismissRequest = {
245                     isDatePickerVisible = false }
246                 ) {
247                     DatePicker(
248                         selectedDate =
249                         selectedDate,
250                         onDateSelected = { date
251                             ->
252                                 selectedDate.value
253                                 = date
254                                 isDatePickerVisible
255                                 = false
256                             }
257                         )
258                     }
259                     Spacer(modifier = Modifier.
260                         height(26.dp))
261                     }
262                     Spacer(modifier = Modifier.height(24
263                         .dp))
264                     Button(
265                         onClick = {
266                             Toast.makeText(
267                                 context,
268                                 "Scheduled on the given
269                                 date",
270                                 Toast.LENGTH_LONG
271                                 ).show()
272                                 name = ""
273                                 age = ""
274                                 group = ""
275                                 selectedDate.value = null
276                             }
277                         )
278                     {
279                         Text(text = "Submit")
280                     }
281                 }
282             }
```



```

274
275     }
276 }
277 }
278 @OptIn(ExperimentalMaterial3Api::class)
279 @RequiresApi(Build.VERSION_CODES.O)
280 @Composable
281 fun DatePicker(selectedDate: MutableState<LocalDate
    ?>, onDateSelected: (LocalDate) -> Unit) {
282     val context = LocalContext.current
283     val dateFormatter = remember { SimpleDateFormat(
284         "dd-MM-yyyy", Locale.getDefault()) }
285     val calendar = Calendar.getInstance()
286     val onClick: () -> Unit = {
287         val datePickerDialog = DatePickerDialog(
288             context,
289             { _, year, month, day ->
290                 calendar.set(year, month, day)
291                 val date = calendar.time
292                 selectedDate.value = date.toInstant
293                     ().atZone(ZoneId.systemDefault()).toLocalDate()
294             },
295             calendar.get(Calendar.YEAR),
296             calendar.get(Calendar.MONTH),
297             calendar.get(Calendar.DAY_OF_MONTH)
298         )
299         datePickerDialog.show()
300     }
301     TextField(
302         value = selectedDate.value?.let {
303             dateFormatter.format(Date.from(it.atStartOfDay(
304                 ZoneId.systemDefault()).toInstant())) } ?: "",
305         onValueChange = { },
306         label = { Text(text = "Select Date") },
307         readOnly = true,
308         trailingIcon = {
309             IconButton(onClick = onClick) {
310                 Icon(imageVector = Icons.Default.
311                     DateRange, contentDescription = "Select Date")

```

```
309         }
310     }
311 )
312 }
313
314 @Composable
315 fun details(name:String,age:String,group:String){
316     Column(modifier = Modifier.fillMaxSize(0.5f),
317         verticalArrangement = Arrangement.
318         SpaceEvenly) {
319         Text(text = name)
320         Text(text = age)
321         Text(text =group)
322     }
```