Control Flow Test for Update Goals Activity ()

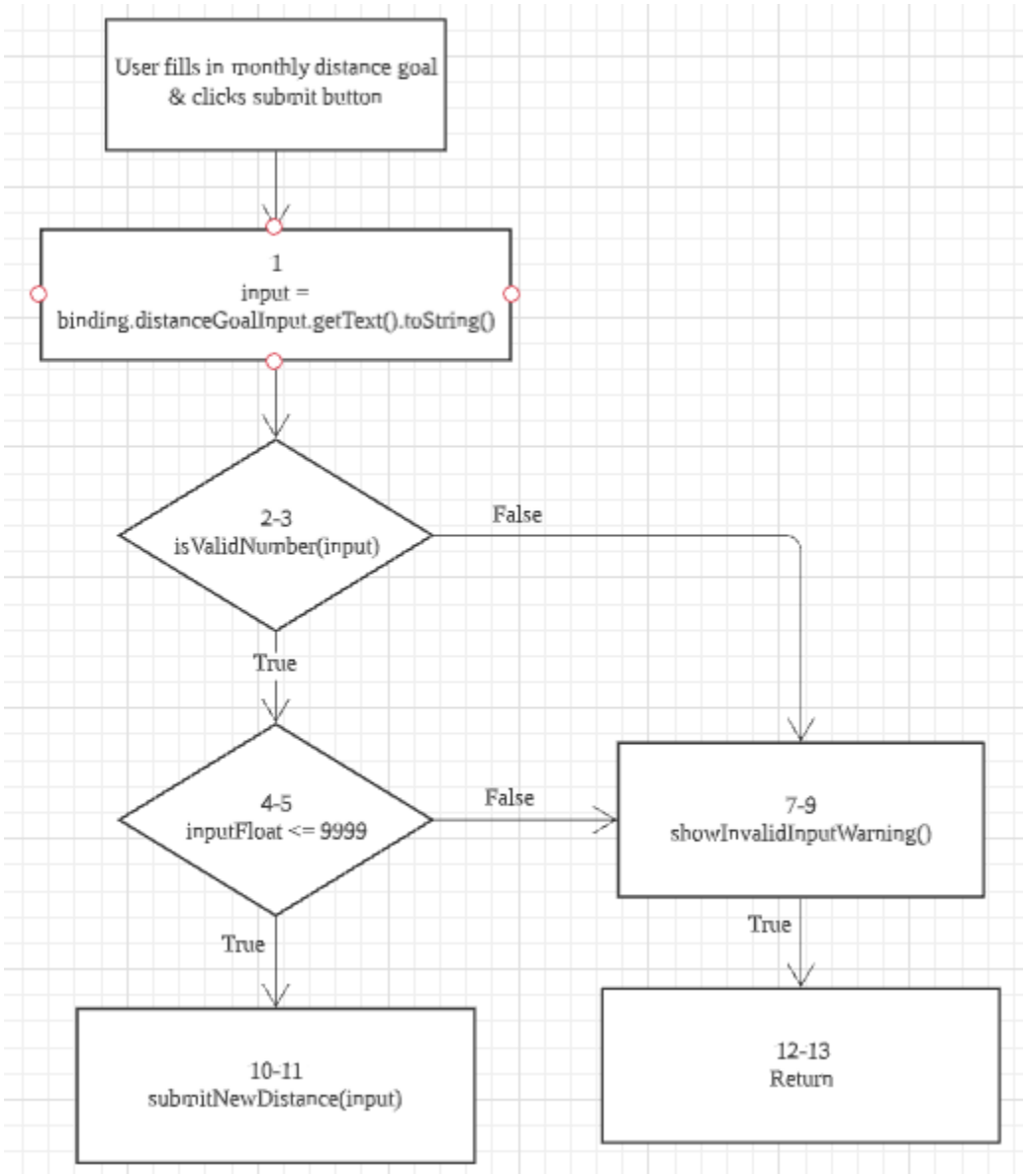
Update Distance Goal

Code

```
        public void onClick(View view) {
1                   String input = binding.distanceGoalInput.getText().toString();
2                   boolean valid = isValidDistance(input);
8                   if (!valid) {
9                   showInvalidInputWarning();
10                  } else {
11                  submitNewDistance(input);
12              }
13              return
                }
        }

        private boolean isValidDistance(String input) {
3           if (isValidNumber(input)) {
4               float inputFloat = Float.parseFloat(input);
5               if (inputFloat <= 9999) return true;
6           }
7           return false;
        }
```

Control Flow Graph



*Kindly refer to the separate pdf document “Control Flow Graph 1” attached for a clearer view

Cyclomatic Complexity

Taking Cyclomatic complexity: |decision points| + 1 = 2 + 1 = **3**

Test Cases

I. User fills in valid number and distance
II. User fills in valid number and invalid distance

III. User fills in invalid number and distance

Basis Paths

I. Path 1 (Baseline): 1, 2-3, 4-5, 10-11
II. Path 2: 1, 2-3, 4-5, 7-9, 12-13 (User fills in valid number and invalid distance)
III. Path 3: 1, 2-3, 7-9, 12-13 (User fills in invalid number and distance)

Test Results

Path	**Result**
1	Pass
2	Pass
3	Pass

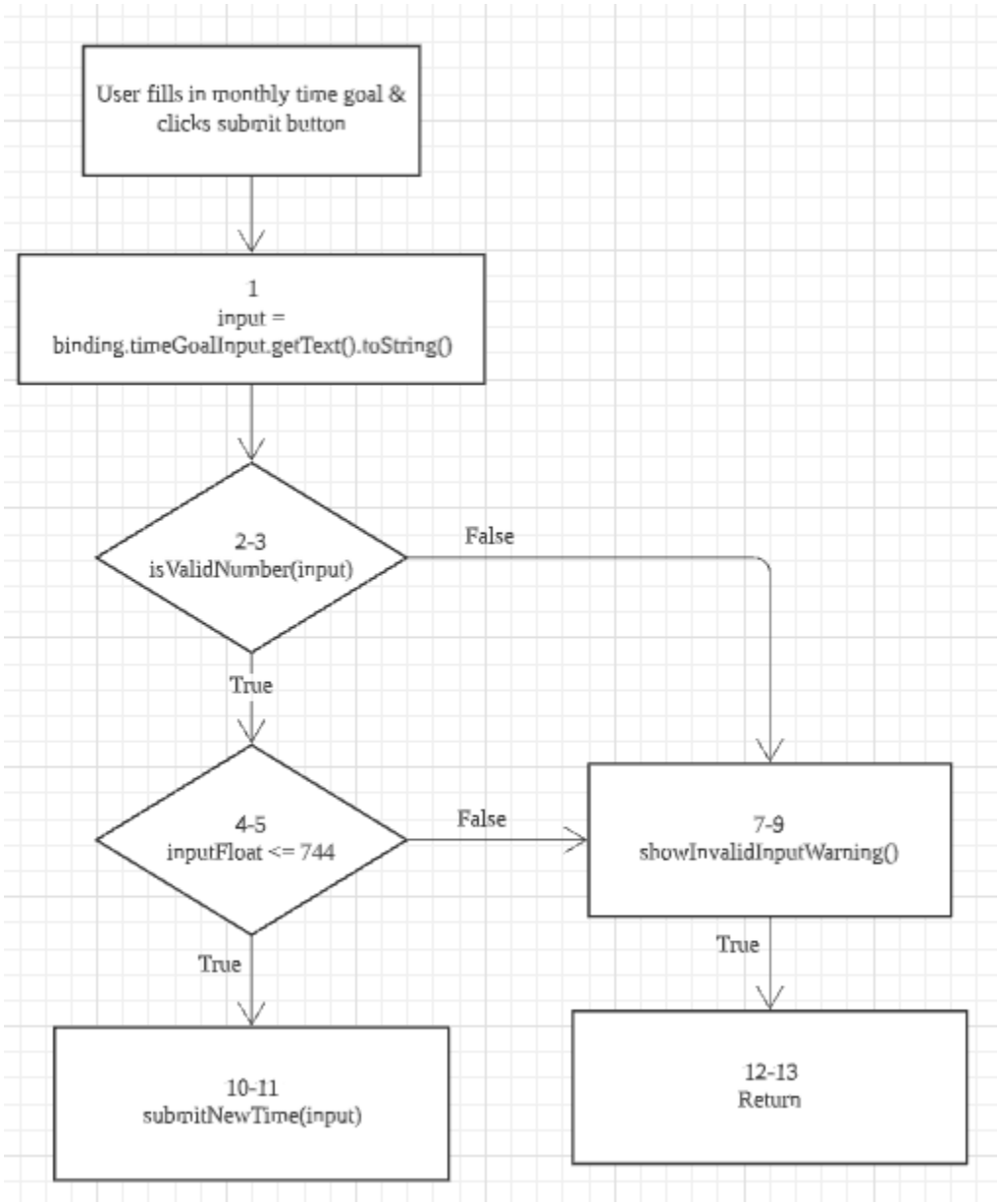
Update Time Goal

Code

```
        public void onClick(View view) {
1                       String input = binding.timeGoalInput.getText().toString();
2                       boolean valid = isValidDistance(input);
8                       if (!valid) {
9                       showInvalidInputWarning();
10                      } else {
11                      submitNewTime(input);
12              }
13              return
        }

        private boolean isValidTime(String input) {
3               if (isValidNumber(input)) {
4                   float inputFloat = Float.parseFloat(input);
5                   if (inputFloat <= 744) return true;
6               }
7               return false;
        }
```

Control Flow



*Kindly refer to the separate pdf document “Control Flow Graph 2” attached for a clearer view

Cyclomatic Complexity

Taking Cyclomatic complexity: |decision points| + 1 = 2 + 1 = **3**

Test Cases

IV. User fills in valid number and time

V. User fills in valid number and invalid time
VI. User fills in invalid number and time

Basis Paths

IV. Path 1 (Baseline): 1, 2-3, 4-5, 10-11
V. Path 2: 1, 2-3, 4-5, 7-9, 12-13 (User fills in valid number and invalid time)
VI. Path 3: 1, 2-3, 7-9, 12-13 (User fills in invalid number and time)

Test Results

Path	**Result**
1	Pass
2	Pass
3	Pass

Control Flow Test for Email Login Activity ()

Code

```
private void userLogin() {
1        String email = mBinding.email.getText().toString().trim();
2       String password = mBinding.password.getText().toString().trim();
3        if (email.isEmpty()) {
4            mBinding.email.setError("Email required");
5           mBinding.email.requestFocus();
6            return; }
7       if (!Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
8            mBinding.email.setError("Please provide valid email");
9            mBinding.email.requestFocus();
10             return; }
11       if (password.isEmpty()) {
12             mBinding.password.setError("Password required");
13             mBinding.password.requestFocus();
14             return; }
15        if (password.length() < 6) {
16             mBinding.password.setError("Minimum password length is 6 characters");
17             mBinding.password.requestFocus();
18             return;} }

19 mAuth.signInWithEmailAndPassword(email,
password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
```

```
20          @Override
21          public void onComplete(@NonNull Task<AuthResult> task) {
22              if (task.isSuccessful()) {
23                  FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser(); }
24              else {
25                  Toast.makeText(EmailLoginActivity.this, "Failed to login. Check
credentials", Toast.LENGTH_LONG).show(); }
```

Control Flow

*Kindly refer to the separate pdf document “Control Flow Graph 3” attached for a clearer view

Cyclomatic Complexity

Taking Cyclomatic complexity: |decision points| + 1 = 5 + 1 = **6**

Test Cases

I. User fills in correct email address and password
II. User fills in correct email address but incorrect password
III. User fills in correct email address but empty password

Basis Paths

I. Path 1 (Baseline): 1-2,3,7,11,15,19,22,23
II. Path 2: 1-2,3,7,11,15,19,22,21
III. Path 3: 1-2,3,7,11,12

Test Results

Path	**Result**
1	Pass
2	Pass
3	Pass