**🟢 STEP 1: Create a New Project**

* Open Xojo
* Choose: iOS → Blank
* Name the project: TEGRotemApp
* Save as .xojo\_binary\_project or default format

**🟢 STEP 2: Add Classes**

**TEGResult**

1. Add → Class
2. Name: TEGResult
3. Add properties:

Public CKHR As Double

Public LY30 As Double

**ROTEMResult**

1. Add → Class
2. Name: ROTEMResult
3. Add properties:

Public ExtemA5 As Double

Public FibtemA5 As Double

Public ExtemCT As Double

Public FibtemML As Double

**🟢 STEP 3: Modify the App**

**Class**

1. In App → Add properties:

Public Property Teg As TEGResult

Public Property Rotem As ROTEMResult

Public Property Mode As String

1. In App.Opening event:

Sub Opening()

  // Example TEG test values

  Var t As New TEGResult

  t.CKHR = 13.2

  t.LY30 = 10.5

  Self.Teg = t

  Self.Mode = "TEG"

End Sub

**🟢 STEP 4: Add a New Screen:**

**AlgorithmStepScreen**

1. Add → iOSScreen → Name: AlgorithmStepScreen
2. Drag in:
   * iOSScrollView → Name: StepContainer
   * iOSButton → Name: ButtonNextStep → Caption: “Next Step”
   * iOSButton → Name: ButtonRestart → Caption: “Restart” → Initially hidden

**🟢 STEP 5: Add a Container:**

**StepRowContainer**

1. Add → iOSContainerControl
2. Name: StepRowContainer
3. Add:
   * iOSImageView → Name: IconView
   * iOSLabel → Name: StepLabel → Multiline enabled
4. Add this method:

Public Sub Setup(iconText As String, stepText As String)

Select Case iconText

Case "⚠️"

IconView.Image = iOSImage.SystemImage("exclamationmark.triangle")

Case "✅"

IconView.Image = iOSImage.SystemImage("checkmark.circle")

Case "⏳"

IconView.Image = iOSImage.SystemImage("hourglass")

End Select

StepLabel.Text = stepText

End Sub

**🟢 STEP 6: Add Logic to**

**AlgorithmStepScreen**

**a. Add property:**

Private CurrentStep As Integer = 0

**b. Add this method:**

Public Sub AddStepToUI(icon As String, text As String)

Var row As New StepRowContainer

row.Setup(icon, text)

StepContainer.AddControl(row)

End Sub

**c. In ButtonNextStep.Pressed:**

**Sub ButtonNextStep\_Pressed()**

**Var icon As String**

**Var output As String**

**If App.Mode = "TEG" Then**

**Var t As TEGResult = App.Teg**

**Select Case CurrentStep**

**Case 0**

**output = "CKH-R = " + t.CKHR.ToString + " min"**

**If t.CKHR > 11 Then**

**icon = "⚠️"**

**output = output + " → Abnormal → Give FFP or PCC"**

**Else**

**icon = "✅"**

**output = output + " → Normal"**

**End If**

**Case 1**

**output = "LY30 = " + t.LY30.ToString + "%"**

**If t.LY30 > 7.5 Then**

**icon = "⚠️"**

**output = output + " → Abnormal → Give Tranexamic Acid"**

**Else**

**icon = "✅"**

**output = output + " → Normal"**

**End If**

**Case 2**

**icon = "⏳"**

**output = "If bleeding continues, wait 10 minutes and repeat TEG."**

**ButtonNextStep.Enabled = False**

**ButtonRestart.Visible = True**

**End Select**

**AddStepToUI(icon, output)**

**CurrentStep = CurrentStep + 1**

**End If**

**End Sub**

**d. In ButtonRestart.Pressed:**

**Sub ButtonRestart\_Pressed()**

**StepContainer.RemoveAll**

**CurrentStep = 0**

**ButtonNextStep.Enabled = True**

**ButtonRestart.Visible = False**

**End Sub**

**🟢 STEP 7: Set the Initial Screen**

* In the Navigator, set AlgorithmStepScreen as the default iOS screen under App → iOSLayout.

**✅ Done! Now Run It**

* Press ▶️ to build and simulate
* You’ll see steps appear with icons as you press **Next Step**
* On final step, it shows the “wait 10 minutes” message
* **✅ Step 1: Add ROTEM Logic to**
* **ButtonNextStep\_Pressed**
* Expand the existing logic in AlgorithmStepScreen:

**Var r As ROTEMResult**

Then insert this block after the If Mode = "TEG" section:

ElseIf Mode = "ROTEM" Then

r = App.Rotem

Select Case CurrentStep

Case 0

output = "EXTEM A5 = " + r.ExtemA5.ToString + " mm"

If r.ExtemA5 < 35 Then

icon = "⚠️"

output = output + " → Low clot strength → Consider platelets or fibrinogen"

Else

icon = "✅"

output = output + " → Normal clot strength"

End If

Case 1

output = "FIBTEM A5 = " + r.FibtemA5.ToString + " mm"

If r.FibtemA5 < 10 Then

icon = "⚠️"

output = output + " → Low fibrinogen → Consider cryoprecipitate or fibrinogen concentrate"

Else

icon = "✅"

output = output + " → Normal fibrinogen"

End If

Case 2

output = "EXTEM CT = " + r.ExtemCT.ToString + " sec"

If r.ExtemCT > 80 Then

icon = "⚠️"

output = output + " → Prolonged clotting time → Give FFP or PCC"

Else

icon = "✅"

output = output + " → Normal clotting time"

End If

Case 3

output = "FIBTEM ML = " + r.FibtemML.ToString + " %"

If r.FibtemML > 15 Then

icon = "⚠️"

output = output + " → Hyperfibrinolysis → Give tranexamic acid"

Else

icon = "✅"

output = output + " → No hyperfibrinolysis"

End If

Case 4

icon = "⏳"

output = "If bleeding continues, wait 10 minutes and repeat ROTEM."

ButtonNextStep.Enabled = False

ButtonRestart.Visible = True

End Select

End If

**✅ Step 2: Increment Step Logic**

At the bottom of the ButtonNextStep\_Pressed event:

AddStepToUI(icon, output)

CurrentStep = CurrentStep + 1

**✅ Step 3: Provide Example ROTEM Test Input**

In App.Opening, for testing:

Var r As New ROTEMResult

r.ExtemA5 = 32

r.FibtemA5 = 7.5

r.ExtemCT = 95

r.FibtemML = 20

Self.Rotem = r

Self.Mode = "ROTEM"

💡 Later, you can replace this with values from user input in a separate screen.

**✅ Step-by-Step ROTEM Algorithm Logic (continued)**

Extend ButtonNextStep\_Pressed → ROTEM section like so:

ElseIf Mode = "ROTEM" Then

r = App.Rotem

Select Case CurrentStep

Case 0

output = "EXTEM A5 = " + r.ExtemA5.ToString + " mm"

If r.ExtemA5 < 35 Then

icon = "⚠️"

output = output + " → Low clot strength → Consider platelets or fibrinogen"

Else

icon = "✅"

output = output + " → Normal clot strength"

End If

Case 1

output = "FIBTEM A5 = " + r.FibtemA5.ToString + " mm"

If r.FibtemA5 < 10 Then

icon = "⚠️"

output = output + " → Low fibrinogen → Consider cryoprecipitate or fibrinogen concentrate"

Else

icon = "✅"

output = output + " → Normal fibrinogen"

End If

Case 2

output = "EXTEM CT = " + r.ExtemCT.ToString + " sec"

If r.ExtemCT > 80 Then

icon = "⚠️"

output = output + " → Prolonged clotting time → Give FFP or PCC"

Else

icon = "✅"

output = output + " → Normal clotting time"

End If

Case 3

output = "FIBTEM ML = " + r.FibtemML.ToString + " %"

If r.FibtemML > 15 Then

icon = "⚠️"

output = output + " → Excessive fibrinolysis → Consider tranexamic acid"

Else

icon = "✅"

output = output + " → No evidence of fibrinolysis"

End If

Case 4

icon = "⏳"

output = "If bleeding continues, wait 10 minutes and repeat ROTEM. Restart algorithm with new values."

ButtonNextStep.Enabled = False

ButtonRestart.Visible = True

End Select

End If

**🧩 Shared Logic After Step Evaluation**

At the end of the method, outside both If Mode = blocks:

If output <> "" Then

AddStepToUI(icon, output)

CurrentStep = CurrentStep + 1

End If

**🔁 Restart Button Logic**

Sub ButtonRestart\_Pressed()

StepContainer.RemoveAll

CurrentStep = 0

ButtonNextStep.Enabled = True

ButtonRestart.Visible = False

End Sub

**✅ Test ROTEM Inputs**

In App.Opening, you can test the logic like this (temporarily):

Var r As New ROTEMResult

r.ExtemA5 = 28

r.FibtemA5 = 7

r.ExtemCT = 95

r.FibtemML = 20

App.Rotem = r

App.Mode = "ROTEM"

Then run the app and go through the ROTEM steps interactively.

**✅ 1. Complete the**

**StepRowContainer**

**With Icons and TEG Compatibility**

**🔧 Inside**

**StepRowContainer**

Assuming you’ve already created the container with:

* IconView (iOSImageView)
* StepLabel (iOSLabel)

Update the method:

Public Sub Setup(iconText As String, stepText As String)

StepLabel.Text = stepText

Select Case iconText

Case "⚠️"

// Set warning icon

IconView.Image = iOSImage.SystemImage("exclamationmark.triangle.fill")

IconView.TintColor = Color.Red

Case "✅"

IconView.Image = iOSImage.SystemImage("checkmark.circle.fill")

IconView.TintColor = Color.Green

Case "⏳"

IconView.Image = iOSImage.SystemImage("hourglass")

IconView.TintColor = Color.Orange

Else

IconView.Image = Nil

End Select

End Sub

**✅ 2. Add a “Start Over” Selection Screen**

**📱 Screen:**

**StartScreen**

Add a new iOSScreen called StartScreen. Layout:

* Title label: "Select Test Type"
* Two iOSButtons:
  + ButtonTEG → Title: "Use TEG"
  + ButtonROTEM → Title: "Use ROTEM"

**🎯 Code for Buttons**

In ButtonTEG.Pressed:

Sub ButtonTEG\_Pressed()

App.Mode = "TEG"

// Example values

Var t As New TEGResult

t.CKHR = 13.1

t.LY30 = 8.0

App.Teg = t

AlgorithmStepScreen.CurrentStep = 0

AlgorithmStepScreen.StepContainer.Content = Nil

AlgorithmStepScreen.Show

End Sub

In ButtonROTEM.Pressed:

Sub ButtonROTEM\_Pressed()

App.Mode = "ROTEM"

Var r As New ROTEMResult

r.ExtemA5 = 34

r.FibtemA5 = 9

r.ExtemCT = 95

r.FibtemML = 20

App.Rotem = r

AlgorithmStepScreen.CurrentStep = 0

AlgorithmStepScreen.StepContainer.Content = Nil

AlgorithmStepScreen.Show

End Sub

**Final Touch: Restart Button on**

**AlgorithmStepScreen**

In ButtonRestart\_Pressed:

Sub ButtonRestart\_Pressed()

App.Teg = Nil

App.Rotem = Nil

App.Mode = ""

StartScreen.Show

End Sub

┌────────────────────────────────────────┐

│ [StartScreen] │

│ │

│ Select Test Type │

│ │

│ [ Use TEG ] [ Use ROTEM ] │

└────────────────────────────────────────┘

┌────────────────────────────────────────┐

│ [AlgorithmStepScreen] │

│ │

│ [StepContainer - ScrollView] │

│ ┌───────────────────────────────┐ │

│ │ ✅ EXTEM A5 = 42 mm → Normal │ │

│ ├───────────────────────────────┤ │

│ │ ⚠️ FIBTEM A5 = 8 mm → Low Fg │ │

│ ├───────────────────────────────┤ │

│ │ ⏳ Wait 10 min, reassess │ │

│ └───────────────────────────────┘ │

│ │

│ [Next Step] [Restart] │

└────────────────────────────────────────┘

Each row in the scroll area uses StepRowContainer with:

* IconView (system image)
* StepLabel (dynamic guidance)

**✅ Final Xojo Project: Asset Checklist**

**📁 Project File**

* Save as: TEGRotemApp.xojo\_binary\_project
* **📦 App Properties (in App class)**

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Mode | String | "TEG" or "ROTEM" |
| Teg | TEGResult | TEG test result |
| Rotem | ROTEMResult | ROTEM test result |

**📦 Classes**

| **Class Name** | **Properties** |
| --- | --- |
| TEGResult | CKHR As Double, LY30 As Double |
| ROTEMResult | ExtemA5, FibtemA5, ExtemCT, FibtemML (All Double) |

**📱 Screens**

| **Screen Name** | **Description** |
| --- | --- |
| StartScreen | Buttons to choose TEG or ROTEM |
| AlgorithmStepScreen | Scrollable algorithm output |

### **🧩 Controls**

#### StartScreen

* LabelTitle
* ButtonTEG
* ButtonROTEM

#### AlgorithmStepScreen

* StepContainer (iOSScrollView)
* ButtonNextStep
* ButtonRestart

#### StepRowContainer

* IconView (iOSImageView)
* StepLabel (iOSLabel)
* Method: Setup(iconText As String, stepText As String)

### 🛠 Methods

| **Class/Screen** | **Method** |
| --- | --- |
| StepRowContainer | Setup(iconText, stepText) |
| AlgorithmStepScreen | AddStepToUI(icon, message) |
| ButtonNextStep\_Pressed | Core logic for TEG and ROTEM steps |
| ButtonRestart\_Pressed | Resets state and returns to StartScreen |

Here’s the **complete final part** of the checklist you need to finish assembling and testing your Xojo-based iOS app in under 10 minutes:

## ✅ Controls (continued)

### 📱

### AlgorithmStepScreen

| **Control** | **Type** | **Notes** |
| --- | --- | --- |
| StepContainer | iOSScrollableArea | Contains StepRowContainers dynamically |
| ButtonNextStep | iOSButton | Advances to next algorithm step |
| ButtonRestart | iOSButton | Restarts to StartScreen (initially hidden) |

### 🧩

### StepRowContainer UI

| **Control** | **Type** | **Purpose** |
| --- | --- | --- |
| IconView | iOSImageView | Displays ✅ / ⚠️ / ⏳ |
| StepLabel | iOSLabel | Displays algorithm step guidance |

## 🧠 Key Logic Methods

| **Location** | **Method** | **Purpose** |
| --- | --- | --- |
| StepRowContainer | Setup(iconText, stepText) | Loads icon + step text |
| AlgorithmStepScreen | AddStepToUI(icon, text) | Adds a step to the scroll area |
| AlgorithmStepScreen | ButtonNextStep\_Pressed | Executes algorithm logic (TEG/ROTEM) |
| AlgorithmStepScreen | ButtonRestart\_Pressed | Resets state + returns to start |
| StartScreen | ButtonTEG\_Pressed | Sets TEG mode + dummy data |
| StartScreen | ButtonROTEM\_Pressed | Sets ROTEM mode + dummy data |

## 💾 Suggested File Save Format

When saving:

* **Name**: TEGRotemApp.xojo\_binary\_project
* **Export**: Use “File > Export > Xojo Binary Project”

Once saved, zip the project folder:

* Include all .xojo\_binary\_project, .xojo\_code, .xojo\_image, and Builds - iOS (optional).

## 📦 Zip Template Contents (suggested)

/TEGRotemApp/

├── TEGResult.xojo\_code

├── ROTEMResult.xojo\_code

├── StepRowContainer.xojo\_code

├── AlgorithmStepScreen.xojo\_code

├── StartScreen.xojo\_code

├── App.xojo\_code

├── TEGRotemApp.xojo\_binary\_project

└── README.txt (optional instructions)