Report on homework 2
« Fun with layouts »

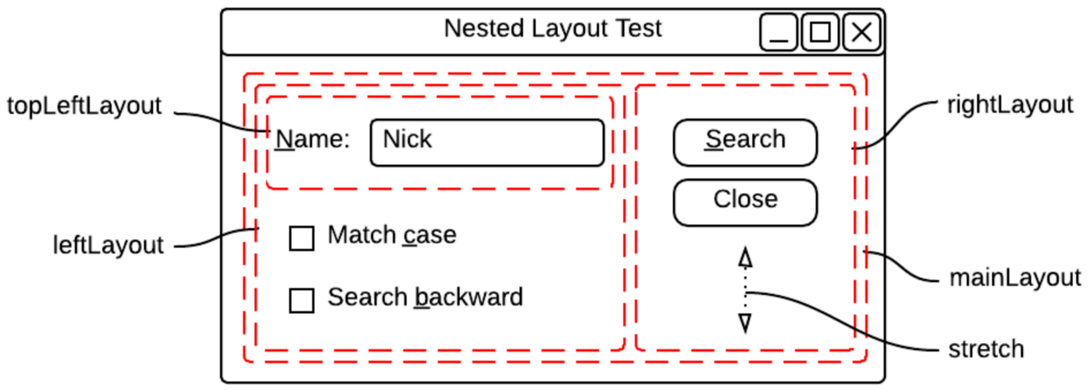
Tarek Kamel
Mohamed Abdallaoui

Table of contents:

I - Nested Layouts :

1- introduction .

This exercise aims to learn and analyse the construction of a form and code it. In order to achieve this we had to implement a Form that looks like this :



Nested Layout.

Let's analyse this window :

It comprises 4 layouts :

- MainLayout that gathers 3 other layouts in the form of a QHBoxLayout

-LeftLayout (QVBoxLayout) that comprises TopLeftLayout and two widgets with a empty checkable square QCheckBox,

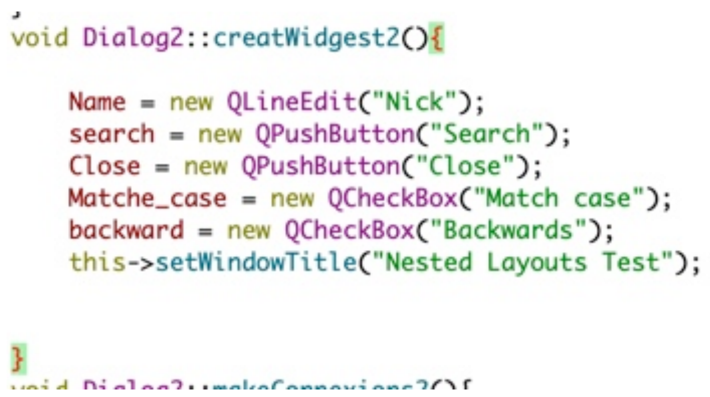
-TopLeftLayout(QHBoxLayout) that comprises QLineEdit(Nick) and a QLabel(Name:).

-RightLayout(QVBoxLayout) that includes two widgets of type QPuchButton.

2- code explanation:

We have created a class named Dialog2 that inherits form QWidget, in which we implemented the following methods:

- CreatWidget2() : which basically creates all the widgets needed for this window.

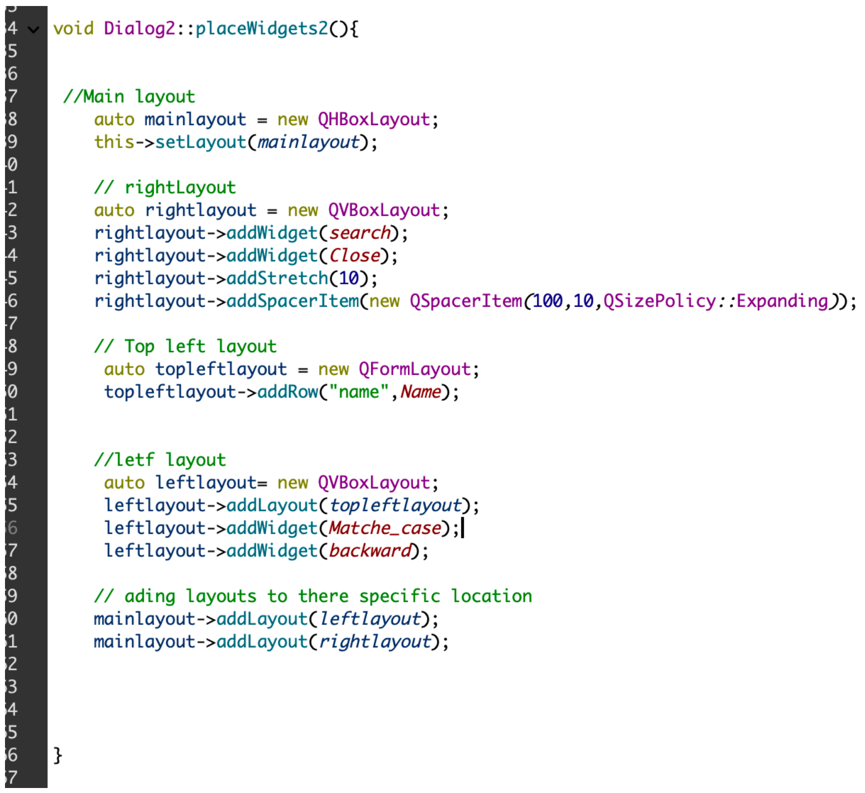


```
void Dialog2::creatWidgest2(){

     Name = new QLineEdit("Nick");
     search = new QPushButton("Search");
     Close = new QPushButton("Close");
     Matche_case = new QCheckBox("Match case");
     backward = new QCheckBox("Backwards");
     this->setWindowTitle("Nested Layouts Test");


}
```

- PlaceWidget2() : which places the different widgets created



```
void Dialog2::placeWidgets2(){


 //Main layout
    auto mainlayout = new QHBoxLayout;
    this->setLayout(mainlayout);

    // rightLayout
    auto rightlayout = new QVBoxLayout;
    rightlayout->addWidget(search);
    rightlayout->addWidget(Close);
    rightlayout->addStretch(10);
    rightlayout->addSpacerItem(new QSpacerItem(100,10,QSizePolicy::Expanding));

    // Top left layout
     auto topleftlayout = new QFormLayout;
     topleftlayout->addRow("name",Name);


    //letf layout
     auto leftlayout= new QVBoxLayout;
     leftlayout->addLayout(topleftlayout);
     leftlayout->addWidget(Matche_case);
     leftlayout->addWidget(backward);

    // ading layouts to there specific location
    mainlayout->addLayout(leftlayout);
    mainlayout->addLayout(rightlayout);




}
```

We've implement above two new predefine methods that inherits from QBoxLayout that are:

> addStretch :

> addSpacerItem :

That add a vertical space stretch

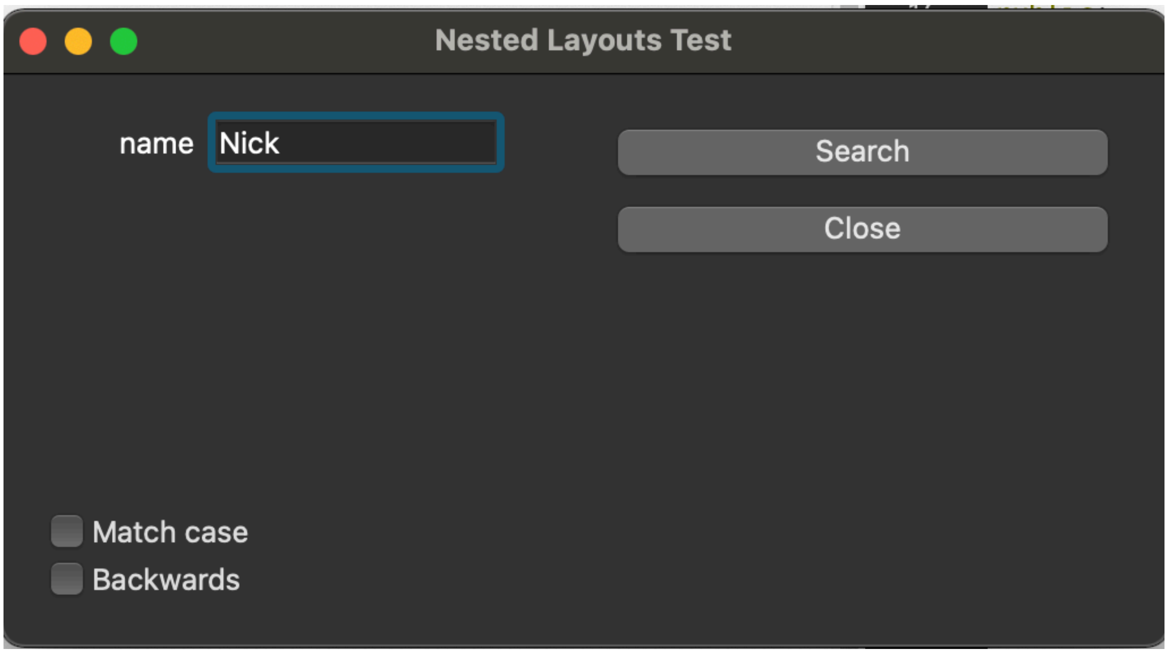
- makeConnections2(): which connects the the QPushButton « close » to a slot that closes the window

```
}
void Dialog2::makeConnexions2(){
    connect(Close, &QPushButton::clicked, qApp ,& QApplication::exit);

}
```

3- output results.

Below our final result after compiling the program



II - Bug report form

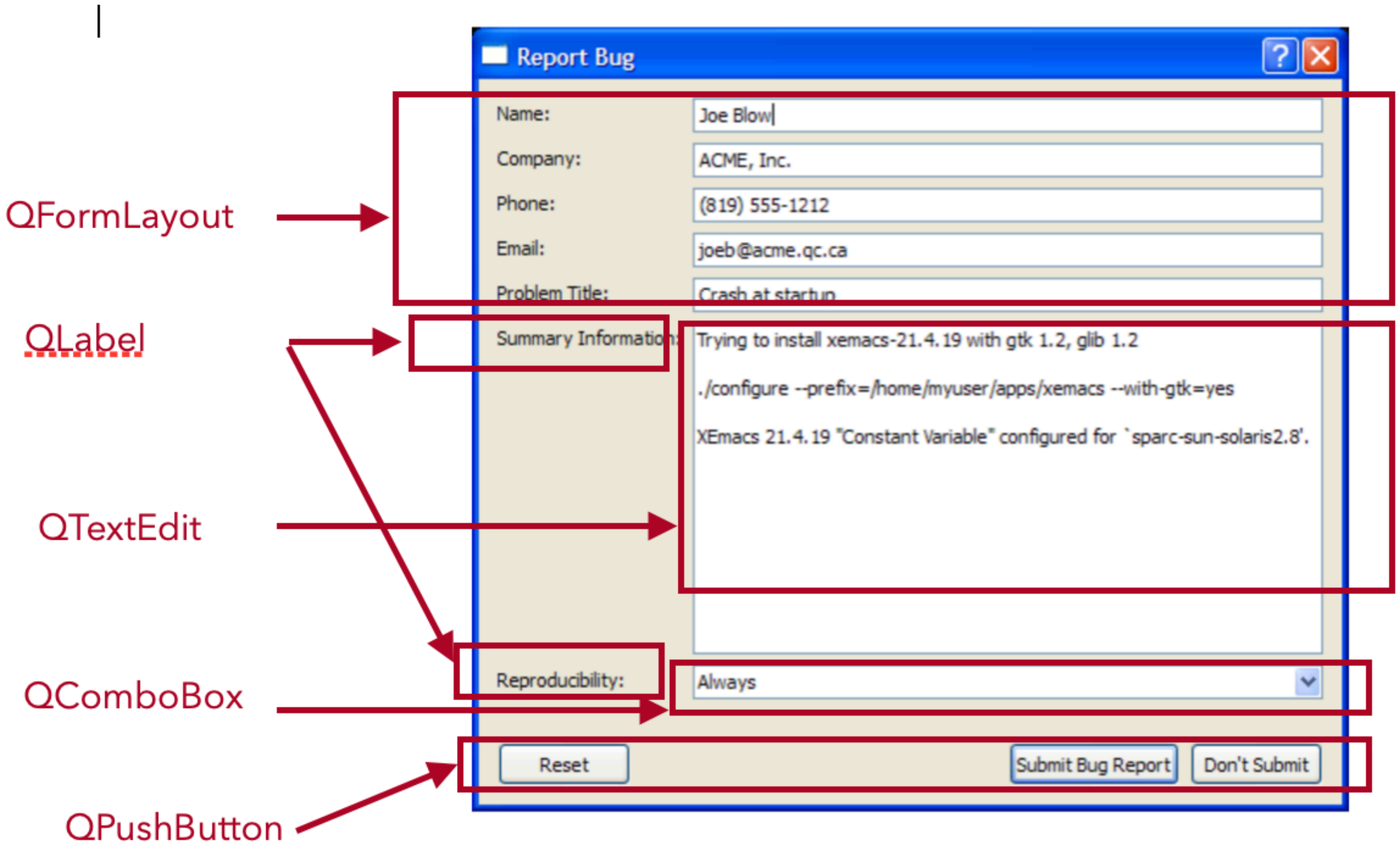
1- introduction .

The goal of this exercise is to work on a bug reporting window using different layouts

The final output should look like this :

- this is how we proceeded for the labels :

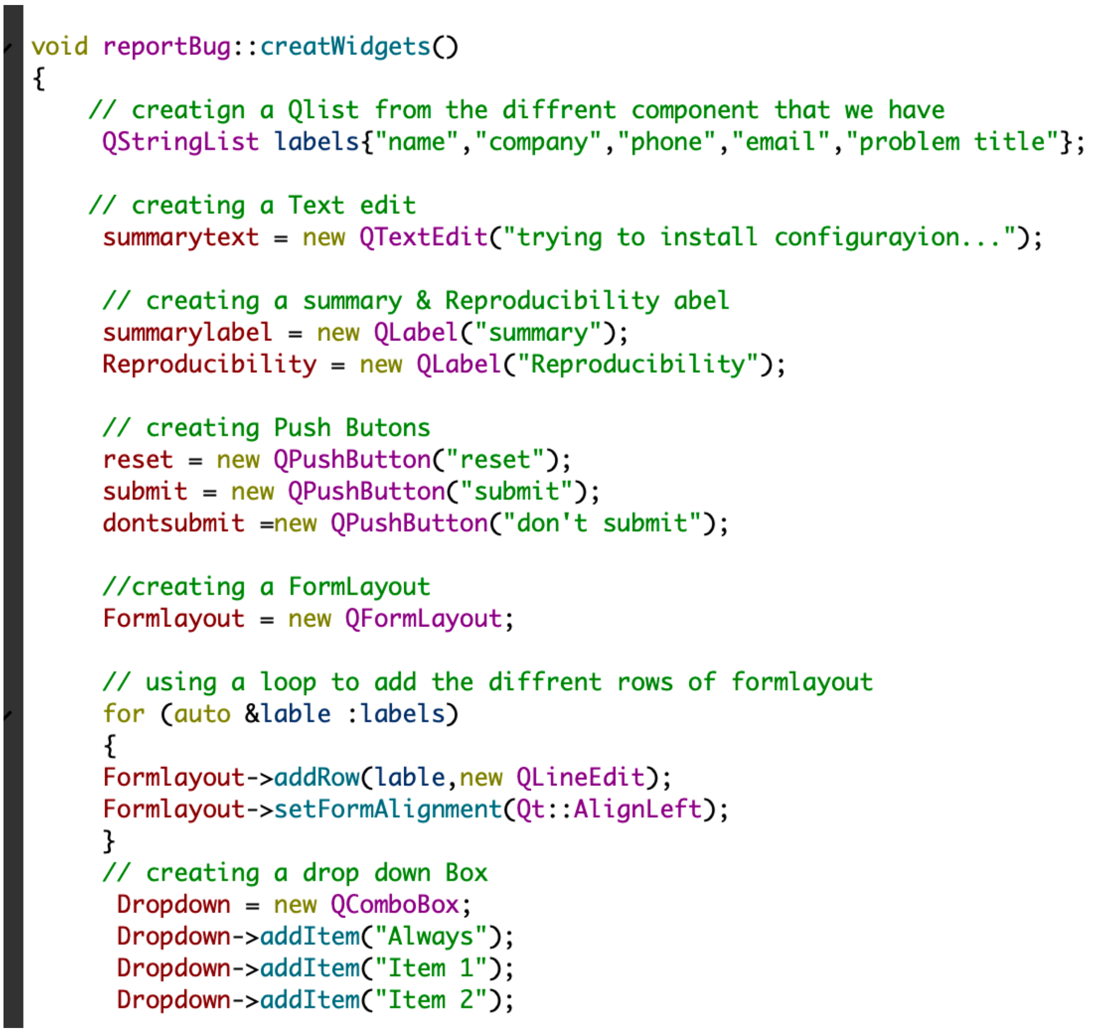
|



Name:	Joe Blow
Company:	ACME, Inc.
Phone:	(819) 555-1212
Email:	joeb@acme.qc.ca
Problem Title:	Crash at startup
Summary Information:	Trying to install xemacs-21.4.19 with gtk 1.2, glib 1.2 ./configure --prefix=/home/myuser/apps/xemacs --with-gtk=yes XEmacs 21.4.19 "Constant Variable" configured for `sparc-sun-solaris2.8'.
Reproducibility:	Always

2- code explanation.

- Starting first with our method creatWidget that's basically is implemented to create the different components of our window



```
void reportBug::creatWidgets()
{
    // creatign a Qlist from the diffrent component that we have
     QStringList labels{"name","company","phone","email","problem title"};

    // creating a Text edit
     summarytext = new QTextEdit("trying to install configurayion...");

     // creating a summary & Reproducibility abel
     summarylabel = new QLabel("summary");
     Reproducibility = new QLabel("Reproducibility");

     // creating Push Butons
     reset = new QPushButton("reset");
     submit = new QPushButton("submit");
     dontsubmit =new QPushButton("don't submit");

     //creating a FormLayout
     Formlayout = new QFormLayout;

     // using a loop to add the diffrent rows of formlayout
     for (auto &lable :labels)
     {
     Formlayout->addRow(lable,new QLineEdit);
     Formlayout->setFormAlignment(Qt::AlignLeft);
     }
     // creating a drop down Box
      Dropdown = new QComboBox;
      Dropdown->addItem("Always");
      Dropdown->addItem("Item 1");
      Dropdown->addItem("Item 2");
```

- Moving on next to out second method placeWidget in which we are going to place the different labels and layouts that we have created previously

```
void reportBug::placeWidgets()
{
    // place main layout
   auto mainlayout = new QVBoxLayout;
   this->setLayout(mainlayout);

//   placing left layout
   mainlayout->addLayout(Formlayout);
   //place sumary text edit in left layout :
   auto summarylayout= new QHBoxLayout;
   mainlayout->addLayout(summarylayout);

    //placing summary
   summarylayout->addWidget(summarylabel);
   summarylayout->addWidget(summarytext);

   //place Reproducibility
    auto myRepo = new QHBoxLayout;
    mainlayout->addLayout(myRepo);


    myRepo->addWidget(Reproducibility);
    myRepo->addWidget(Dropdown);

    auto bottom = new QHBoxLayout;
    auto Botom1 = new QHBoxLayout;
    auto Botom2 = new QHBoxLayout;

    // ading Botom2 and Botom1 to botom
    bottom->addLayout(Botom2);
    bottom->addLayout(Botom1);

    //ading bottom to main layout
    mainlayout->addLayout(bottom);

    //placing botoms widgets
     Botom2->setSpacing(200);
     Botom2 ->addWidget(reset);
     Botom2 ->addWidget(submit);
     Botom1 ->addWidget(dontsubmit);


}
```

- this is how we proceeded to place the different layout

▬ ▬ ▬ ▬ Mainlayout «main layout »

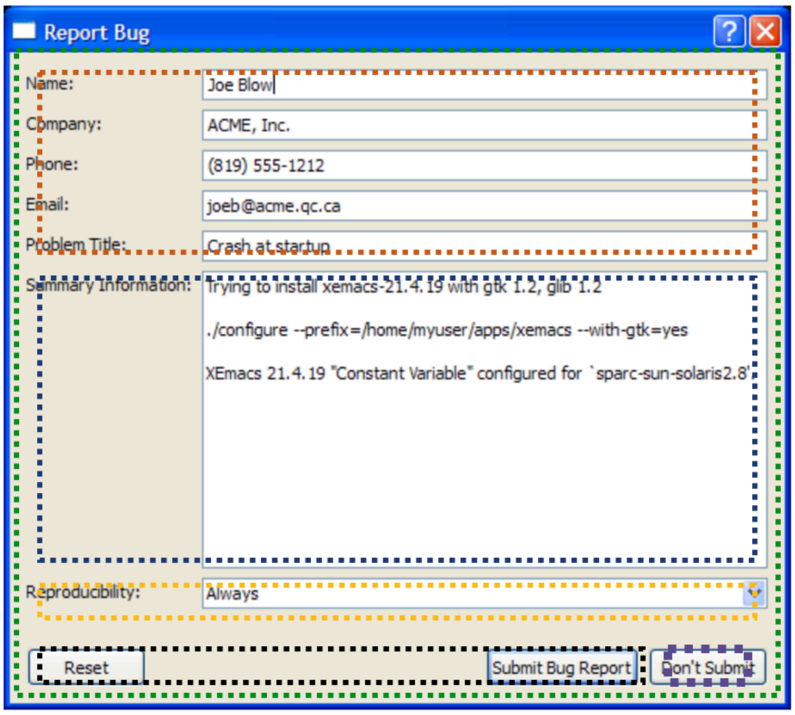
▬ ▬ ▬ QFormLayout

▬ ▬ ▬ ▬ QHBoxLayout « summarylayout »

▬ ▬ ▬ ▬ QHBoxLayout « myRepo »

▬ ▬ ▬ QHBoxLayout « botom1 »

▬ ▬ ▬ QHBoxLayout « botom2 »



Report Bug

Name: Joe Blow

Company: ACME, Inc.

Phone: (819) 555-1212

Email: joeb@acme.qc.ca

Problem Title: Crash at startup

Summary Information: Trying to install xemacs-21.4.19 with gtk 1.2, glib 1.2

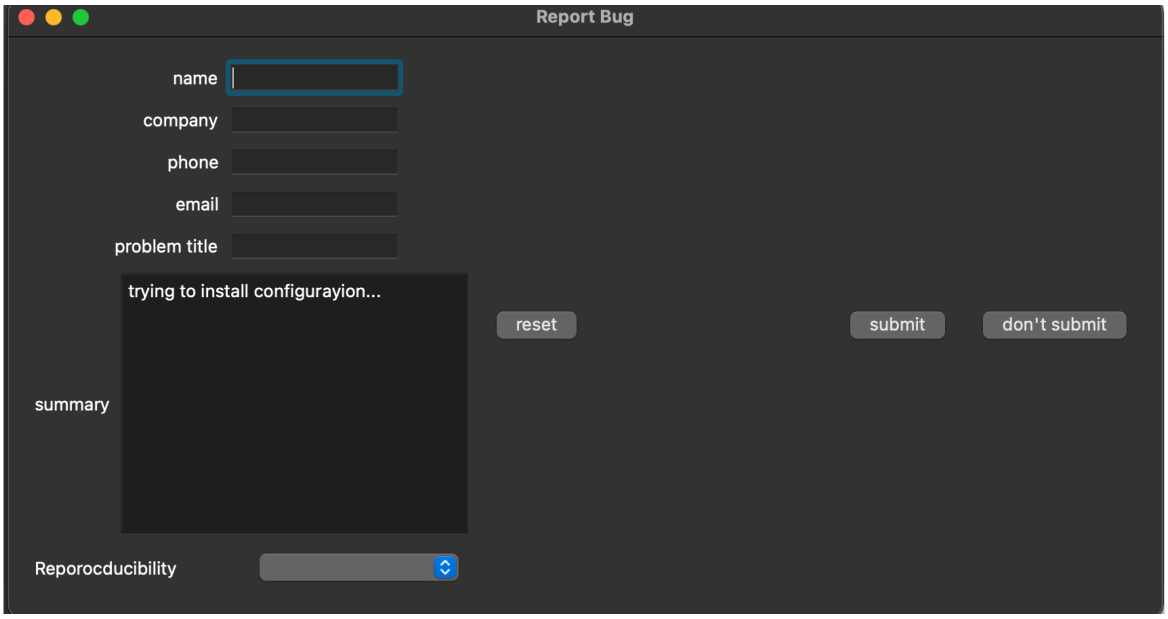
./configure --prefix=/home/myuser/apps/xemacs --with-gtk=yes

XEmacs 21.4.19 "Constant Variable" configured for `sparc-sun-solaris2.8'

Reproducibility: Always

3- difficulties.

- We have encountered during this project many difficulties among was the placement the different layout (result below)



name

company

phone

email

problem title

summary: trying to install configurayion...

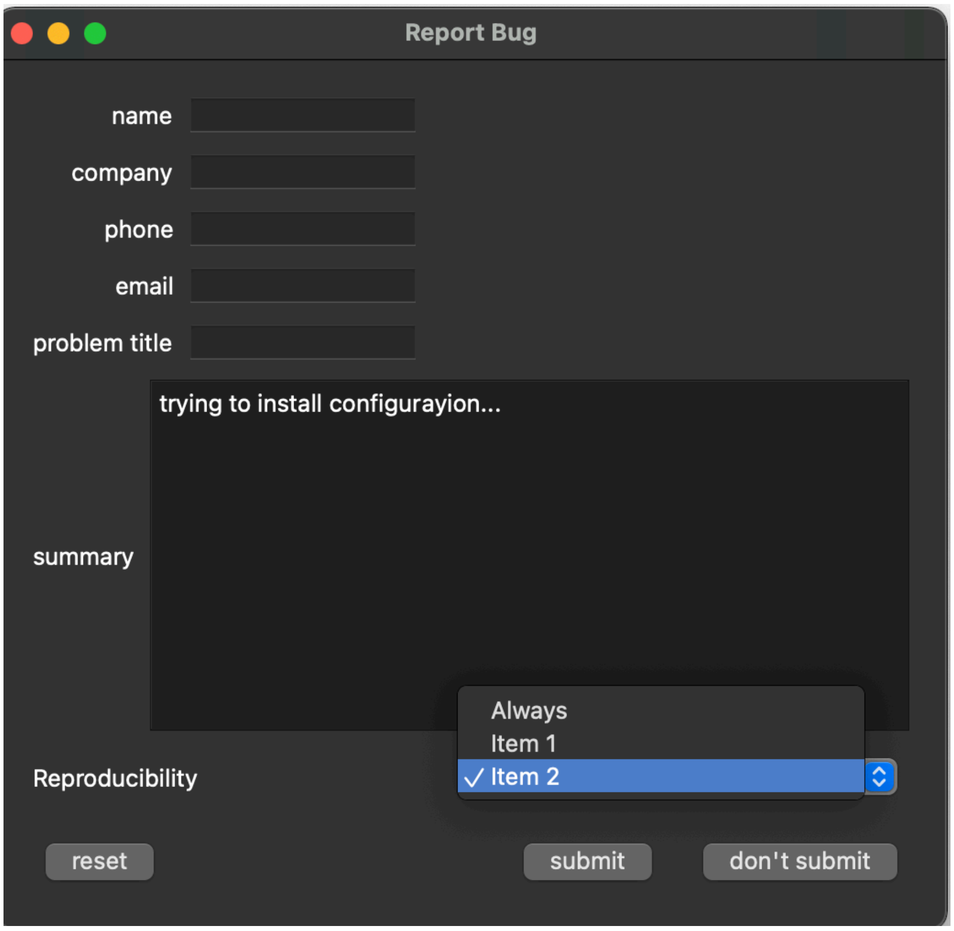
Reporocducibility

- To solve these problems we've proceeded in the following way :
- 1st we've created a QVBoxlayout (for the main layout) **<u>that is going to solve the misplacement</u>** problem of (rest , submit and don't submit)
- Then we added the QFormlayout that includes (name, company, email, problem title) to the main layout
- Next we have created a (QHBox) "Summarylayout" that includes two widgets QLabel "summary" and "summarytext" QTextEdit and then added to the main layout
- Then we have created a (QHBox) "myRepo" that includes two widgets "Reproducibility " QLabel and "Dropdown" Which is a QCombox

- At the end we added a spacer between rest and submit as follows
- Pre defined Method implemented for spacing :
- Botom2->setSpacing(200)
- Procedure behind the result (**solution**) :
- We have created two different QHbox botom2 that includes ("submit" , "reset") and botom1 with ("don't submit") as to implement setSpacing between submit and reset that belongs to the same QHBox
- In order to align "Formlayout" with the rest of the labels we called a predefined method :
- Formlayout->setFormAlignment(Qt::AlignLeft)

4- output results.

This is our final result



Report Bug

name

company

phone

email

problem title

summary: trying to install configurayion...

Reproducibility: Always / Item 1 / ✓ Item 2