



# Homework #7 Retrospect

- Full Credit
  - Minor changes to main() to initialize app and run Main\_window
  - Otherwise, just need to baseline Nim's Main\_window class and modify it to manage the Library Management System instead!
- Bonus
  - Just add tool bar setup and connect signals in Main\_window
- Extreme Bonus
  - Add a TreeModel to contain display data AND modify Library et. al. to support syncing to it
    - OR (better) redesign Library to act as the tree model directly
  - Add a TreeView linked to the TreeModel to a ScrollingWindow, and add that to Main\_window below the menu and toolbar

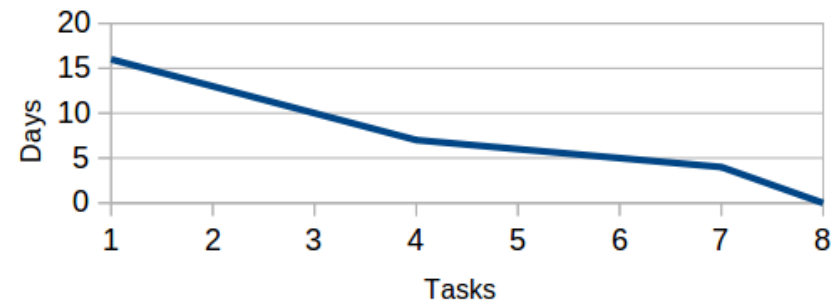
# Sprint #3 Backlog

Sprint # 3  
Start on Mar 08  
End on Mar 22  
Demo on TBD

**Complete Fields in Green!!!**

\*\*\* Create a new Sprint Backlog for EVERY SPRINT \*\*\*

Sprint Burn Chart



	Remaining	Completed (this day)
Total Task	16	
Day 1 Left	13	3
Day 2 Left	10	3
Day 3 Left	7	3
Day 4 Left	6	1
Day 5 Left	5	1
Day 6 Left	4	1
Day 7 Left	0	4

Feature ID	Description	Status
MW	Add Main_window class with menu creation code	Completed Day 1
MW	Create pass-through handlers for menu items to <u>controller.execute_cmd</u>	Completed Day 1
MW	Fix main() to create <u>app</u> and run Main_window instance	Completed Day 1
MT	Add <u>toolbar</u> creation code to Main_window class	Completed Day 2
MT	Find icons	Completed Day 2
MT	Update About dialog to credit artists	Completed Day 2
MP	Publication: Add to_strings method to extract every attribute in a vector<string>	Completed Day 3
MP	Library: Add publication_to_strings to pass through to_strings from Publication	Completed Day 3
MP	Controller: Add number_of_pubs and pub_to_strings methods	Completed Day 3
MP	Main_window: Define nested <u>Gtk::TreeModel::ColumnRecord</u> class to specify columns	Completed Day 4
MP	Main_window: Define <u>Glib::RefPtr&lt;Gtk::ListStore&gt;</u> as the Tree View to manage the data	Completed Day 5
MP	Main_window: Define <u>Gtk::TreeView</u> widget and <u>Gtk::ScrolledWindow</u> to scroll it	Completed Day 6
MP	Main_window: Define add_row method to add a new row to the tree model	Completed Day 7
MP	Main_window: Define update_rows to <u>resync</u> checkout data from library to tree model	Completed Day 7
MP	Main_window: Add widgets to <u>VBox</u> in the constructor	Completed Day 7
MP	Main_window: Update callbacks to sync data to tree model as the library changes	Completed Day 7



# Full Credit

## Changes to main.cpp

### Homework #6

```
#include <gtkmm.h>
#include "controller.h"
#include "library.h"

int main(int argc, char *argv[]) {
    Gtk::Main kit(argc, argv);
    Library library;
    Controller controller(library);
    controller.cli();
}
```

### Homework #7

```
#include <gtkmm.h>
#include "controller.h"
#include "library.h"
#include "main_window.h"

int main(int argc, char *argv[]) {
    auto app = Gtk::Application::create(
        argc, argv, "edu.uta.cse1325.lms");
    Library library;
    Controller controller(library);
    Main_window win(controller);
    win.set_title(
        "Library Management System");
    return app->run(win);
}
```

# Full Credit main\_window.h

```
#ifndef MAIN_WINDOW_H
#define MAIN_WINDOW_H

#include "controller.h"
#include <gtkmm.h>
```

```
class Main_window : public Gtk::Window
{
```

```
    public:
```

```
        Main_window(Controller& controller);
        virtual ~Main_window();
```

```
    protected:
```

```
        void on_list_pubs_click();           // Create a new publication
        void on_add_pub_click();             // Add a new publication
        void on_checkout_click();            // Checkout an existing publication
        void on_checkin_click();             // Checkout an existing publication
        void on_list_patrons_click();        // List all patrons
        void on_add_patron_click();          // Add a patron
        void on_manual_click();              // Display Help dialog
        void on_about_click();               // Display About dialog
        void on_quit_click();                // Exit the LMS
        void on_test_click();                // "Easter egg"
```

```
    private:
```

```
        Gtk::Label *msg;                    // Status message display
        Controller& _controller;             // The controller instance for LMS
```

```
};
#endif
```

**Constructors / Destructors**

**Callbacks (Observers)**

**Private Data**

# Full Credit

# main\_window.cpp

## Selected Code

```
#include "main_window.h"
```

```
Main_window::Main_window(Controller& controller) : _controller{controller} {  
    set_default_size(640, 480);
```

```
    // Put a vertical box container as the Window contents  
    Gtk::Box *vbox = Gtk::manage(new Gtk::Box(Gtk::ORIENTATION_VERTICAL, 0));  
    add(*vbox);
```

```
    // //////////
```

```
    // M E N U
```

```
    // Add a menu bar as the top item in the vertical box
```

```
    Gtk::MenuBar *menubar = Gtk::manage(new Gtk::MenuBar());
```

```
    vbox->pack_start(*menubar, Gtk::PACK_SHRINK, 0);
```

```
    //      F I L E
```

```
    // Create a File menu and add to the menu bar
```

```
    Gtk::MenuItem *menuitem_file = Gtk::manage(new Gtk::MenuItem("_File", true));
```

```
    menubar->append(*menuitem_file);
```

```
    Gtk::Menu *filemenu = Gtk::manage(new Gtk::Menu());
```

```
    menuitem_file->set_submenu(*filemenu);
```

```
    //      T E S T
```

```
    // Append Test (the "Easter egg") to the File menu
```

```
    Gtk::MenuItem *menuitem_test = Gtk::manage(new Gtk::MenuItem("_Test", true));
```

```
    menuitem_test->signal_activate().connect(  
        sigc::mem_fun(*this, &Main_window::on_test_click));
```

```
    filemenu->append(*menuitem_test);
```

```
    //... the rest of the menu goes here
```



# Full Credit

## main\_window.cpp

### Selected Code

```
// ... the rest of the menu goes here

// Make the box and everything in it visible
vbox->show_all();
}

// ///////////////////////////////////
// C A L L B A C K S
// ///////////////////////////////////

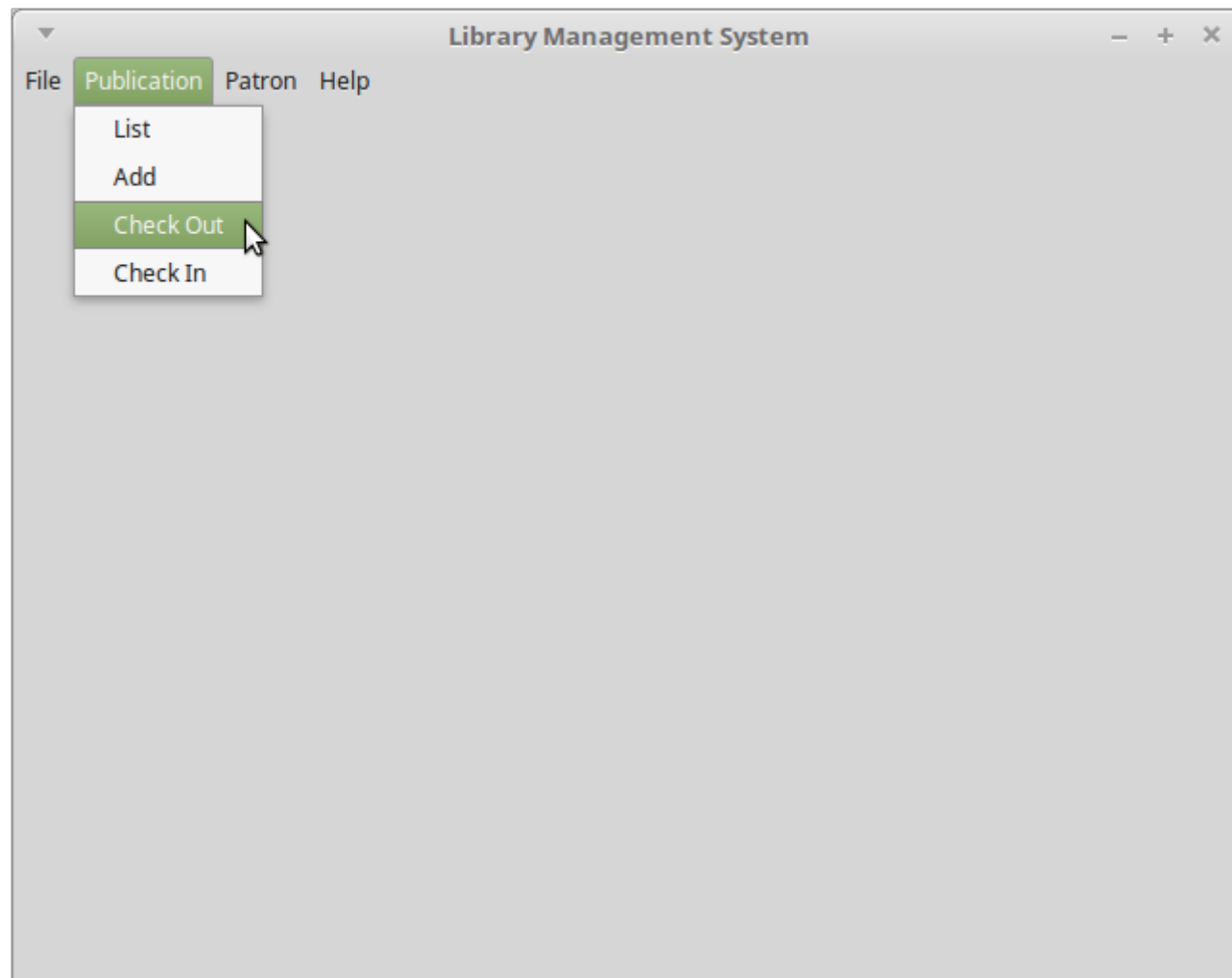
void Main_window::on_test_click() {_controller.execute_cmd(99);}
void Main_window::on_list_pubs_click() {_controller.execute_cmd(2);}
void Main_window::on_add_pub_click() {_controller.execute_cmd(1);}
void Main_window::on_checkout_click() {_controller.execute_cmd(3);}
void Main_window::on_checkin_click() {_controller.execute_cmd(4);}
void Main_window::on_list_patrons_click() {_controller.execute_cmd(6);}
void Main_window::on_add_patron_click() {_controller.execute_cmd(5);}
void Main_window::on_manual_click() {_controller.execute_cmd(9);}

void Main_window::on_about_click() {
    Glib::ustring s = "<span size='20000' weight='bold'>Library Management System</span>\n<span
size='large'>Copyright 2018 by George F. Rice</span>\n<span size='small'>Licensed under Creative Commons Attribution
4.0 International\nRobot icon created by Freepik, used under free attribution license</span>";
    Gtk::MessageDialog dlg(*this, s, true, Gtk::MESSAGE_INFO, Gtk::BUTTONS_OK, true);
    dlg.run();
}

void Main_window::on_quit_click() {
    hide();
}
```

The MVC pattern really shines here!

# Full Credit



# Bonus

## main\_window.cpp

### Selected Code

```
// ... Menu Bar code goes here
// //////////////////////////////////
// T O O L B A R
// Add a toolbar to the vertical box below the menu
Gtk::Toolbar *toolbar = Gtk::manage(new Gtk::Toolbar);
vbox->add(*toolbar);

//      N E W      P U B L I C A T I O N
// Add a new publication
Gtk::ToolButton *new_pub_button = Gtk::manage(
    new Gtk::ToolButton(Gtk::Stock::NEW));
new_pub_button->set_tooltip_markup("Create a new publication");
new_pub_button->signal_clicked().connect(sigc::mem_fun(
    *this, &Main_window::on_add_pub_click));
toolbar->append(*new_pub_button);

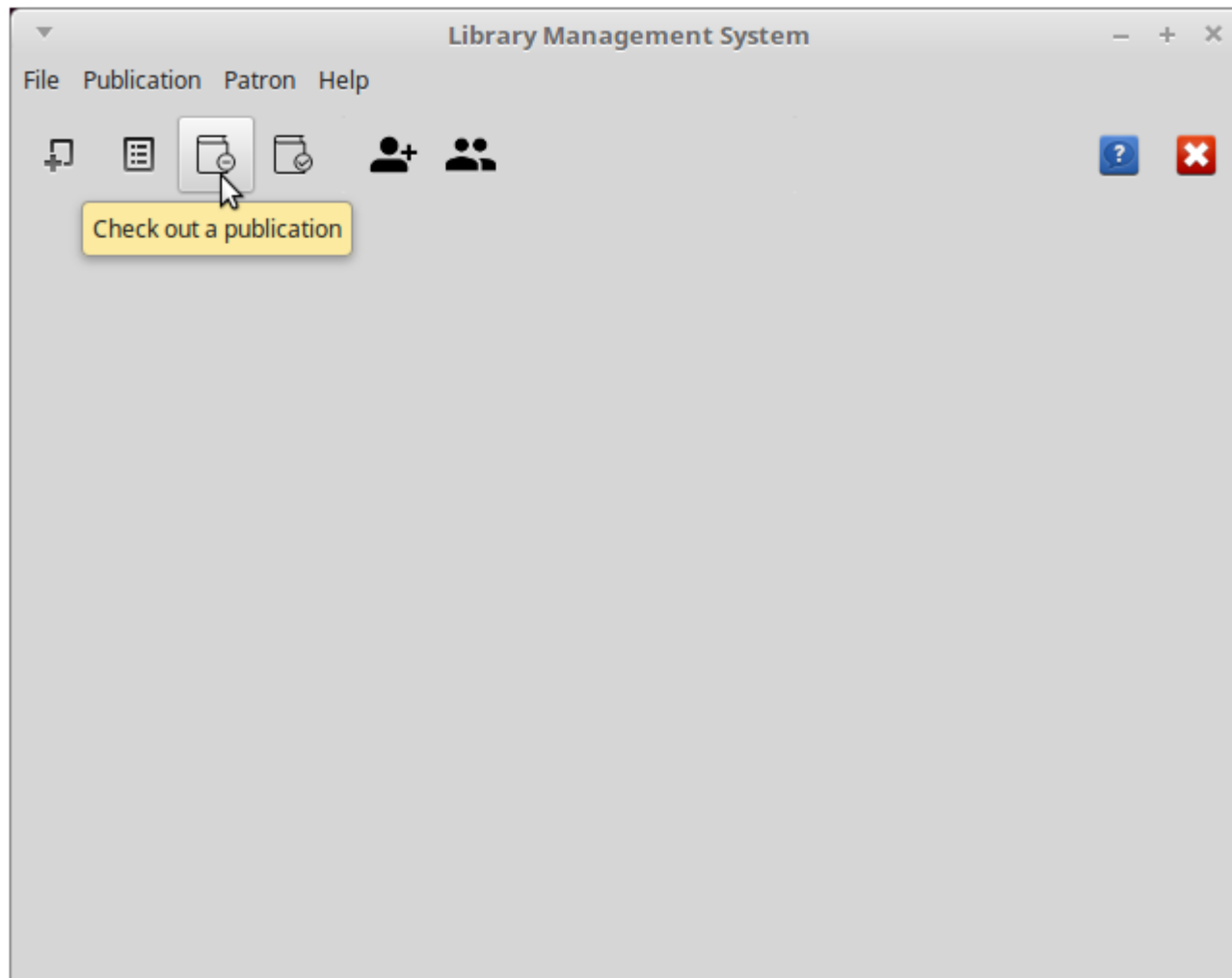
//...

//      C H E C K O U T
// Check out a publication
Gtk::Image *checkout_image = Gtk::manage(new Gtk::Image("checkout.png"));
Gtk::ToolButton *checkout_button = Gtk::manage(
    new Gtk::ToolButton(*checkout_image));
checkout_button->set_tooltip_markup("Check out a publication");
checkout_button->signal_clicked().connect(sigc::mem_fun(
    *this, &Main_window::on_checkout_click));
toolbar->append(*checkout_button);

//...
```



# Bonus



# Extreme Bonus Top Level Diagram

Main\_window

Library Management System

File Publication Patron Help

Gtk::ScrolledWindow

tree\_view.append\_column

ID	Title	Author	Copyright	Target Age	Genre	Media	ISBN	Patron	Patron Phone
0	The Firm	1951	adult	0440243323	action	book	John Grisham		
1	Foundation	1942	adult	0385177259	fiction	book	Isaac Asimov	Larry	817-555-1111
2	Foundation and Empire	1942	adult	0385177259	fiction	book	Isaac Asimov		
3	Second Foundation	1944	adult	0385177259	fiction	book	Isaac Asimov	Curly	817-555-2222
4	War of the Worlds	1977	teen	9780711969148	performance	audio	Jeff Wayne	Moe	817-555-3333
5	Willy Wonka and the Chocolate Factory	1971	children	0142410314	performance	video	Roald Dahl		

Gtk::TreeModel::ColumnRecord  
(aka ModelColumns)

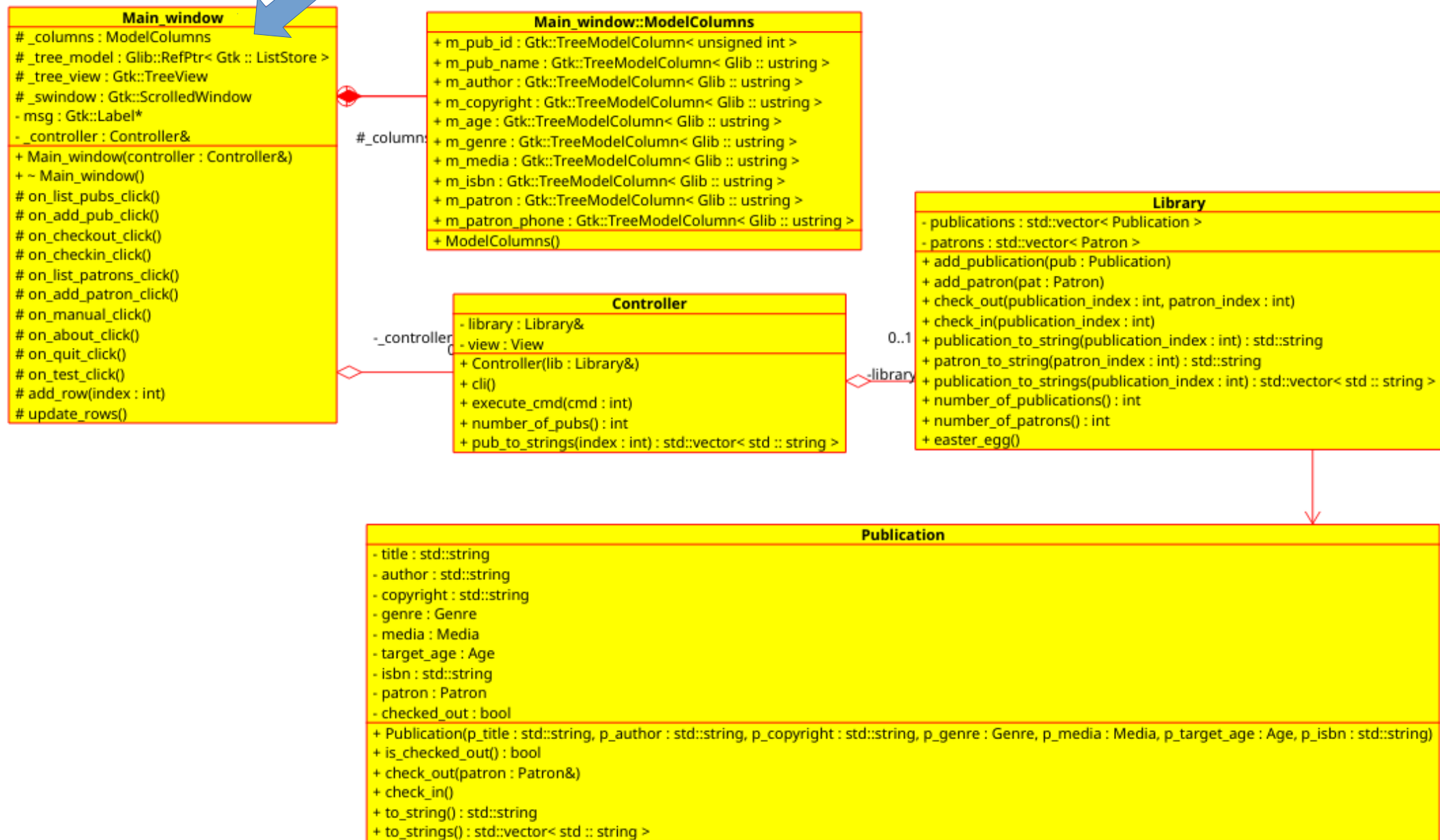
Gtk::TreeView

Library

Gtk::ListStore  
(aka TreeModel)



# Extreme Bonus Class Diagram



# Extreme Bonus

## main\_window.h

### Selected Code

```
class ModelColumns : public Gtk::TreeModel::ColumnRecord {
public:
    ModelColumns() {
        add(m_pub_id);
        add(m_pub_name);
        add(m_author);
        add(m_copyright);
        add(m_age);
        add(m_genre);
        add(m_media);
        add(m_isbn);
        add(m_patron);
        add(m_patron_phone);
    }
```

The ModelColumns class is nested inside Main\_window, and defines the columns that may be displayed.

```
Gtk::TreeModelColumn<unsigned int> m_pub_id;
Gtk::TreeModelColumn<Glib::ustring> m_pub_name;
Gtk::TreeModelColumn<Glib::ustring> m_author;
Gtk::TreeModelColumn<Glib::ustring> m_copyright;
Gtk::TreeModelColumn<Glib::ustring> m_age;
Gtk::TreeModelColumn<Glib::ustring> m_genre;
Gtk::TreeModelColumn<Glib::ustring> m_media;
Gtk::TreeModelColumn<Glib::ustring> m_isbn;
Gtk::TreeModelColumn<Glib::ustring> m_patron;
Gtk::TreeModelColumn<Glib::ustring> m_patron_phone;

};
ModelColumns _columns;
```



# Extreme Bonus

## main\_window.h

### Selected Code

```
Glib::RefPtr<Gtk::ListStore> _tree_model;
```

The `_tree_model` holds the data to be displayed. We'll need to keep it updated with changes to Library.

```
Gtk::TreeView _tree_view;
```

The `_tree_view` is the widget that actually displays the data. It will be linked to `_tree_model`.

```
Gtk::ScrolledWindow _swindow;
```

The `_swindow` contains the `_tree_view` and adds scroll bars any time the entire table can't be displayed. This is what is actually added to the VBox in the Window.

```
// This utility method syncs a (usually new) row from the library  
// to the tree model, so that the tree view will display it
```

```
void add_row(int index);
```

```
// This utility method updates the patron name and phone number  
// for all publications in the library, usually after a check out  
// or check in.
```

```
void update_rows();
```

# Extreme Bonus

## main\_window.cpp

### Selected Code

```
// Constructor: Set up menu bar and tool bar, then
```

```
// P U B L I C A T I O N   T A B L E
```

```
_tree_model = Gtk::ListStore::create(_columns);    Add the columns to _tree_model
```

```
_tree_view.set_model(_tree_model);                Link _tree_model to _tree_view
```

```
_tree_view.append_column("ID", _columns.m_pub_id); Add the columns to _tree_view
```

```
_tree_view.append_column("Title", _columns.m_pub_name);
```

```
_tree_view.append_column("Author", _columns.m_author);
```

```
_tree_view.append_column("Copyright", _columns.m_copyright);
```

```
_tree_view.append_column("Target Age", _columns.m_age);
```

```
_tree_view.append_column("Genre", _columns.m_genre);
```

```
_tree_view.append_column("Media", _columns.m_media);
```

```
_tree_view.append_column("ISBN", _columns.m_isbn);
```

```
_tree_view.append_column("Patron", _columns.m_patron);
```

```
_tree_view.append_column("Patron Phone", _columns.m_patron_phone);
```

```
_swindow.add(_tree_view);                          Add _tree_view to the scrolled window
```

```
_swindow.set_policy(Gtk::POLICY_AUTOMATIC, Gtk::POLICY_AUTOMATIC);
```

```
_swindow.set_hexpand(true);
```

```
_swindow.set_vexpand(true);
```

```
vbox->pack_start(_swindow);
```

Add the scrolled window to the main window  
just below the menu and tool bars

```
vbox->show_all();
```

```
}
```



# Extreme Bonus

## main\_window.cpp

### Selected Code

```
void Main_window::add_row(int index) {
    // Add a new row to the tree model to hold the new publication
    Gtk::TreeModel::Row row = *(_tree_model->append());

    // Load the data into the new row (this is a new Controller method!)
    std::vector<std::string> record = _controller.pub_to_strings(index);

    row[_columns.m_pub_id] = index;
    row[_columns.m_pub_name] = record[0];
    row[_columns.m_isbn] = record[1];
    row[_columns.m_author] = record[2];
    row[_columns.m_copyright] = record[3];
    row[_columns.m_genre] = record[4];
    row[_columns.m_media] = record[5];
    row[_columns.m_age] = record[6];
    row[_columns.m_patron] = record[7];
    row[_columns.m_patron_phone] = record[8];
}

void Main_window::update_rows() {
    // Reload all of the _tree_view cells that may have changed in _tree_model
    Gtk::TreeIter iter = _tree_model->get_iter("0");
    for(int i=0; i<_controller.number_of_pubs(); ++i) {
        std::vector<std::string> record = _controller.pub_to_strings(i);
        (*iter)[_columns.m_patron] = record[7];
        (*iter)[_columns.m_patron_phone] = record[8];
        iter++;
    }
}
```

# Extreme Bonus

## main\_window.cpp

### Selected Code

```
void Main_window::on_test_click() {
    // Remember how many publications already are displayed in _tree_view
    int index = _controller.number_of_pubs();

    // Add 1 or more?
    _controller.execute_cmd(99);

    // Add rows to _tree_view for any new publications added above
    for (int i=index; i<_controller.number_of_pubs(); ++i) add_row(i);
}

void Main_window::on_add_pub_click() {
    int index = _controller.number_of_pubs();
    _controller.execute_cmd(1);
    for (int i=index; i<_controller.number_of_pubs(); ++i) add_row(i);
}

void Main_window::on_checkout_click() {
    _controller.execute_cmd(3);

    // Sync _tree_view's checkin / checkout cells from Library
    update_rows();
}

void Main_window::on_checkin_click() {
    _controller.execute_cmd(4);
    update_rows();
}
```



# Extreme Bonus

Library Management System									
File Publication Patron Help									
									
ID	Title	Author	Copyright	Target Age	Genre	Media	ISBN	Patron	Patron Phone
0	The Firm	1991	adult	0440245923	fiction	book	John Grisham		
1	Foundation	1942	adult	0385177259	fiction	book	Isaac Asimov	Larry	817-555-1111
2	Foundation and Empire	1943	adult	0385177259	fiction	book	Isaac Asimov		
3	Second Foundation	1944	adult	0385177259	fiction	book	Isaac Asimov	Curly	817-555-2222
4	War of the Worlds	1977	teen	9780711969148	performance	audio	Jeff Wayne	Moe	817-555-3333
5	Willy Wonka and the Chocolate Factory	1971	children	0142410314	performance	video	Roald Dahl		