WILFRID LAURIER UNIVERSITY
LAURIER
Inspiring Lives.

An Architectural Framework for Consistent UI in Android App Development

Abdul-Rahman Mawlood-Yunis
Wilfrid Laurier University
Waterloo, Canada
amawloodyunis@wlu.ca

The Thirteenth International Conference on

Building and Exploring Web Based Environments

WEB 2025

March 09 to March 13, 2025 - Lisbon, Portugal

Introduction

- The UI is crucial in app development.
- A framework for a consistent UI in Android apps is proposed.
- Utilizes OOP principles like abstraction and inheritance.

Introduction

The UI is crucial in app development

- The user interface (UI) holds significant importance in interactive apps and application development, particularly in mobile apps
- An attractive and user-friendly visual interface becomes increasingly crucial and plays an essential role in determining their success
- Extensive research in software engineering, design patterns, software architecture, human interaction, and related fields have been dedicated to the proper UI design for interactive apps and applications

Introduction

- The UI is crucial in app development.

A framework for a consistent UI in Android apps is proposed.

this work takes a broad approach by focusing on reusing the whole or important components of the page as the user navigates between different screens of an app. In other words, this work develops an architectural framework that enables persistent UI across app screens, ensuring a cohesive user experience as users move from one screen to another.

WILFRID LAURIER UNIVERSITY LAURIER *Inspiring Lives.*

Introduction

- The UI is crucial in app development.
- A framework for a consistent UI in Android apps is proposed.
- Utilizes OOP principles like abstraction and inheritance.

Overview of the Proposed Architecture

- Key components:

- Base View

- Derived Views

- Customizing Views Behavior

WILFRID LAURIER UNIVERSITY LAURIER Inspiring Lives.

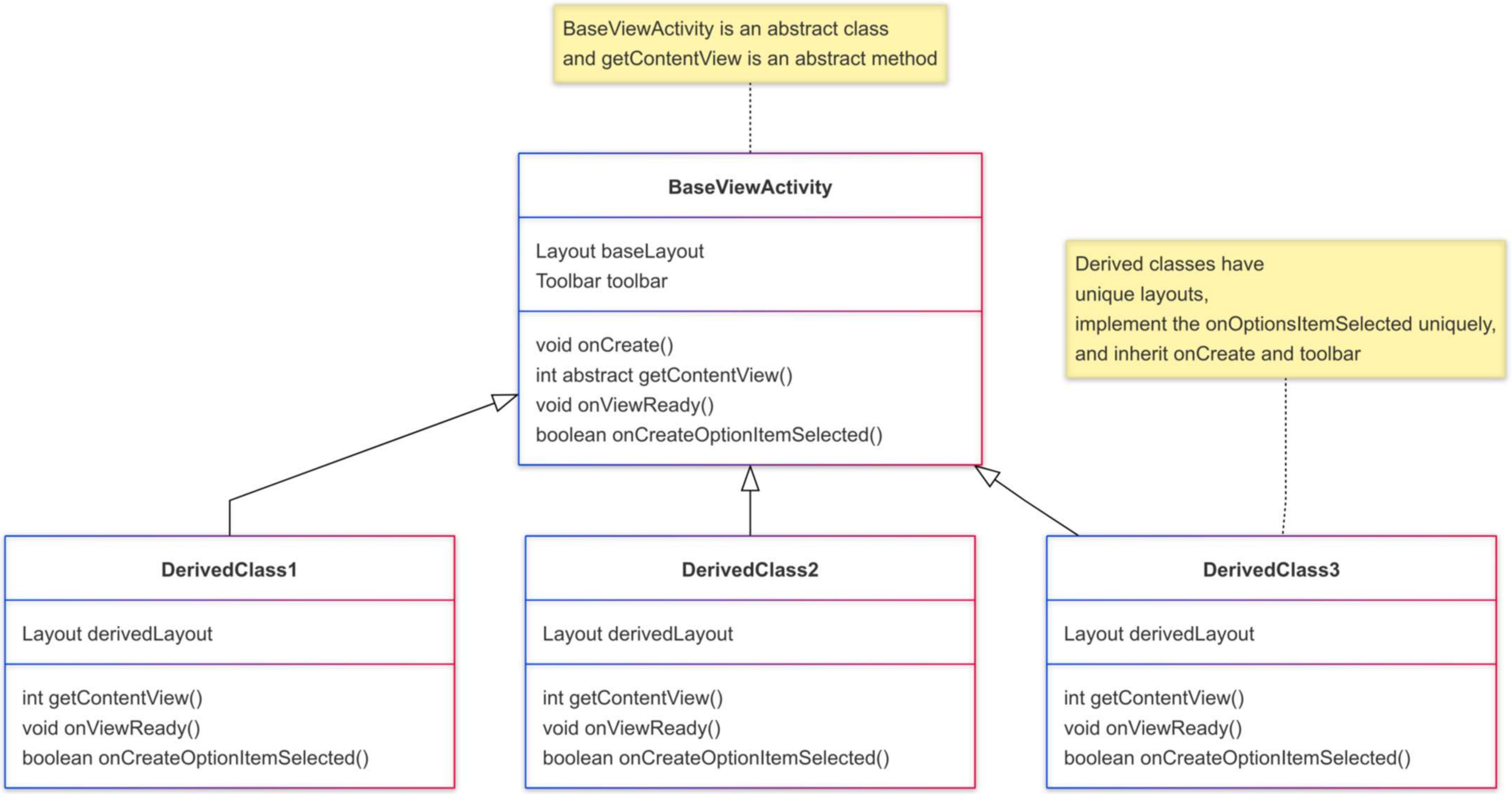
Base View

- Defines a common structure for all screens.
- Includes shared UI components.
- Implement feature that enables extension, i.e., set page content
- Include helper methods used by derived classes as abstract method.

Derived Views

- Extend the Base View, to inherit free functionality (code reuse)
- Customize UI components and behavior.
- Override abstract methods.

Architecture Diagram



WILFRID LAURIER UNIVERSITY
LAURIER
Inspiring Lives.

Implementation

The main components that implementation to be implemented include:

- - Base View implementation
- - Derived Views implementation
- - UI customization

Base View snippet

```
public abstract class BaseActivity extends AppCompatActivity {
…

   @Override
   protected void onCreate(Bundle savedInstanceState) {

       setContentView(R.layout.activity_base);

 Toolbar myToolbar = findViewById(R.id.in_base_my_toolbar);

 setSupportActionBar(myToolbar);


       onViewReady(savedInstanceState, getIntent());
   }

   protected void onViewReady(Bundle savedInstanceState, Intent
intent) {
       // To be used by child activities.
   }

   protected abstract int getContentView();

public boolean onOptionsItemSelected(MenuItem menuItem) {
…
}
```

WILFRID LAURIER UNIVERSITY
LAURIER
Inspiring Lives.

Derived Views implementation Snippet

```
public class NewMainActivity extends BaseActivity implements
OnItemSelectedListener {

    LinearLayout linearLayout;

    @Override
    protected int getContentView() {
        return R.layout.first_page;
    }

    @Override
    protected void onViewReady(Bundle savedInstanceState,
                      Intent intent) {
        super.onViewReady(savedInstanceState, intent);

        linearLayout = findViewById(R.id.baseLayout);
        LayoutInflater layoutInflater =
            LayoutInflater.from(NewMainActivity.this);

     layoutInflater.inflate(getContentView(), linearLayout, true);

    }


}
```

UI customization snippet

```
public class ChildActivity extends BaseActivity {
…
public boolean onOptionsItemSelected(MenuItem menuItem) {
        int id = menuItem.getItemId();
        String phoneNumber = "613 000 0000";

   switch (id) {

   case R.id.action_one:
        Intent intent = new Intent(ChildActivity.this, ThirdActivity.class);
                startActivity(intent);
                break;
    case R.id.action_two:

       intent = new Intent(ChildActivity.this, NewMainActivity.class);
                startActivity(intent);
                break;
    case R.id.action_three:
                intent = new Intent(Intent.ACTION_DIAL);
                intent.setData(Uri.parse("tel:" + phoneNumber));
                if (intent.resolveActivity(getPackageManager()) != null) {
                    startActivity(intent);
                }
                break;

     case R.id.action_about:

        }
        return true;
    }
}
```

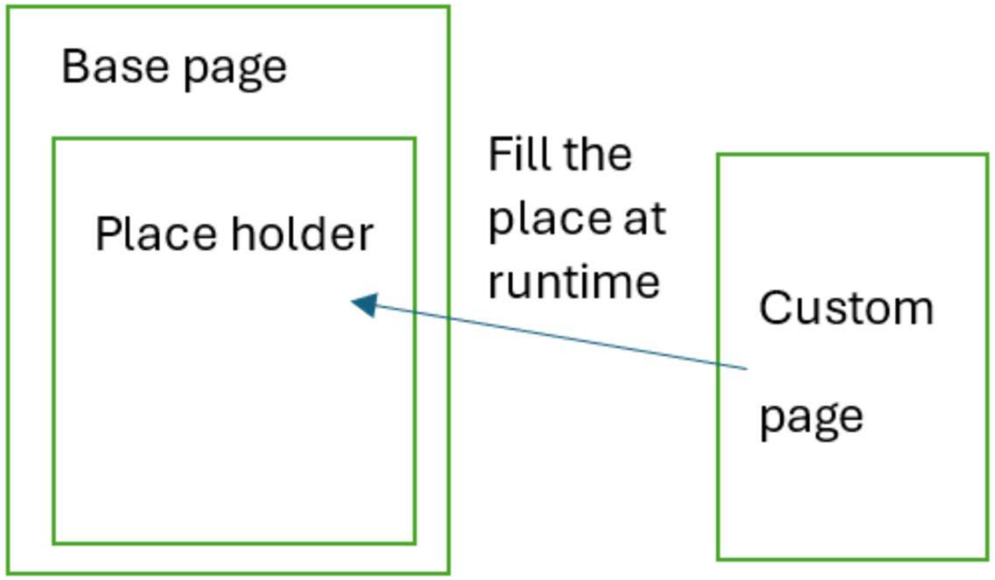
WILFRID LAURIER UNIVERSITY
LAURIER
Inspiring Lives.

Architecture Validation

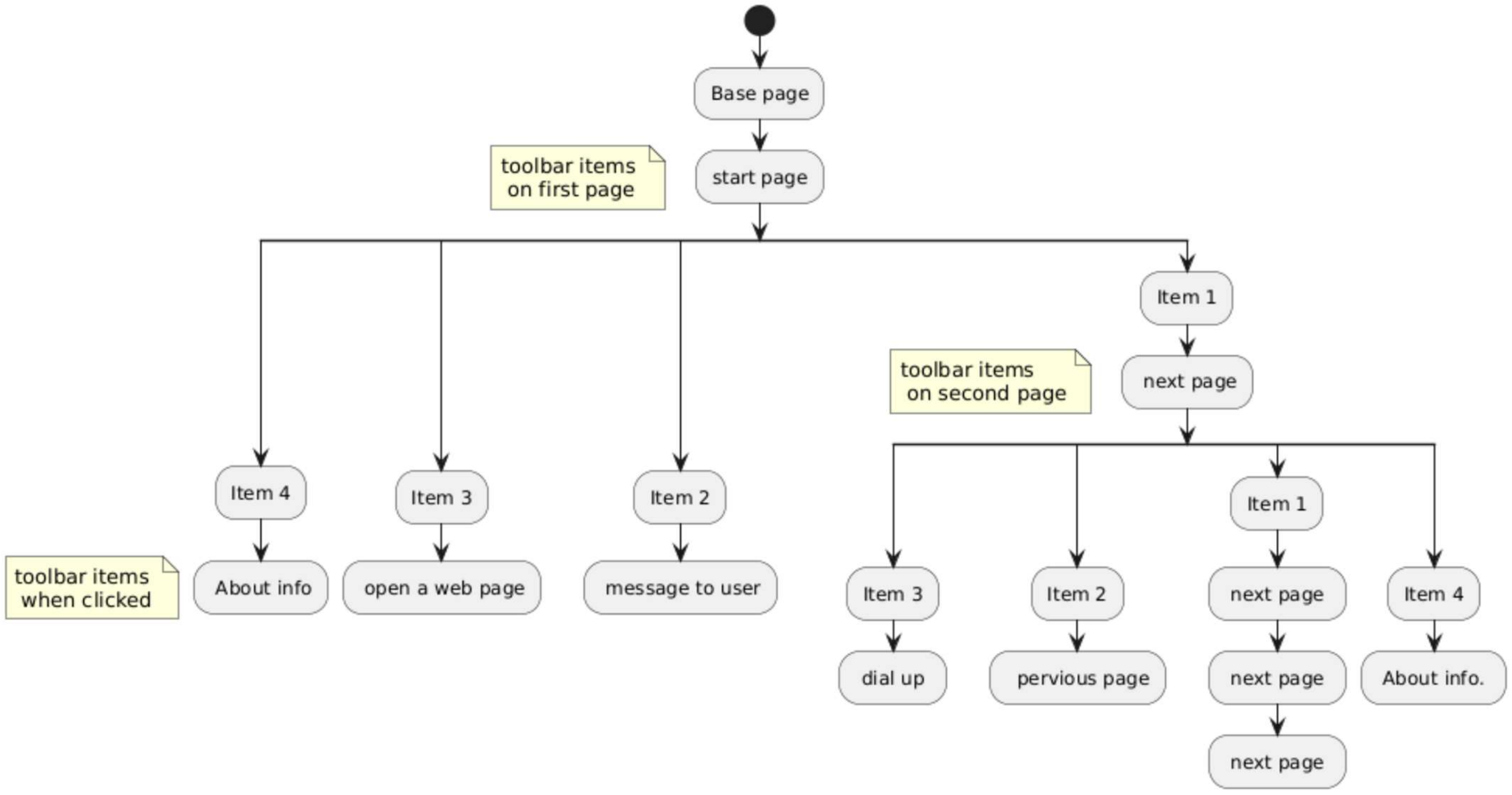
- An app was developed to validate the framework
  - newMian page inherit free toolbar,
  - Child1 page, implements different toolbar functionalities
  - All pages have consistent toolbar
- Consistent UI across multiple screens tested
- Toolbar component reused.

WILFRID LAURIER UNIVERSITY
LAURIER
Inspiring Lives.

- At run time, page content will replace the place holder content
- The replace could be done any number of times, it depends on the application
- Each page is a custom page of the app/application



App Navigation implementation using proposed framework



WILFRID LAURIER UNIVERSITY LAURIER *Inspiring Lives.*

Conclusion

- The proposed architecture:
  - Enhances UI consistency in Android apps.
  - Facilitates code reuse and modularity.
  - Can be extended to web development.

Thanks

You can download entire code here:

http://bohr.wlu.ca/amawloodyunis/framework/ArchitecturalFramework.zip

WILFRID LAURIER UNIVERSITY
LAURIER
Inspiring Lives.