Lab 3: GUI to post photos to timeline

Main UI

Profile Photo

Name of the user

Message of the post

Location of the post
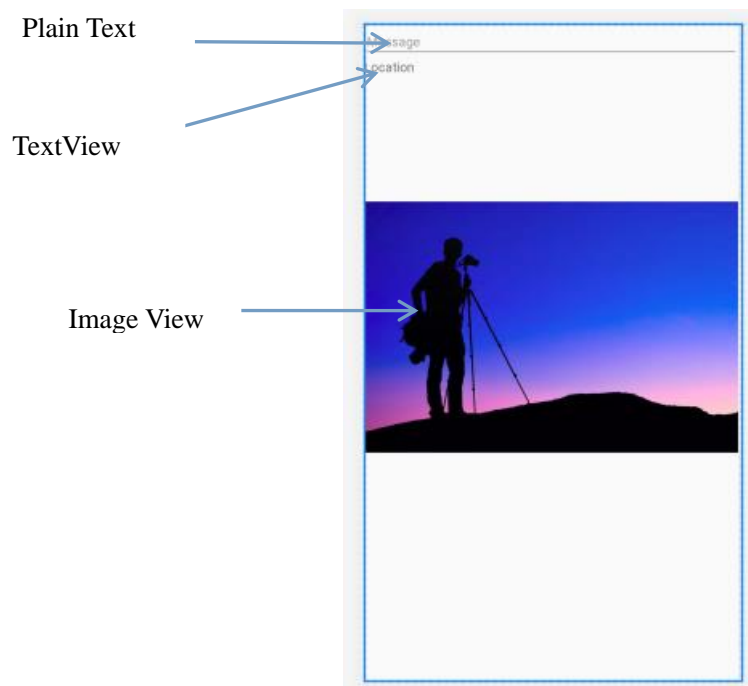
Photo of the post

Message of the post

Location of the post

Photo of the post

New Post Button
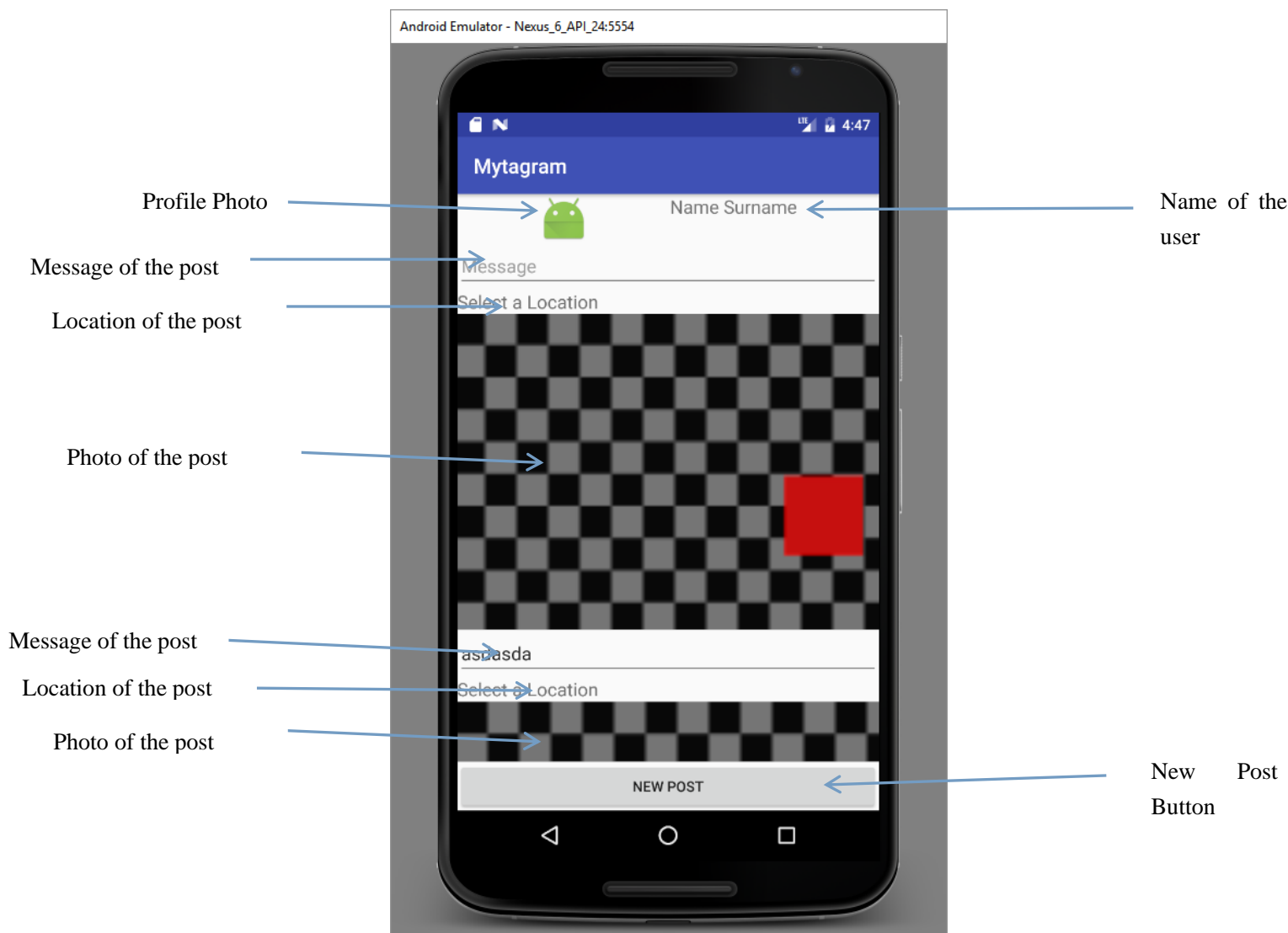
Exercise 1 (Design the main UI)

1. Design UI of the main activity using ConstraintLayout

Image View →

TextView

Name Surname

List View →

NEW POST   Button
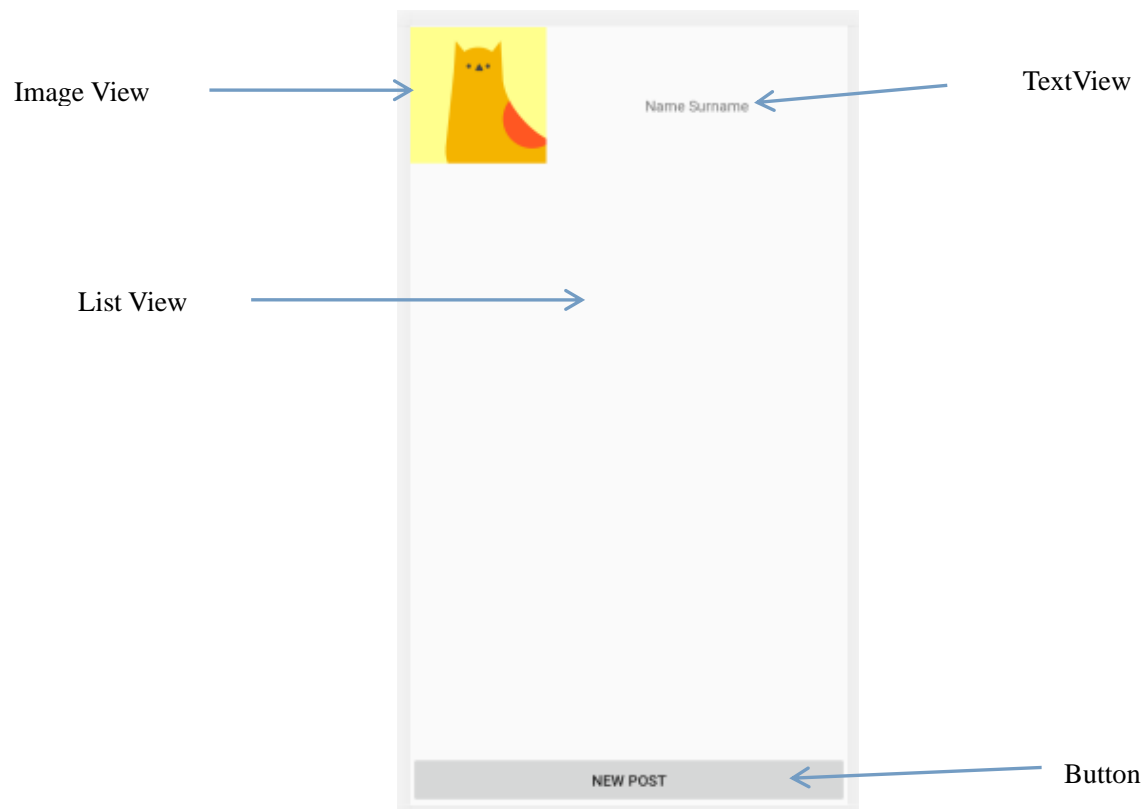
2. Create row.xml to display with Listview and design it as shown below.

Plain Text →   Message

          Location

TextView →

Image View →

3. Create a new Class named Post as below.

```java
public class Post {
    private Bitmap image;
    private Location location;
    private String message;

    public Bitmap getImage() {
        return image;
    }

    public void setImage(Bitmap image) {
        this.image = image;
    }

    public Location getLocation() {
        return location;
    }

    public void setLocation(Location location) {
        this.location = location;
    }

    public String getMessage() {
        return message;
    }

    public void setMessage(String message) {
        this.message = message;
    }

}
```

4. Create PostAdapter Class as below:

```java
public class PostAdapter extends BaseAdapter {

    List<Post> posts;

    private LayoutInflater inflater;

    public PostAdapter(Activity activity, List<Post> posts){

        this.posts = posts;

        inflater = (LayoutInflater) activity.getSystemService(

                Context.LAYOUT_INFLATER_SERVICE);

    }

    @Override

    public int getCount() {

        return posts.size();

    }

    @Override

    public Object getItem(int position) {

        return posts.get(position);

    }

    @Override

    public long getItemId(int position) {

        return position;

    }

    @Override

    public View getView(int position, View view, ViewGroup viewGroup) {

        View rowView;

        rowView = inflater.inflate(R.layout.row,null);

        EditText txtMessage = (EditText) rowView.findViewById(R.id.txtMessage);

        TextView txtLocation = (TextView) rowView.findViewById(R.id.txtLocation);

        ImageView imageView = (ImageView) rowView.findViewById(R.id.imageView);


        Post post = posts.get(position);

        txtMessage.setText(post.getMessage());

        imageView.setImageBitmap(post.getImage());

        if (post.getLocation() != null) {

            txtLocation.setText(post.getLocation().getLatitude() + " "

                    + post.getLocation().getLongitude());

        }

        return rowView;

    }

}
```
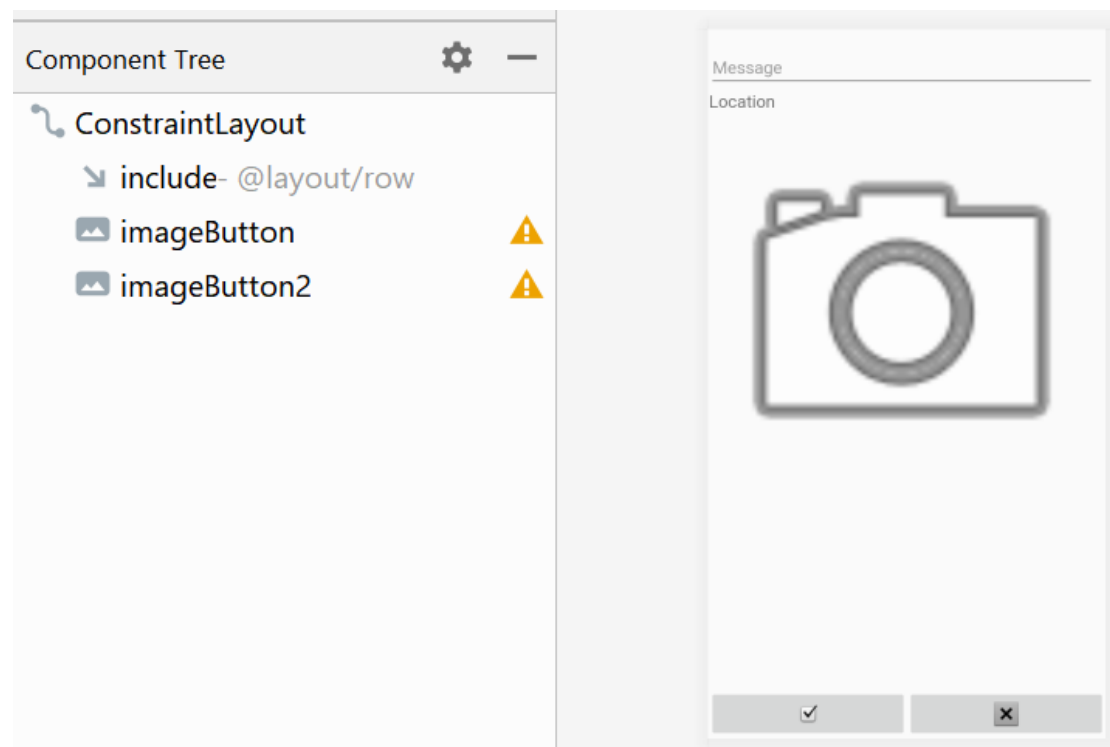
5. Set the Adapter of the ListView in the Main activity

```java
ListView listView = (ListView) findViewById(R.id.listview);

PostAdapter adapter = new PostAdapter(this, posts);

listView.setAdapter(adapter);
```

Exercise 2 (Design the New Post UI)
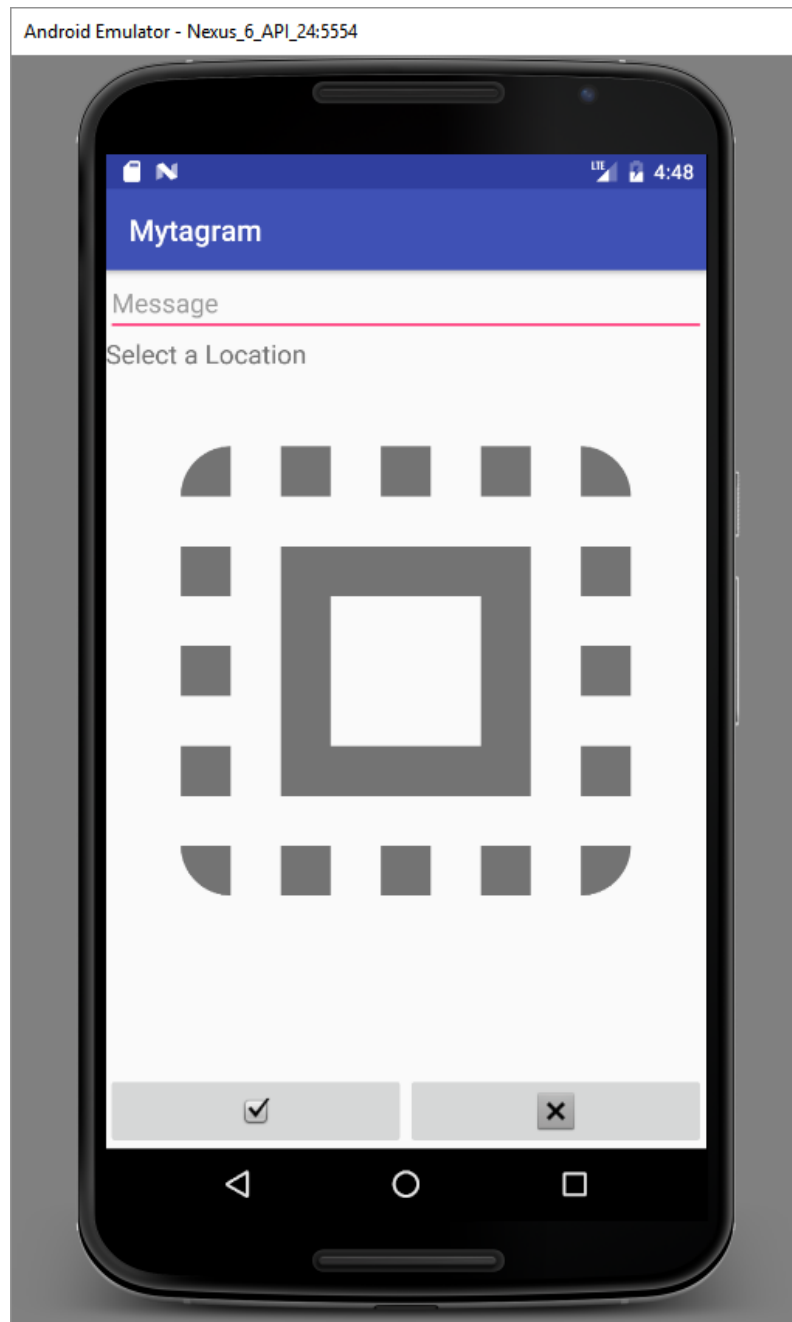
6. Create PostActivity as an empty activity and design its layout as below:

7. Set an onclick listener for the Post button in the main activity.

```java
btnPost = (Button) findViewById(R.id.btnPost);

btnPost.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View view) {

        Intent intent = new Intent(MainActivity.this, PostActivity.class);
        startActivityForResult(intent, POST_REQUEST);

    }

});
```

8. Override the onActivityResult as in MainActivity Below.

```java
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {

    if (requestCode == POST_REQUEST && resultCode == Activity.RESULT_OK){

        Post post = new Post();

        post.setMessage(data.getCharSequenceExtra("msg").toString());

        post.setImage((Bitmap) data.getParcelableExtra("bitmap"));

        posts.add(post);

        ((PostAdapter) listView.getAdapter()).notifyDataSetChanged();

    }

}
```

9. Set an onclick listener for the ImageView button in the post activity.

```java
img = (ImageView) findViewById(R.id.imageView);

img.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View view) {

        Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);

        if (intent.resolveActivity(getPackageManager()) != null){

            startActivityForResult(intent, CAPTURE_IMAGE);

        }

    }

});
```

10. Override the onActivityResult as in PostActivity Below.

```java
@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

    if (requestCode == CAPTURE_IMAGE && resultCode == Activity.RESULT_OK){

      Bundle bundle = data.getExtras();

       Bitmap image = (Bitmap) bundle.get("data");

       img.setImageBitmap(image);

    }

}
```

11. Set an onclick listener for Ok and Cancel Buttons in the post activity.

```java
txtMsg = (EditText)findViewById(R.id.txtMessage);

btnOk = (ImageButton) findViewById(R.id.btnOk);

btnOk.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View view) {

        Intent intent = new Intent();

        Bundle bundle = new Bundle();

        bundle.putCharSequence("msg",txtMsg.getText());

        bundle.putParcelable("bitmap",((BitmapDrawable)img.getDrawable()).getBitmap());

        intent.putExtras(bundle);

        setResult(Activity.RESULT_OK,intent);

        finish();

    }

});

btnCancel = (ImageButton) findViewById(R.id.btnCancel);

btnCancel.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View view) {

        setResult(Activity.RESULT_CANCELED);

        finish();
```

```
    }
});
```