

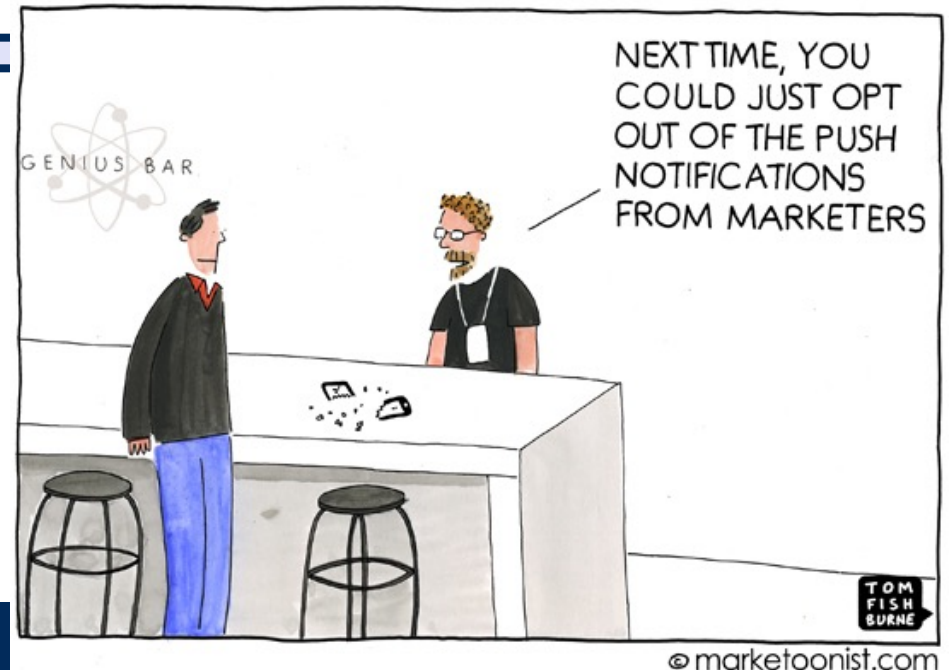
# Mobile Applications

## CSCI 448

### Lecture 36



## Notifications



# Learning Outcomes For Today



- List the components of a notification to the user
- Create a notification

# On Tap For Today



- Notifications
- Practice

# On Tap For Today



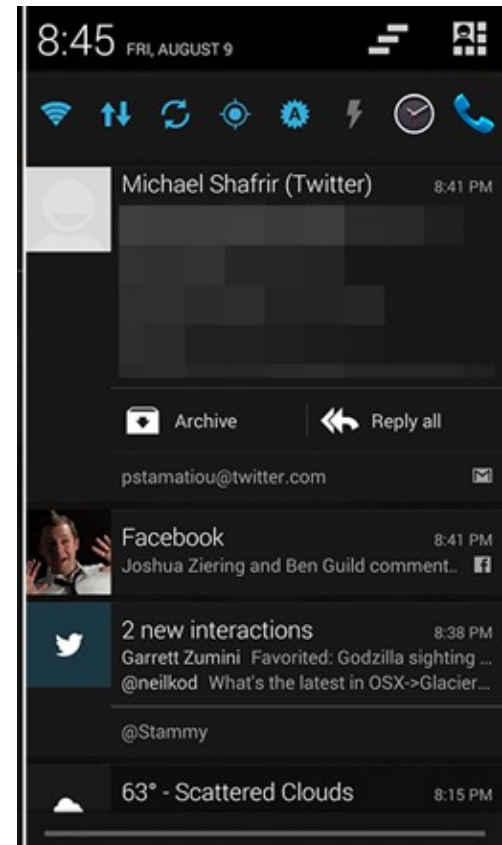
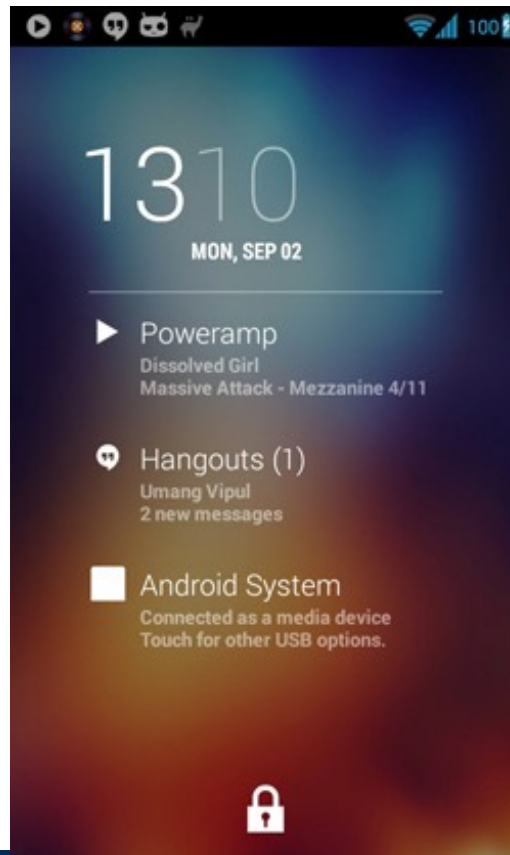
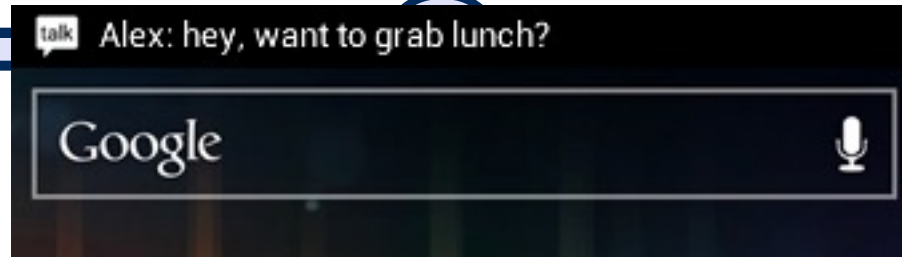
- Notifications
- Practice

# Notifications



- Since work can run in the background, if app is closed
  - Need to notify the user something has happened
  - “Hey! Go Check Out the App!”
- Use a notification

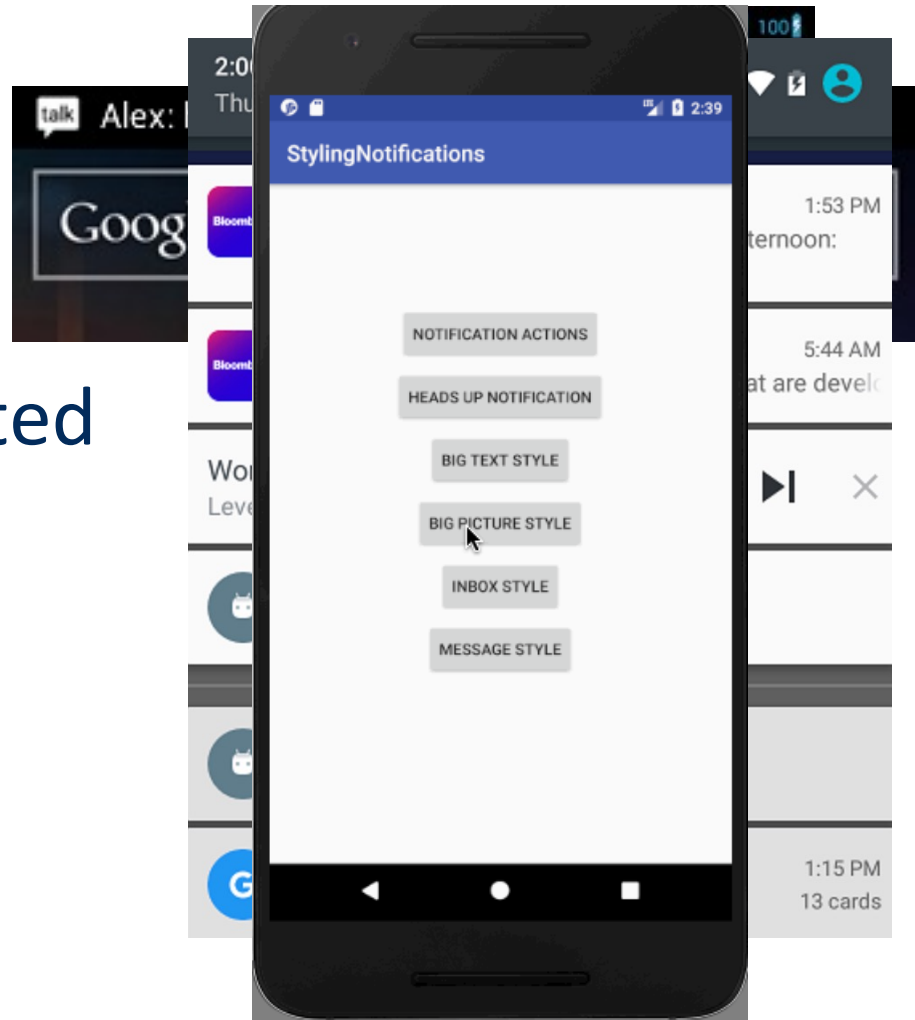
# Notifications



# Pieces of a Notification



- Notification Icon
- Ticker Text
- Content View
- Action when selected
- And many others



# Step 1 – Notification Manager



- Get the NotificationManager
  - Lives in the OS

```
val notificationManager = NotificationManagerCompat.from(context)
```

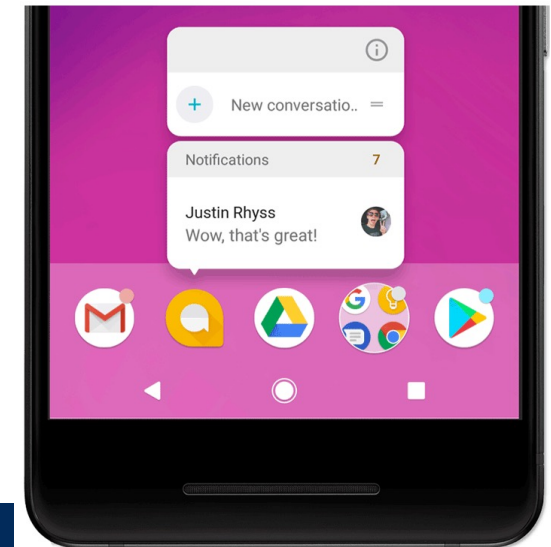


# Step 2 - Channel ID



- Create a NotificationChannel

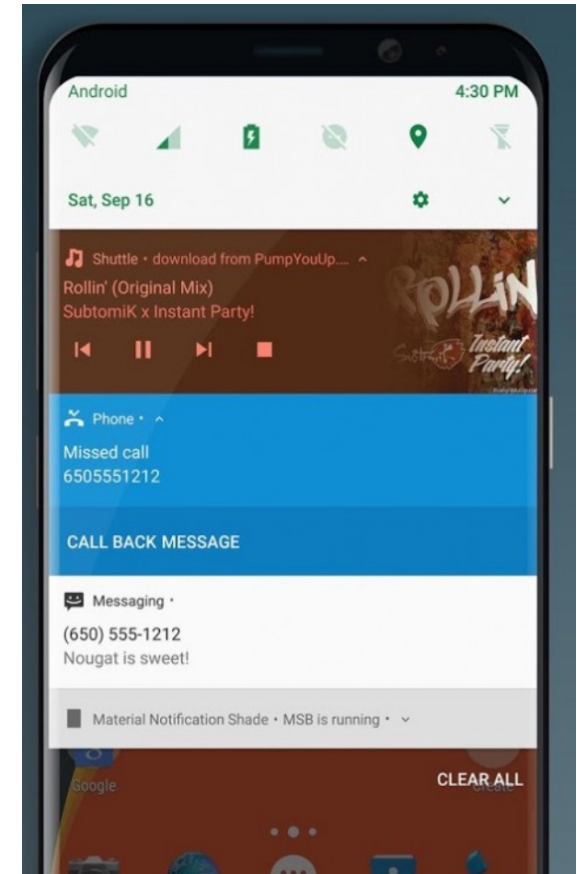
```
val CHANNEL_ID = "myMagicId"
val name = "ChannelName"
val importance = NotificationManager.IMPORTANCE_DEFAULT
val channel = NotificationChannel( CHANNEL_ID, name, importance ).apply {
    description = "This channel is for..."
    enableLights(true)
    lightColor = Color.BLUE
    enableVibrate(true)
    vibrationPattern = listOf(0L, 400L, 200L, 400L).toLongArray()
    setShowBadge(true)
    // other settings
}
notificationManager.createNotificationChannel(channel)
```



# Notification Importance



- None
  - Does not show in the shade
- Min
  - Only shows in the shade
- Low
  - Shows everywhere, not intrusive
- Default
  - Shows everywhere, makes noise, does not visually intrude
- High
  - Shows everywhere, makes noise, and peeks.



# Step 3 – Make the Notification



```
val notification = NotificationCompat.Builder(context, CHANNEL_ID)
    .setSmallIcon(R.drawable.icon) // REQUIRED!
    .setContentTitle("title to display")
    .setContextText("longer message to print")
    .setAutoCancel(true) // remove notification when user selects
    .build() // make the actual notification

notificationManager.notify(notifyID, notification)
```

# Android Design Patterns




- Behavioral Patterns
  1. **Command** – UI Event Handling, Retrofit Request Callback, Activity Result Callback, Permissions Callback, Location
  2. **Observer** – State, Flow, LiveData, Broadcast Receiver
  3. **Template Method** - IScreenSpec
- Creational Patterns
  4. **Builder** – Compose NavGraph, WorkRequest, Constraints, Retrofit, LocationRequest, Notification
  5. **Factory** – ViewModelFactory
  6. **Singleton** – ViewModelProvider, Repository, Room Database
- Structural Patterns
  7. **Decorator** – View Model
  8. **Façade** – DAO, Repository

# And many more features




- <https://developer.android.com/develop/ui/views/notifications/build-notification>


Collapsed

 Android System • 2 min ▾


**Screenshot captured**  
Tap to view your screenshot



Expanded

 Android System • 2 min ▲

**Screenshot captured**  
Tap to view your screenshot



**SHARE** **DELETE**


Collapsed

 Gmail • 8 min ▾

**Justin Rhyss**  
Movie night



Expanded

 Gmail • 8 min ▲

**Justin Rhyss**  
Movie night

Hey, do you have any plans for tonight? I was thinking a few of us could go watch a movie at the theater nearby since there won't be much going on for the next couple of weeks. There are some great options at 6 and 7pm, but whatever works best for you. If you have any suggestions for dinner beforehand hit reply!

**REPLY** **ARCHIVE**

**Messages** • now ▾

**Justin Rhyss**  
Do you want to see a movie tonight?



**REPLY** **ARCHIVE**

**Messages** • now ▾

**Justin Rhyss** Do you want to see a movie tonight?  
**You** Yeah, sounds great!



**REPLY** **ARCHIVE**

# On Tap For Today



- Notifications
- Practice

# For Next Time



- Lab12A
  - Have Alarm trigger posting a notification