Clipboard History - Critical Bug Fixes Implementation Guide

Based on Kamila's Testing Feedback - December 5, 2025

Bug Summary

ID	Bug Description	Severity	Status
#1	Share intent doesn't update clipboard	CRITICAL	TODO
#2	All bubbles look identical (no visual distinction)	CRITICAL	TODO
#3	Phantom transparent beige/yellow bubbles	HIGH	TODO
#4	Clipboard monitors after service stopped	HIGH	TODO
#5	Bubbles have no interactions (tap/drag)	CRITICAL	TODO
#6	No permission request on first launch	HIGH	TODO
#7	No quick show/hide bubbles toggle	MEDIUM	TODO

Fix #6: Permission Request Flow (continued)

File: app/src/main/java/com/sparesparrow/cliphist/MainActivity.kt

```
private fun showPermissionOnboarding() {
AlertDialog.Builder(this)
.setTitle("Welcome to Clipboard History!")
.setMessage("""
This app needs the following permissions to work:
        □ Display Over Other Apps
        - Required to show floating clipboard bubbles

        □ Notifications
        - Keep the service running in background

        □ Storage (Optional)
        - Save clipboard history for offline access

        We'll guide you through granting these permissions.
    """.trimIndent())
    .setPositiveButton("Continue") { _, _ ->
        requestOverlayPermission()
    }
    .setNegativeButton("Exit") { _, _ ->
        finish()
    }
    .setCancelable(false)
    .show()
```

```
}

private fun requestOverlayPermission() {
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
if (!Settings.canDrawOverlays(this)) {
AlertDialog.Builder(this)
.setTitle("Step 1/3: Display Over Apps")
.setMessage("Enable 'Display over other apps' to show floating bubbles")
.setPositiveButton("Open Settings") { _, _ ->
val intent = Intent(
Settings.ACTION_MANAGE_OVERLAY_PERMISSION,
Uri.parse("package:$packageName")
)
startActivityForResult(intent, REQUEST_OVERLAY_PERMISSION)
}
.show()
} else {
requestNotificationPermission()
}
}
}
```

```
private fun requestNotificationPermission() {
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.TIRAMISU) {
if (ContextCompat.checkSelfPermission(
this,
Manifest.permission.POST_NOTIFICATIONS
) != PackageManager.PERMISSION_GRANTED
) {
AlertDialog.Builder(this)
.setTitle("Step 2/3: Notifications")
.setMessage("Allow notifications to keep the clipboard service running")
.setPositiveButton("Allow") { _, _ ->
ActivityCompat.requestPermissions(
this,
arrayOf(Manifest.permission.POST_NOTIFICATIONS),
REQUEST_NOTIFICATION_PERMISSION
)
}
.show()
} else {
requestStoragePermission()
}
} else {
requestStoragePermission()
}
}

private fun requestStoragePermission() {
AlertDialog.Builder(this)
.setTitle("Step 3/3: Storage (Optional)")
.setMessage("Allow storage access to save clipboard history offline")
.setPositiveButton("Allow") { _, _ ->
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.R) {
// Android 11+ - Scoped storage
// No permission needed for app-specific storage
finishOnboarding()
} else {
ActivityCompat.requestPermissions(
this,
arrayOf(Manifest.permission.WRITE_EXTERNAL_STORAGE),
REQUEST_STORAGE_PERMISSION
)
}
}
.setNegativeButton("Skip") { _, _ ->
finishOnboarding()
}
.show()
}

private fun finishOnboarding() {
Toast.makeText(this, "Setup complete! Tap + to start clipboard service",
Toast.LENGTH_LONG).show()
// Start service automatically
startClipboardService()
}
```

```
override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
super.onActivityResult(requestCode, resultCode, data)
```

```
when (requestCode) {
    REQUEST_OVERLAY_PERMISSION -> {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M && Settings.canDra
            requestNotificationPermission()
        } else {
            Toast.makeText(this, "Overlay permission required", Toast.LENGTH_SHO
        }
    }
}
```

```
}
```

```
override fun onRequestPermissionsResult(
requestCode: Int,
permissions: Array<out String>,
grantResults: IntArray
) {
super.onRequestPermissionsResult(requestCode, permissions, grantResults)
```

```
when (requestCode) {
    REQUEST_NOTIFICATION_PERMISSION -> {
        if (grantResults.isNotEmpty() && grantResults[0] == PackageManager.PERM
            requestStoragePermission()
        } else {
            Toast.makeText(this, "Notification permission recommended", Toast.LEN
            requestStoragePermission()
        }
    }
    REQUEST_STORAGE_PERMISSION -> {
        finishOnboarding()
    }
}
}

companion object {
private const val REQUEST_OVERLAY_PERMISSION = 1001
private const val REQUEST_NOTIFICATION_PERMISSION = 1002
private const val REQUEST_STORAGE_PERMISSION = 1003
}
```

□ Fix #7: Quick Show/Hide Bubbles Toggle

File: `app/src/main/java/com/sparesparrow/cliphist/FloatingBubbleService.kt`

```
class FloatingBubbleService : Service() {
```

```
    private var bubblesVisible = true
    private val toggleButton: View by lazy { createToggleButton() }

    override fun onCreate() {
        super.onCreate()
        createNotificationChannel()
        startForeground(NOTIFICATION_ID, createNotification())

        // Add persistent toggle button
        addToggleButton()
    }

    private fun createToggleButton(): View {
        return LayoutInflater.from(this).inflate(
            R.layout.bubble_toggle_button,
            null
        ).apply {
            setOnClickListener {
                toggleBubblesVisibility()
            }
        }
    }

    private fun addToggleButton() {
        val params = WindowManager.LayoutParams(
            WindowManager.LayoutParams.WRAP_CONTENT,
            WindowManager.LayoutParams.WRAP_CONTENT,
            WindowManager.LayoutParams.TYPE_APPLICATION_OVERLAY,
            WindowManager.LayoutParams.FLAG_NOT_FOCUSABLE,
            PixelFormat.TRANSLUCENT
        ).apply {
            gravity = Gravity.TOP or Gravity.END
            x = 20
            y = 100
        }

        windowManager.addView(toggleButton, params)
    }

    private fun toggleBubblesVisibility() {
        bubblesVisible = !bubblesVisible

        activeBubbles.forEach { bubble ->
            bubble.visibility = if (bubblesVisible) View.VISIBLE else View.GONE
        }

        // Update toggle button appearance
        val icon = toggleButton.findViewById<ImageView>(R.id.toggle_icon)
        icon.setImageResource(
            if (bubblesVisible) R.drawable.ic_visibility
            else R.drawable.ic_visibility_off
        )

        Toast.makeText(
            this,
            if (bubblesVisible) "Bubbles shown" else "Bubbles hidden",
            Toast.LENGTH_SHORT
        ).show()
    }

    // Alternative: Add notification action
    private fun createNotification(): Notification {
        val toggleIntent = PendingIntent.getBroadcast(
            this,
            0,
            Intent(this, BubbleToggleReceiver::class.java),
            PendingIntent.FLAG_UPDATE_CURRENT or PendingIntent.FLAG_IMMUTA
        )

        return NotificationCompat.Builder(this, CHANNEL_ID)
            .setContentTitle("Clipboard History")
            .setContentText("Monitoring clipboard")
            .setSmallIcon(R.drawable.ic_clipboard)
            .addAction(
                if (bubblesVisible) R.drawable.ic_visibility_off else R.drawable.ic_visibili
                if (bubblesVisible) "Hide Bubbles" else "Show Bubbles",
                toggleIntent
            )
            .build()
    }

    override fun onDestroy() {
        super.onDestroy()
        windowManager.removeView(toggleButton)
    }
```

```
}

// Receiver for notification action
class BubbleToggleReceiver : BroadcastReceiver() {
override fun onReceive(context: Context, intent: Intent) {
val serviceIntent = Intent(context, FloatingBubbleService::class.java).apply {
action = ACTION_TOGGLE_BUBBLES
}
context.startService(serviceIntent)
}
}
```

Layout: `app/src/main/res/layout/bubble_toggle_button.xml`

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="56dp" android:layout_height="56dp"
android:background="@drawable/bubble_toggle_background">
```

```
    <ImageView
        android:id="@+id/toggle_icon"
        android:layout_width="32dp"
        android:layout_height="32dp"
        android:layout_gravity="center"
        android:src="@drawable/ic_visibility"
        android:tint="@color/white" />
```

</FrameLayout>

□ Updated Testing Instructions for Kamila

Test Scenario 1: Share Intent Integration

Test steps:

1. Open any app (Chrome, Messages, etc.)
2. Select and share text to "Clipboard History"
3. Verify "Select Bubble" dialog appears
4. Tap "Replace" - text should appear in clipboard history
5. Share again, tap "Append" - text should be appended
6. Share again, tap "Prepend" - text should be prepended

Expected result:

✅ Shared text appears in clipboard history immediately
✅ Replace/Append/Prepend work correctly
✅ Toast confirmation shows "Added to clipboard"

ADB test command:

```
adb shell am start -a android.intent.action.SEND
--es android.intent.extra.TEXT "Test share integration"
-n com.sparesparrow.cliphist/.ShareActivity
```

Test Scenario 2: Bubble Visual Distinction

Test steps:

1. Copy different content types:
   - Plain text: "Hello world"
   - URL: "https://github.com"
   - Phone: "+420777123456"
   - Email: "test@example.com"
2. Observe bubble colors on screen

Expected colors:

▯ Blue bubble = Plain text
▯ Teal bubble = URL
▯ Orange bubble = Phone number
▯ Purple bubble = Email
▯ Green bubble = Address

Expected content preview:

Each bubble should show first 20 characters of content inside

ADB test commands:

adb shell input text "Hello world"
adb shell input text "https://github.com"
adb shell input text "+420777123456"
adb shell input text "test@example.com"

Test Scenario 3: No Phantom Bubbles

Test steps:

1. Copy 5 different texts
2. Count bubbles on screen - should be exactly 5
3. Change wallpaper to different color
4. Count bubbles again - should still be 5
5. No transparent beige/yellow circles should appear

Expected result:

✅ Only real clipboard content bubbles visible
✅ No phantom bubbles based on background
✅ All bubbles fully opaque (80% opacity)

Czech translation for Kamila:

"Neměly by se objevovat žádné další průhledné bubliny při změně barvy pozadí"

Test Scenario 4: Service Stop Behavior

Test steps:

1. Start clipboard service (green banner "Clipboard service is running")
2. Copy text "Before stop"
3. Stop service (tap banner or notification)
4. Verify red banner "Clipboard service is stopped"
5. Copy text "After stop"
6. Open clipboard history - "After stop" should NOT appear

Expected result:

✅ Service stops monitoring immediately
✅ No new items added after stop
✅ Existing items remain in history

Czech feedback addressed:

"Po zastavení služby se nemá přidávat žádná další položka do historie"

Test Scenario 5: Bubble Interactions

Test steps:

1. Single tap hexagon bubble
   → Should show content preview toast OR flash animation (cube only)
2. Double tap hexagon bubble
   → Should copy content to system clipboard
   → Toast: "Copied to clipboard"
3. Long press hexagon bubble
   → Should show context menu (Delete, Share, Edit)
4. Drag hexagon bubble to screen edge
   → Should show smart action areas
   → Drop on action to execute (Open Link, Call, etc.)

Expected result for Kamila's "gray hexagon" issue:

✅ All bubbles respond to interactions
✅ No "gray line" artifacts
✅ Interactions work on all bubble types

Czech translation:

"Dvojklik na jakoukoliv bublinu zkopíruje obsah do schránky"

Test Scenario 6: Permission Flow

Test steps:

1. Uninstall and reinstall app (fresh install)
2. Launch app for first time
3. Verify welcome dialog appears
4. Tap "Continue"
5. Grant "Display over other apps" permission
6. Grant "Notifications" permission
7. Skip or grant "Storage" permission
8. Service should start automatically

Expected result:

✅ Clear step-by-step permission flow
✅ Each permission explained
✅ Service starts after setup
✅ Toast: "Setup complete!"

ADB test (simulate fresh install):

```
adb shell pm clear com.sparesparrow.cliphist
adb shell am start -n com.sparesparrow.cliphist/.MainActivity
```

Test Scenario 7: Show/Hide Bubbles Toggle

Test steps:

1. Start service - bubbles visible on screen
2. Find toggle button (top-right corner, small eye icon)
3. Tap toggle button
4. Verify all bubbles disappear
5. Toast: "Bubbles hidden"
6. Tap toggle again
7. Verify all bubbles reappear
8. Toast: "Bubbles shown"

Alternative: Notification action

1. Swipe down notification shade
2. Find "Clipboard History" notification
3. Tap "Hide Bubbles" action
4. Bubbles disappear
5. Tap "Show Bubbles" action
6. Bubbles reappear

Expected result:

✅ Toggle works instantly
✅ Service keeps running (monitoring continues)
✅ History still being updated when hidden
✅ Toggle state persists across app restarts

▯ Testing Checklist for Kamila

Before Retesting:

- [ ] Install updated APK with all 7 fixes
- [ ] Clear app data: `adb shell pm clear com.sparesparrow.cliphist`
- [ ] Grant all permissions during onboarding

Test Each Scenario:

- [ ] ✅ Test Scenario 1: Share intent works
- [ ] ✅ Test Scenario 2: Bubbles have different colors
- [ ] ✅ Test Scenario 3: No phantom bubbles
- [ ] ✅ Test Scenario 4: Service stops monitoring when stopped
- [ ] ✅ Test Scenario 5: Bubbles respond to tap/double-tap/long-press/drag
- [ ] ✅ Test Scenario 6: Permission flow on first launch
- [ ] ✅ Test Scenario 7: Show/hide bubbles toggle works

Czech Feedback Items:

- [ ] ✅ Dvojklik na hexagon → kopíruje do schránky (no more "gray line")
- [ ] ✅ Žádné béžové průhledné bubliny
- [ ] ✅ Po zastavení služby se nepřidávají položky

▯ Expected Test Results

Test	Before Fix	After Fix
Share intent	✖ Dialog shows but doesn't update	✅ Updates clipboard
Bubble colors	✖ All cyan/gray	✅ Color-coded by type
Phantom bubbles	✖ Beige circles appear	✅ No phantom bubbles
Service stop	✖ Still monitors	✅ Stops monitoring
Bubble tap	✖ No response	✅ Shows preview/copies
First launch	✖ No permission guide	✅ Step-by-step flow
Toggle visibility	✖ No quick toggle	✅ Hide/show button

▯ Deployment Steps

Step 1: Apply All Fixes (15 hours)

Day 1 (8 hours):

- Implement Fix #1: Share intent (2h)
- Implement Fix #2: Bubble colors (3h)
- Implement Fix #5: Bubble interactions (4h)

Day 2 (7 hours):

- Implement Fix #3: Phantom bubbles (1h)
- Implement Fix #4: Service stop (1h)
- Implement Fix #6: Permission flow (2h)
- Implement Fix #7: Show/hide toggle (2h)
- Testing and integration (1h)

Step 2: Build Test APK

```
./gradlew assembleDebug
adb install -r app/build/outputs/apk/debug/app-debug.apk
```

Step 3: Send to Kamila for Testing

Email template:

Subject: Clipboard History - Bug Fixes Ready for Testing

Ahoj Kamilo,

Děkuji za skvělou zpětnou vazbu! Opravil jsem všech 7 chyb, které jsi našla:

1. ✅ Share integrace nyní funguje - Replace/Append/Prepend aktualizují schránku
2. ✅ Bubliny mají různé barvy podle typu obsahu (modrá, zelená, oranžová, fialová)
3. ✅ Žádné další průhledné bubliny se neobjevují
4. ✅ Po zastavení služby se nepřidávají položky
5. ✅ Bubliny reagují na kliknutí (dvojklik zkopíruje)
6. ✅ První spuštění má průvodce oprávněními
7. ✅ Tlačítko pro skrytí/zobrazení bublin

Prosím otestuj podle přiloženého návodu a dej mi vědět, jestli vše funguje správně.

Díky!

Step 4: Collect Feedback

- Screenshot any remaining issues
- Note any new bugs discovered
- Verify all 7 fixes working

Step 5: Iterate if Needed

- Address any new feedback
- Retest until all issues resolved
- Prepare for production release

✅ Success Criteria

All bugs fixed when:

1. Share intent updates clipboard ✅
2. Each content type has unique bubble color ✅
3. No phantom transparent bubbles ✅
4. Service stops monitoring when stopped ✅
5. Bubbles respond to all interactions ✅
6. First launch guides through permissions ✅
7. Quick toggle to hide/show bubbles ✅

Kamila's feedback addressed:

- ✅ "Dvojklik na hexagon" → Now copies to clipboard properly
- ✅ "Béžové průhledné bubliny" → No more phantom bubbles
- ✅ "Po vypnutí další položka" → Service stops monitoring correctly

Ready for production when:

- All 7 test scenarios pass
- No new bugs discovered
- Kamila confirms fixes work
- Updated screenshots captured