Code Smells in `SharingApp` – Subhatav Dhali, July 17th 2023

1st Code Smell:

- **Smell Type:** Duplicated Code
- **Classes involved:**
  - `ContactList`
  - `ItemList`
- **Methods involved:**
  - `public boolean saveContacts(Context context)`
  - `public boolean saveItems(Context context)`
- **Description:** Both of these methods have similar implementation, and execute the same action on a different object.
- **Problem:** Any changes made to storage later on in project requires changing code on storage implementation throughout the project, which is time consuming as well as has chances that certain occurrences might be missed.
- **Solution:** Create a storage-handling class to hold such kind of implementations.

2nd Code Smell:

- **Smell Type:** Excessive use of Comments
- **Classes involved:** `EditContactActivity`
- **Methods involved:** `public void saveContact(View view)`
- **Description:** Comments are spread out across multiple lines, even in such place where functionality is clearly understandable from looking at the lines of code.
- **Problem:** Commenting on every block of code to give idea on what code is doing, is failure of developer to expose implementation idea to reader. Any future changes made in code will require updating of comments in many different places, and in case some occurrences are missed, future developers will be looking at outdated comments to understand the latest code flow.
- **Solution:** Careful refactoring is needed, like making variable names, method names, etc. self-explanatory, and then, use comments only to outline why that approach is used, rather than what approach is used.