

## Code Smells in the SharingApp

### 1. Inappropriate Use of Comments

Location	Description	Why is this a Problem?	Solution
All files	<i>Insufficiency:</i> Lack of Java documentation in between the imports and the start of the class. Some have comments but still lacking information.	Lack of information overview and other developers may have a hard time scanning through the code to understand faster.	Create a java doc for each of the class stating the functionality of the method, parameters and return value
AddContactActivity.java Line 56 & 62	<i>Excessive:</i> Unnecessary comment.	Redundancy. Code is already readable enough.	Remove comments on stated lines
AddItemActivity.java Line 97 & 103			
EditItemActivity.java Line 114, 120, 183, 189, 202, 209, 211, and 219	<i>Excessive:</i> Unnecessary comment. (114,120,183,189)  <i>Inappropriate:</i> Wrong Comment Placing (202, 209, 211, 219)	Redundancy. Code is already readable enough.  Too many comments going inside the method	Remove comments on stated lines. Then have a java doc on top of the toggleSwitch method if you want to explain the method parts further.
ItemAdapter.java Line 34, 35, 44	<i>Excessive:</i> Unnecessary comment.	Redundancy. Code is already readable enough.  Comments also describe methods too much. This can be an indication of Long Method	Remove comments on stated lines. (See other numbers of this document for more details of solutions to other violations)
ItemsFragment.java Line 38, 62, 82-85, 88-90, 97-99	<i>Excessive:</i> Unnecessary comment.	Redundancy. Code is already readable enough.	Remove comments on stated lines.
MainActivity.Java Line 40	<i>Excessive:</i> Unnecessary comment.	Redundancy. Code is already readable enough.  This also leads to Message Chains and Feature Envy	Remove comments on stated lines. (See other numbers of this document for more details)

		(see other numbers of this document for more details)	of solutions to other violations)
SectionsPagerAdapter.java Line 36	<i>Excessive:</i> Unnecessary comment.	Redundancy. Code is already readable enough.	Remove comments on stated lines.

## 2. Long Method

Location	Description	Why is this a Problem?	Solution
ItemAdapter.java getView Method	The getView method is quite long and handles multiple responsibilities.	Long methods can indicate that the method is more complex or has more occurring within it than it should.	Consider splitting it into smaller, more focused methods for better readability and easier maintenance.
MainActivity.java onCreate Method	The onCreate method is doing quite a bit of work.	Long methods can indicate that the method is more complex or has more occurring within it than it should.	Consider breaking down the setup logic into smaller methods for improved readability and easier testing.

## 3. Feature Envy

Location	Description	Why is this a Problem?	Solution
ItemAdapter.java View logic	The logic to show or hide the status based on the fragment's type should ideally be handled outside the adapter.	Adapters should not handle logic like this directly	It's better to encapsulate such logic within the fragment itself or a separate utility class.
MainActivity.java addItemActivity Method	The addItemActivity method directly handles the logic for starting the AddItemActivity.	More interested in the details of a class other than the one it is in.	Consider migrating it to AddItemActivity class

#### 4. Duplicated Code

Location	Description	Why is this a Problem?	Solution
Loading, Saving, and Editing methods (eg. LoadContacts(), LoadItems(), etc.)	They have basic purpose of reading files from streams and appending data to either a new or an existing file stream. However, the identical process of reading and writing is repeated multiple times within each method, differing only in the file involved.	Modifications to the storage system at a later stage of the project necessitate altering the storage implementation code across the entire project. This process is time-intensive and carries a risk of overlooking specific instances.	Creating a generic storage class capable of accommodating various object types and filenames, while also managing the process of reading and storing data to and from file streams.