

12	Methods with multiple responsibilities	FrmAccountSetting FrmDeleteTaxpayer FrmAddTaxProcess	Fixed	Some methods handled too many responsibilities, making the code harder to read and maintain.	Split methods into smaller, single-responsibility functions.
13	Repetitive code for validation	FrmDeleteTaxpayer FrmAddTaxProcess	Fixed	Repetitive code was used for validation across multiple places.	Reused methods in Utils to handle common validators.
14	Long methods difficult to understand	FrmAccountSetting FrmDeleteTaxpayer FrmAddTaxProcess	Fixed	Long methods made the code harder to understand and maintain.	Refactored into smaller, more descriptive methods.
15	Tight coupling with MongoDBUI	FrmAccountSetting FrmDeleteTaxpayer FrmAddTaxProcess MongoDBUI	Improved	The UI classes were directly dependent on the implementation of MongoDBUI.	Consider using interfaces or dependency injection to reduce coupling.
16	Lack of centralized error handling	FrmAccountSetting FrmDeleteTaxpayer FrmAddTaxProcess Utils	Improved	Error handling was scattered across different methods and not centralized.	Consolidate error handling in a single place or class.
17	Inconsistent SRP application	FrmAccountSetting FrmDeleteTaxpayer FrmAddTaxProcess Utils	Fixed	Methods and classes didn't consistently follow the Single Responsibility Principle (SRP).	Ensure that each class and method has only one responsibility.