

No.	Security Issue	Java File	Possible Vulnerabilities	Solution Implementation
1	Unmask Password	Login.java	<ul style="list-style-type: none"> - Attackers or bystanders can view the password as the user types. - Shoulder surfing or screen capture can easily expose user credentials. - This practice undermines trust and violates secure input handling standards. 	The field shows dots or asterisks instead of the actual characters typed by the user.
2	Password Hashing	<ul style="list-style-type: none"> - Register.java - MgmtUser.java 	Makes confidential information easily accessed/seen by unauthorized users	Utilized the bcrypt library extension
3	Missing Authorization	Login.java, SQLite.java	An attacker could read or access sensitive data thorough hidden admin pages, IDOR, etc.	Seperate product into frames with roles i.e if Login is successful, authenticated user is passed on the the frame.main, frame.admin
4	Missing Login Input Validation	Login.java, Register.java	Logins are not properly validated and thus accounts and data is inacessable	Create a conditional, where in a look up in the database is required to verify the data.
5	No Limit for Incorrect Password Attempts	-Login.java	No limits for attempts makes it vulnerable to brute force attacks and account takeover.	Added a session-based failed attempts counter for when you input an incorrect password on an existing account. The solution locks the user from attempting to log on until the program restarts.
6	Lock/Unlock has no effect	<ul style="list-style-type: none"> -Login.java -MgmtUser.java -SQLite.java 	<ul style="list-style-type: none"> Allows locked accounts to still access the program -Unauthorized Access 	Created a new function in SQLite.java called lockUser(username) to properly toggle the locked state of users.
7	No Password Complexity or Length Limit	-Register.java	Makes accounts easily targettable by brute force attacks and password spraying.	Implemented a minimum password length of 6 characters and requires the user to include an uppcase, lowercase, and a numeric character in their password.
8	High Risk for SQL Injection	SQLite.java	Possible risk of showing the database	Used the recommended commented out class "Prepared Statement in the addUser function
9	Program does not prohibit account registration that use e	<ul style="list-style-type: none"> -Register.java -SQLite.java 	Duplicate accounts can exist and can deny users of access to the program due to the overlap with another user of the same name. It can also cause data corruption	Added a checker in SQLite.java to identify if a user with that specific username already exists.