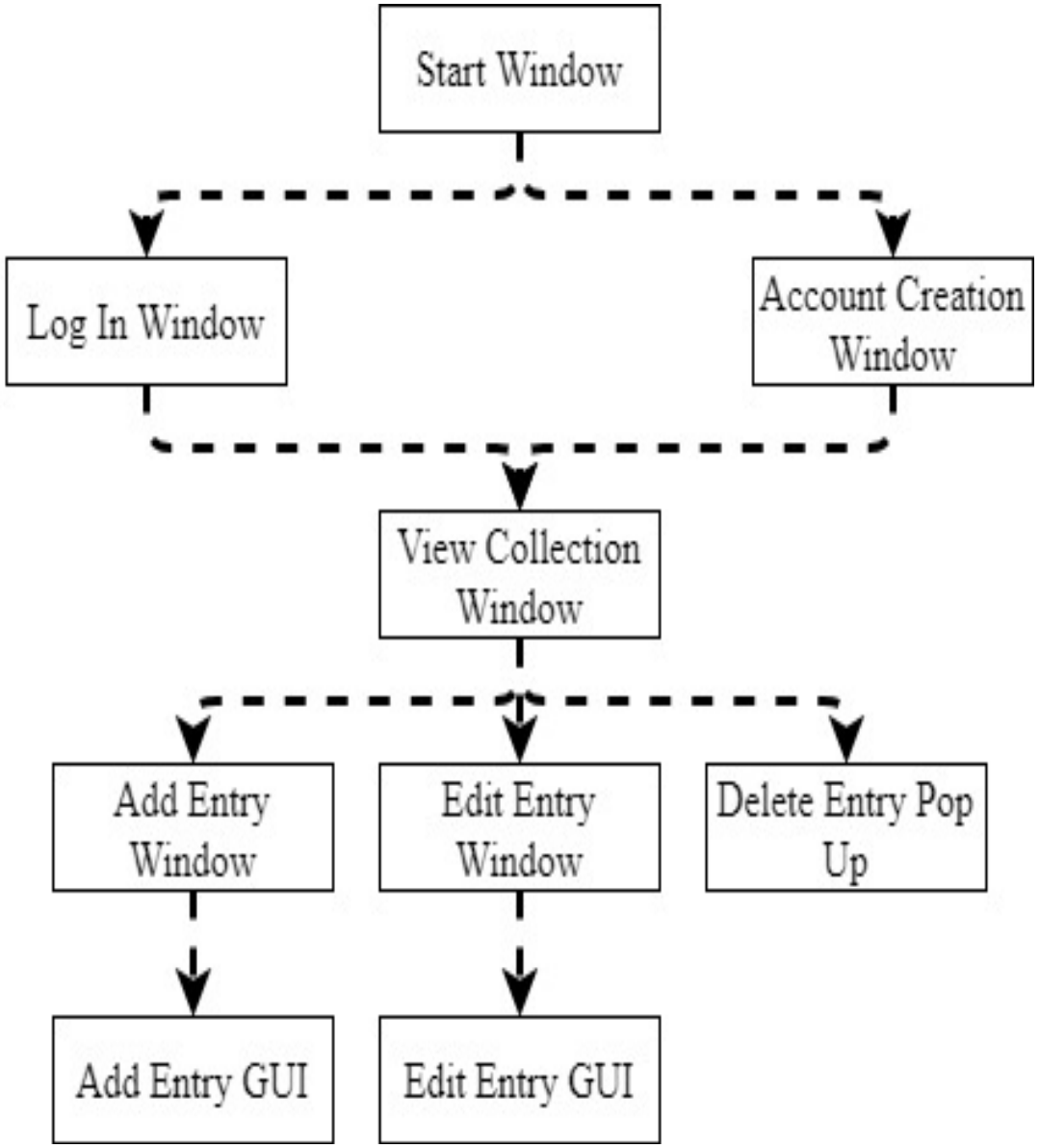
Criterion B: Design

Note: These designs have been modified according to client specifications. They will differ from the final product due to further client specifications. See Appendix A for details.

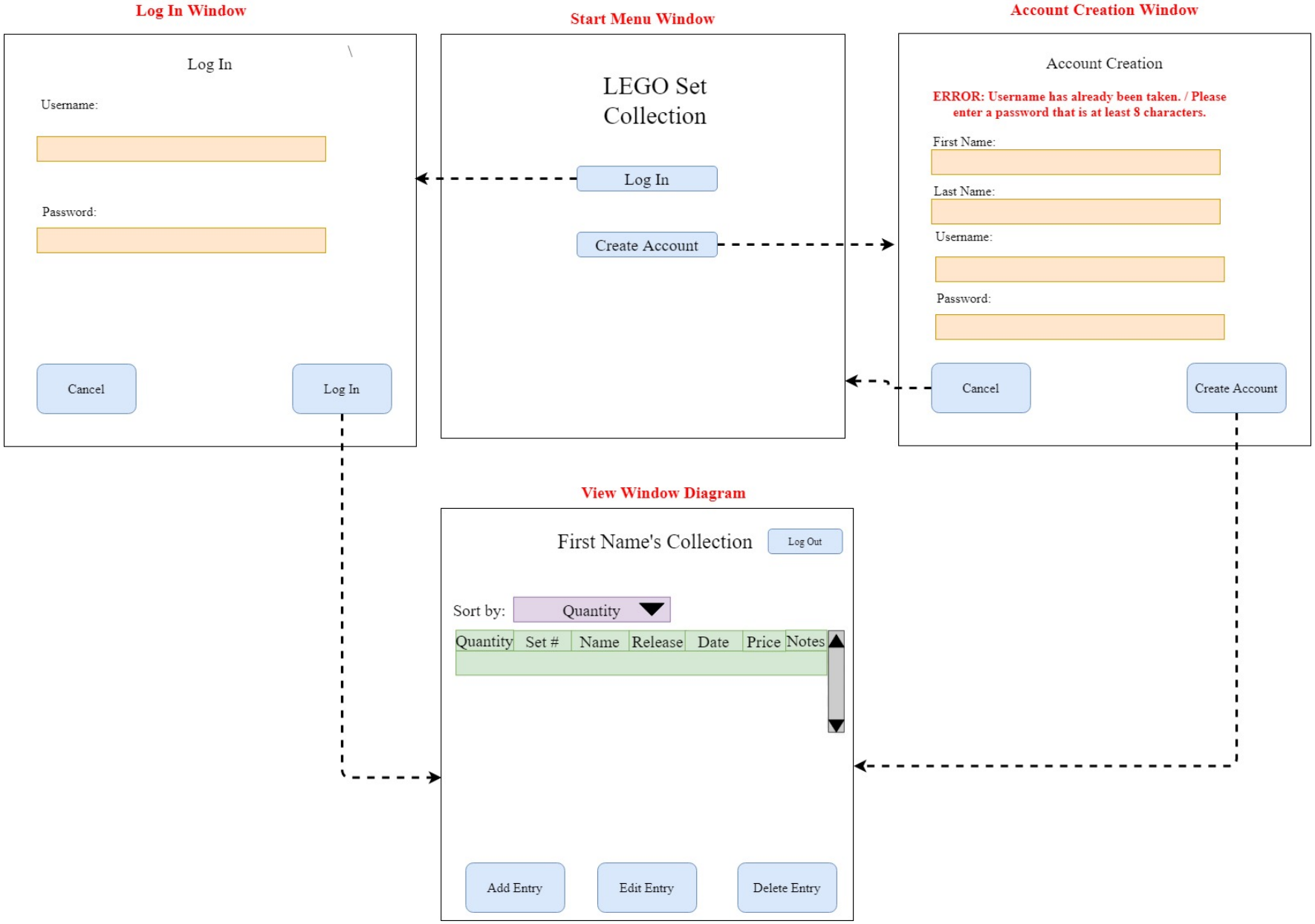
Table of Contents

Complete Diagram 2
Start Menu Diagram 3
Start Menu Flowchart 4
Log In Window Diagram 5
Log In Window Flowchart 6
Account Creation Window 7
Account Creation Flowchart 8
View Window Diagram 9
View Window Flowchart 10
Add Entry Diagram 11
Add Entry Flowchart 12
Edit Entry Diagram 13
Edit Entry Flowchart 14
Delete Entry 15
Delete Entry Flowchart 16
UML Diagram 17
Test Plan 18

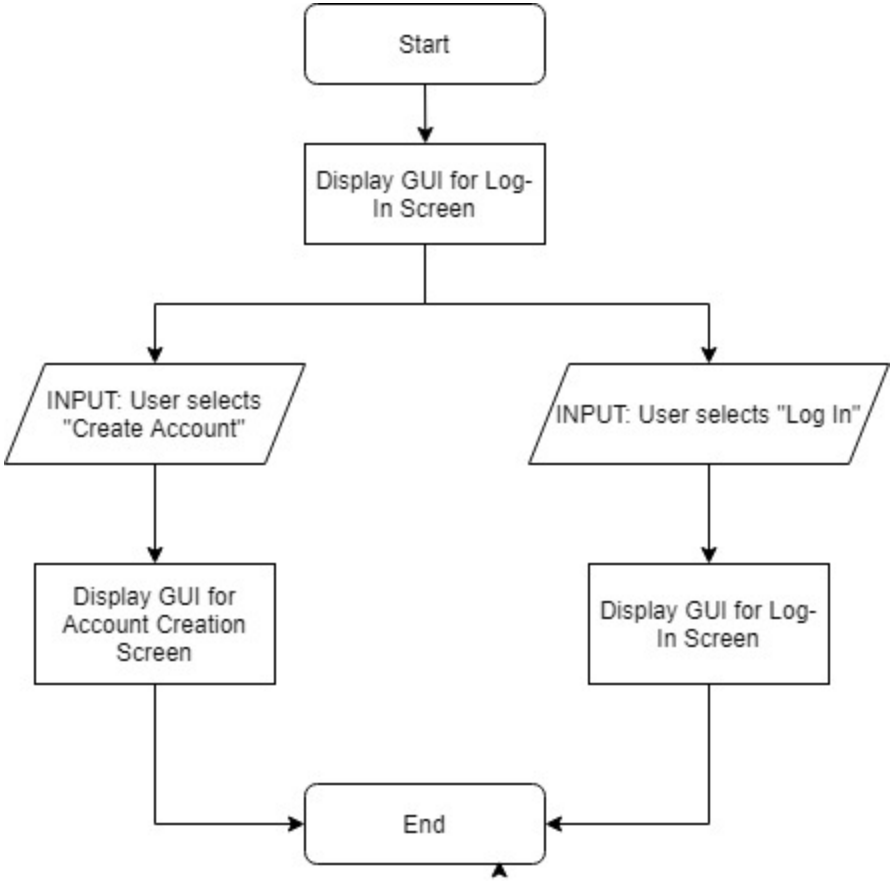
Complete Diagram



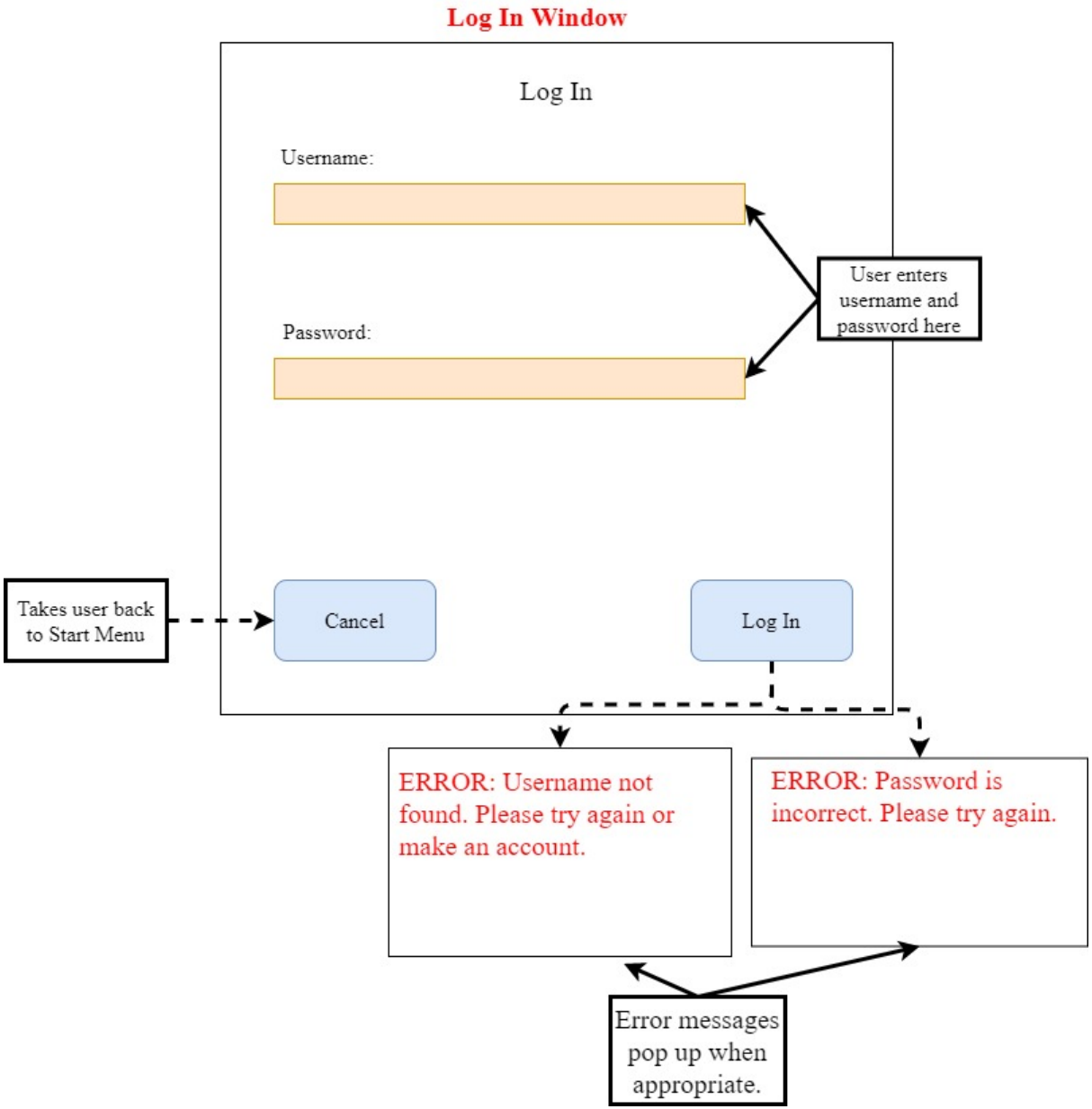
Start Menu Diagram



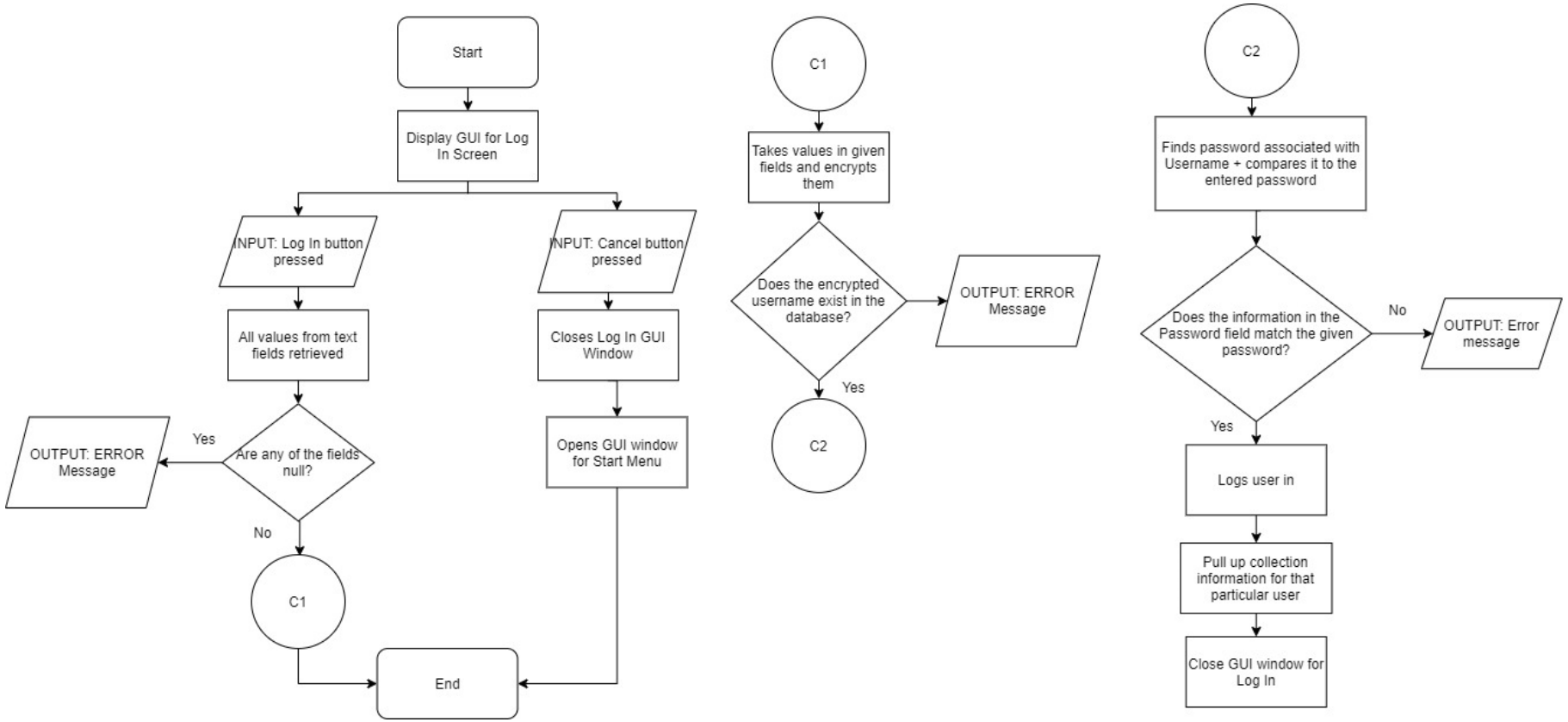
Start Menu Flowchart



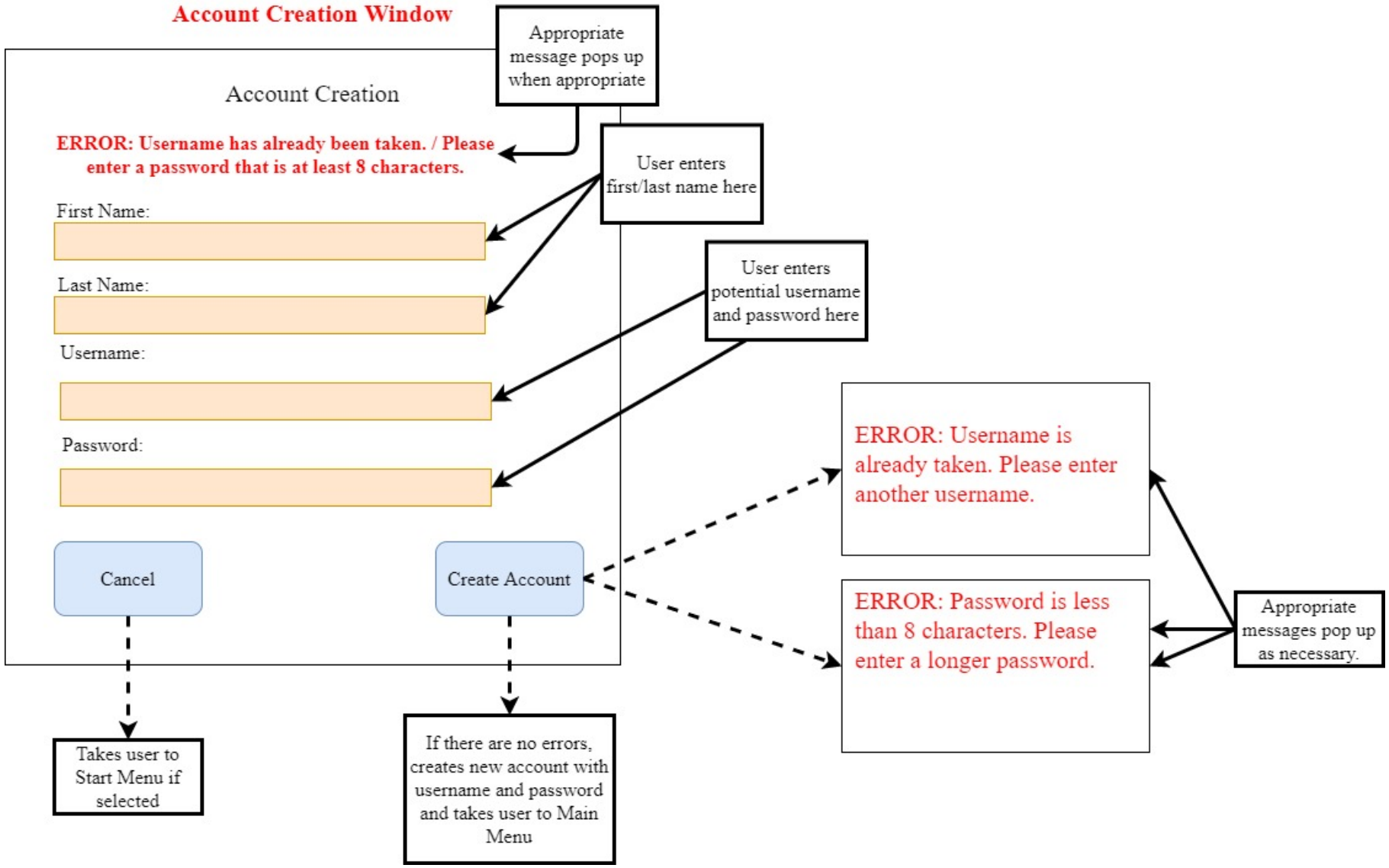
Log In Window Diagram



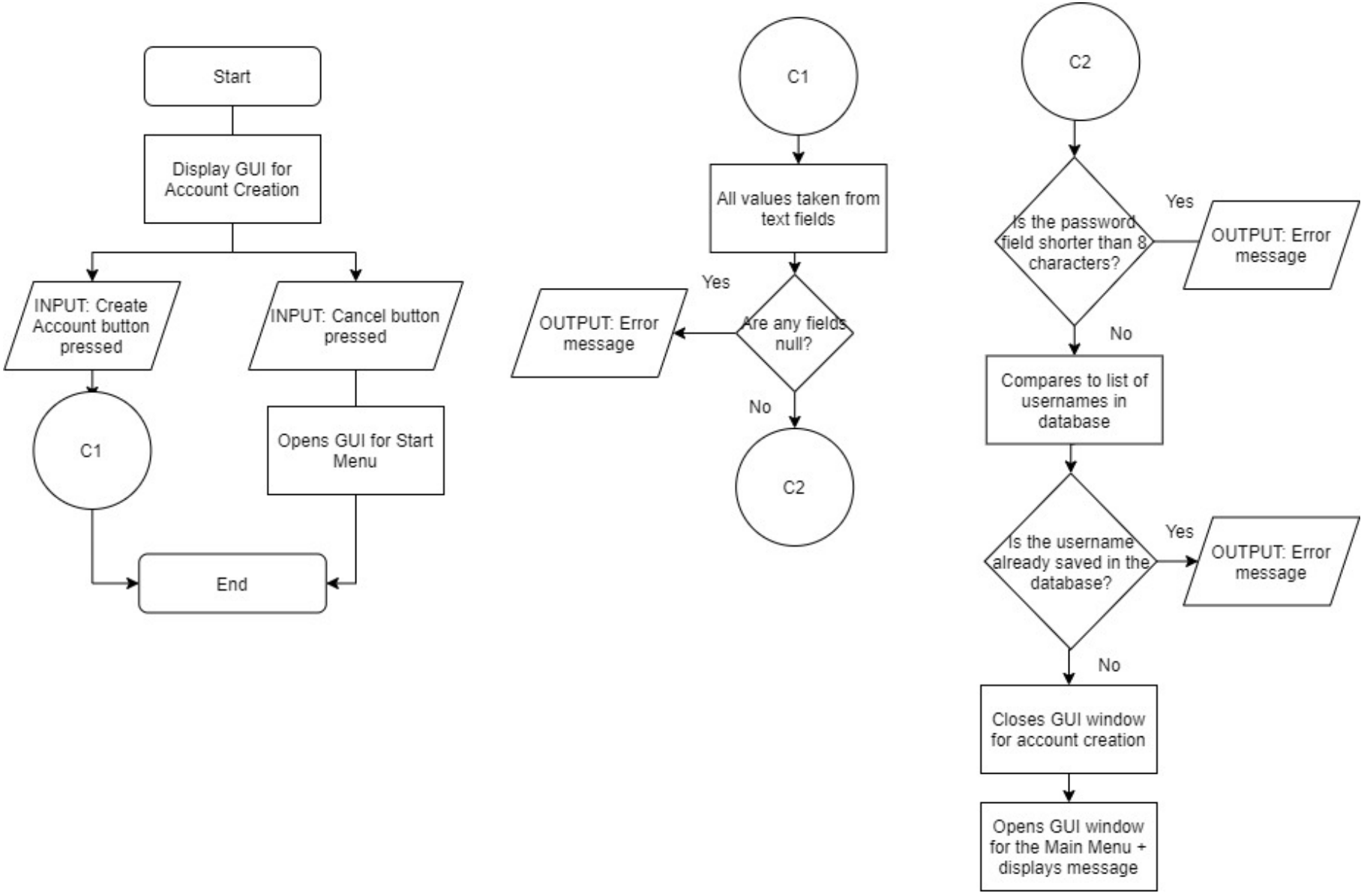
Log In Window Flowchart



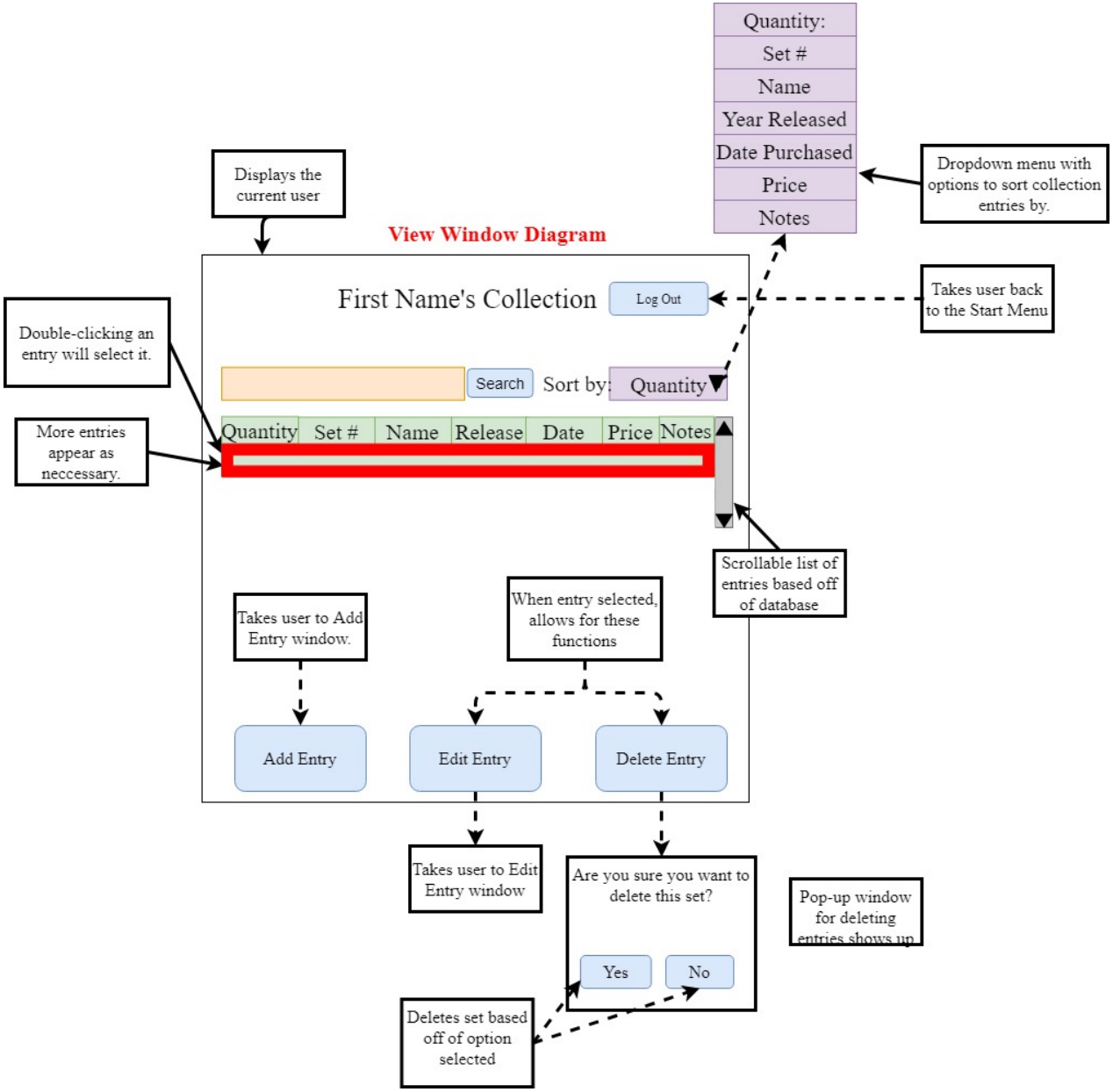
Account Creation Window



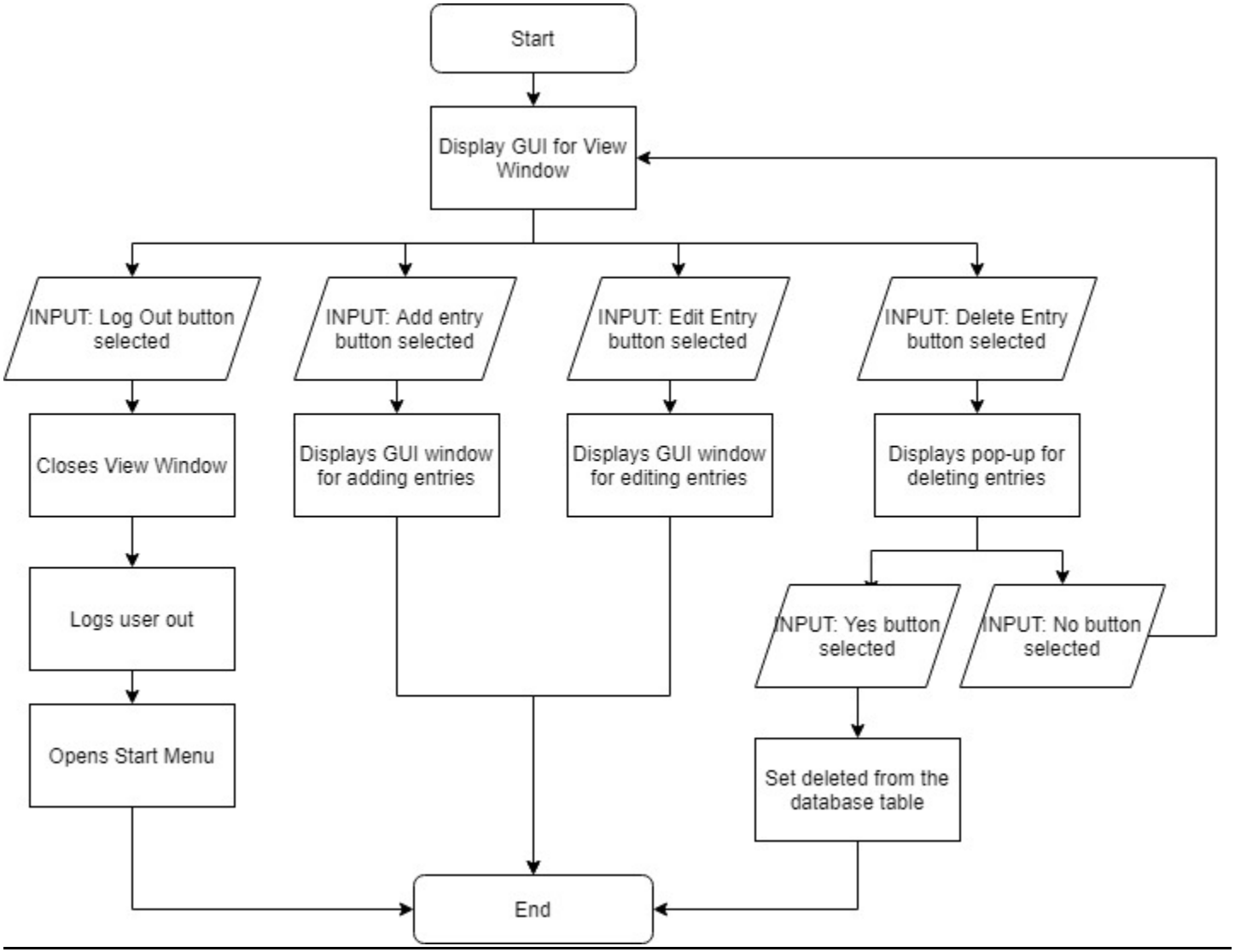
Account Creation Flowchart



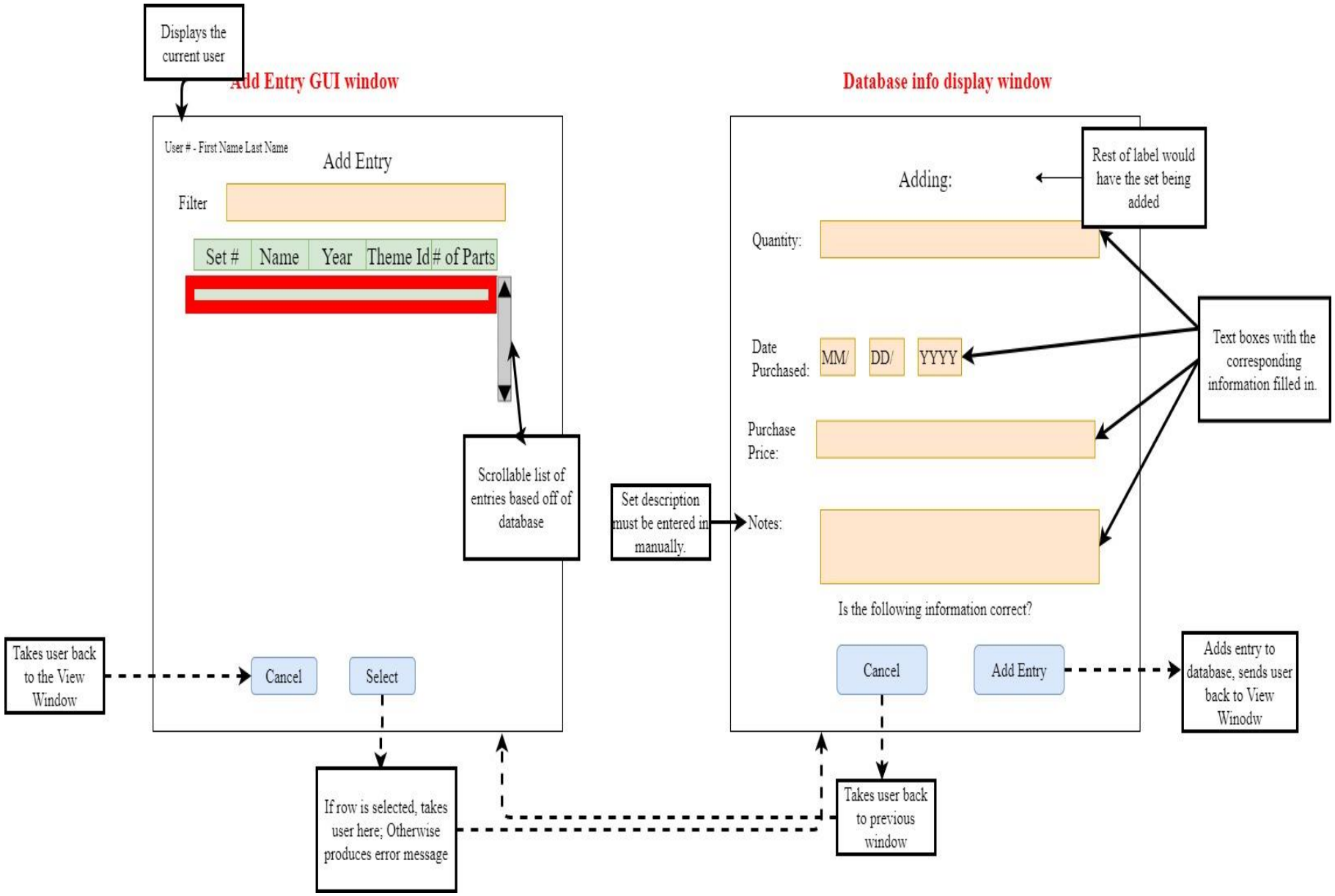
View Window Diagram



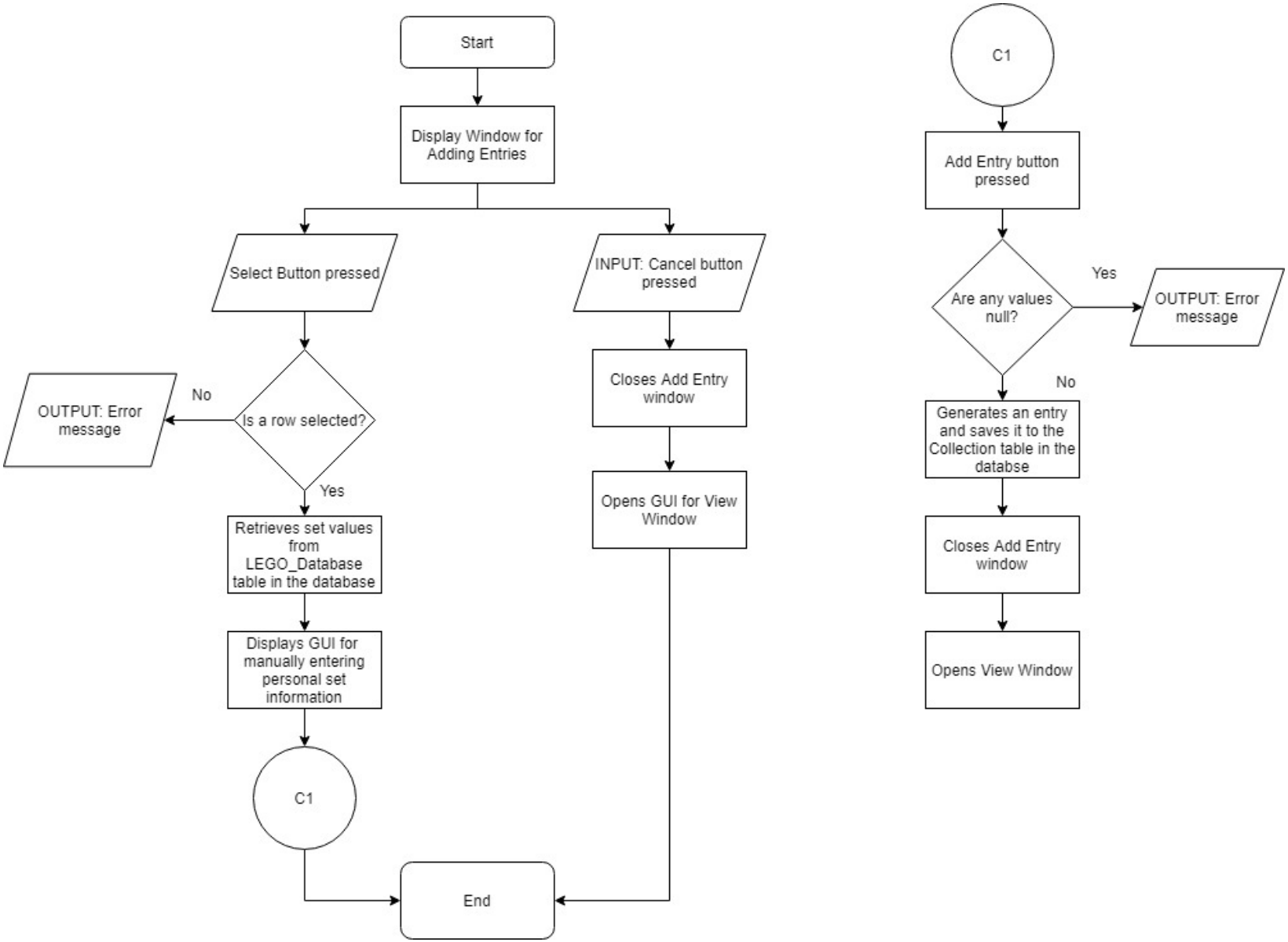
View Window Flowchart



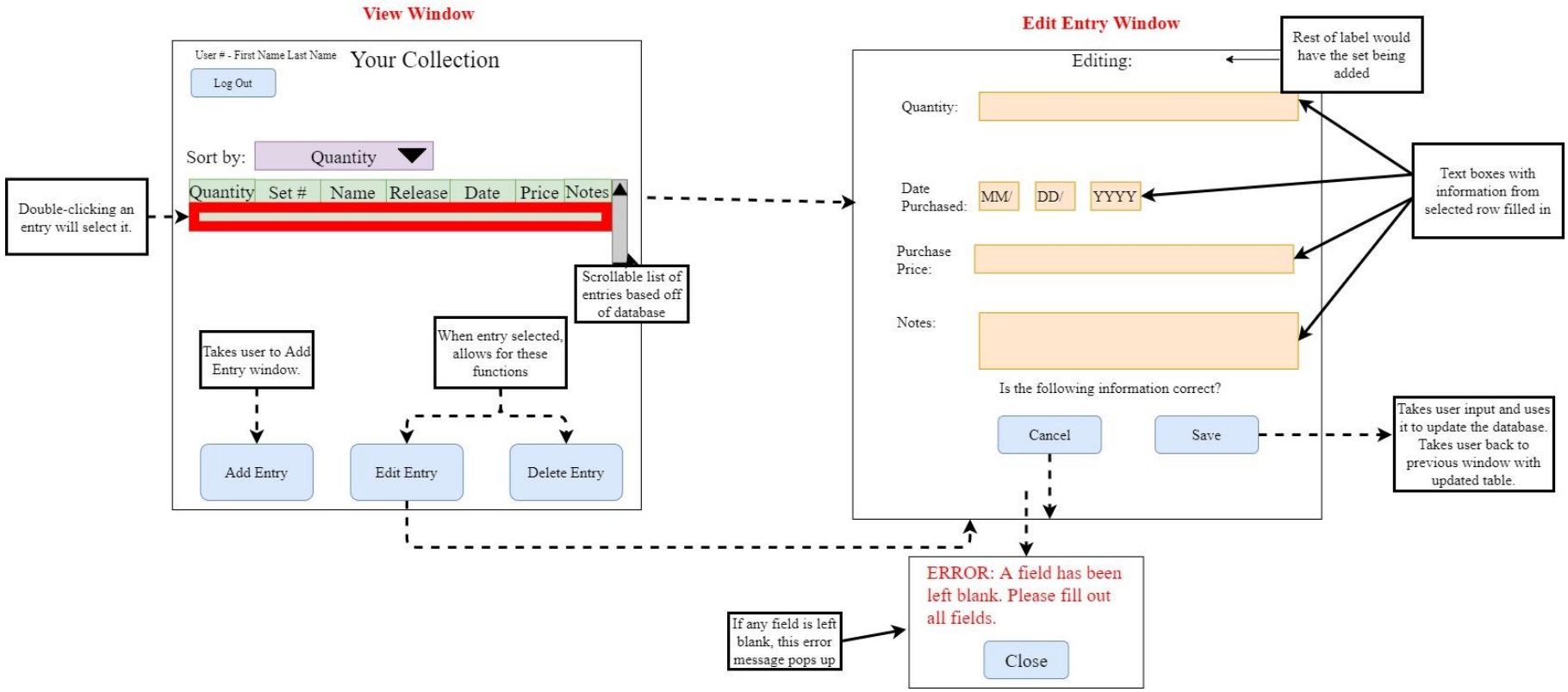
Add Entry Diagram



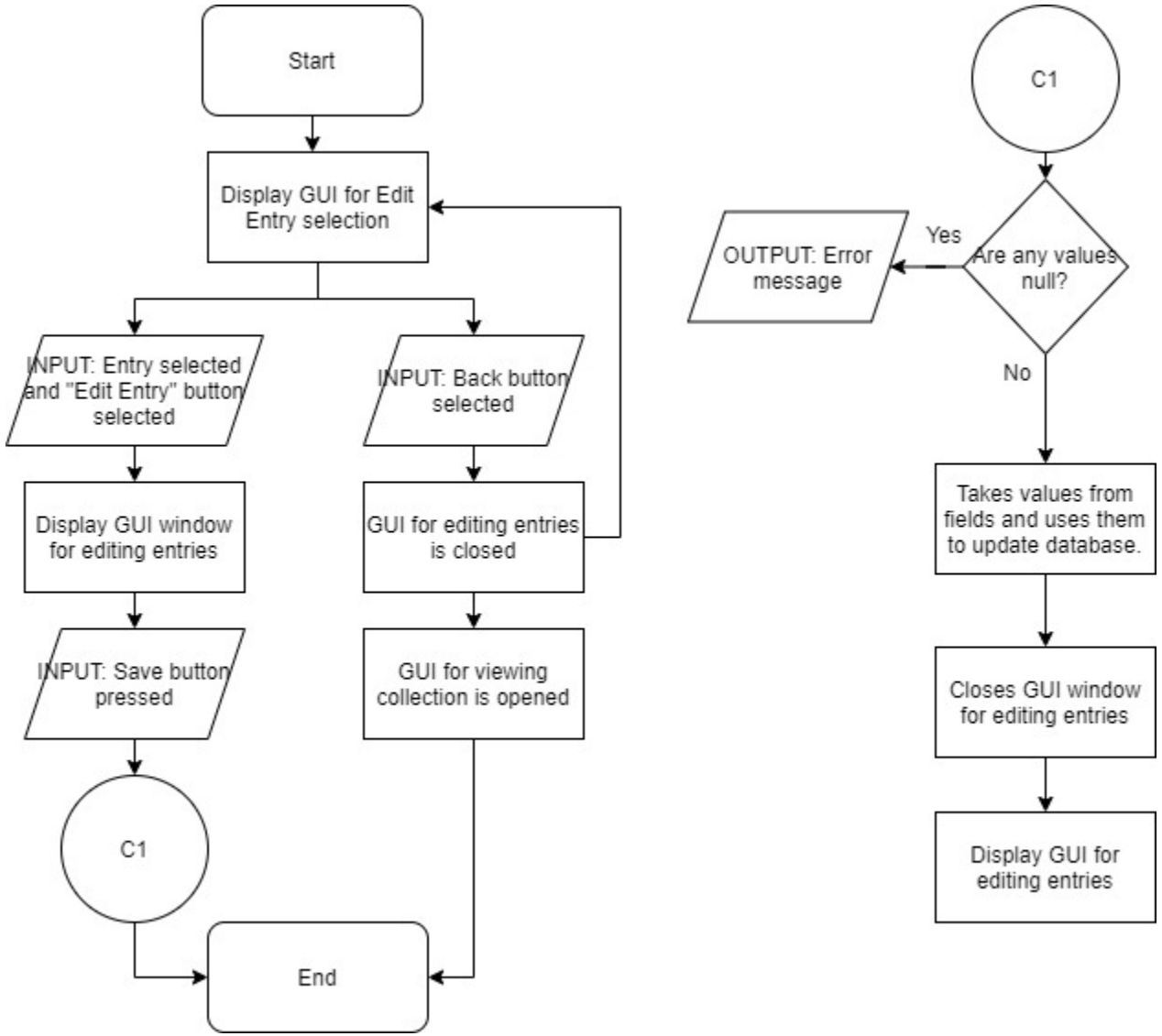
Add Entry Flowchart



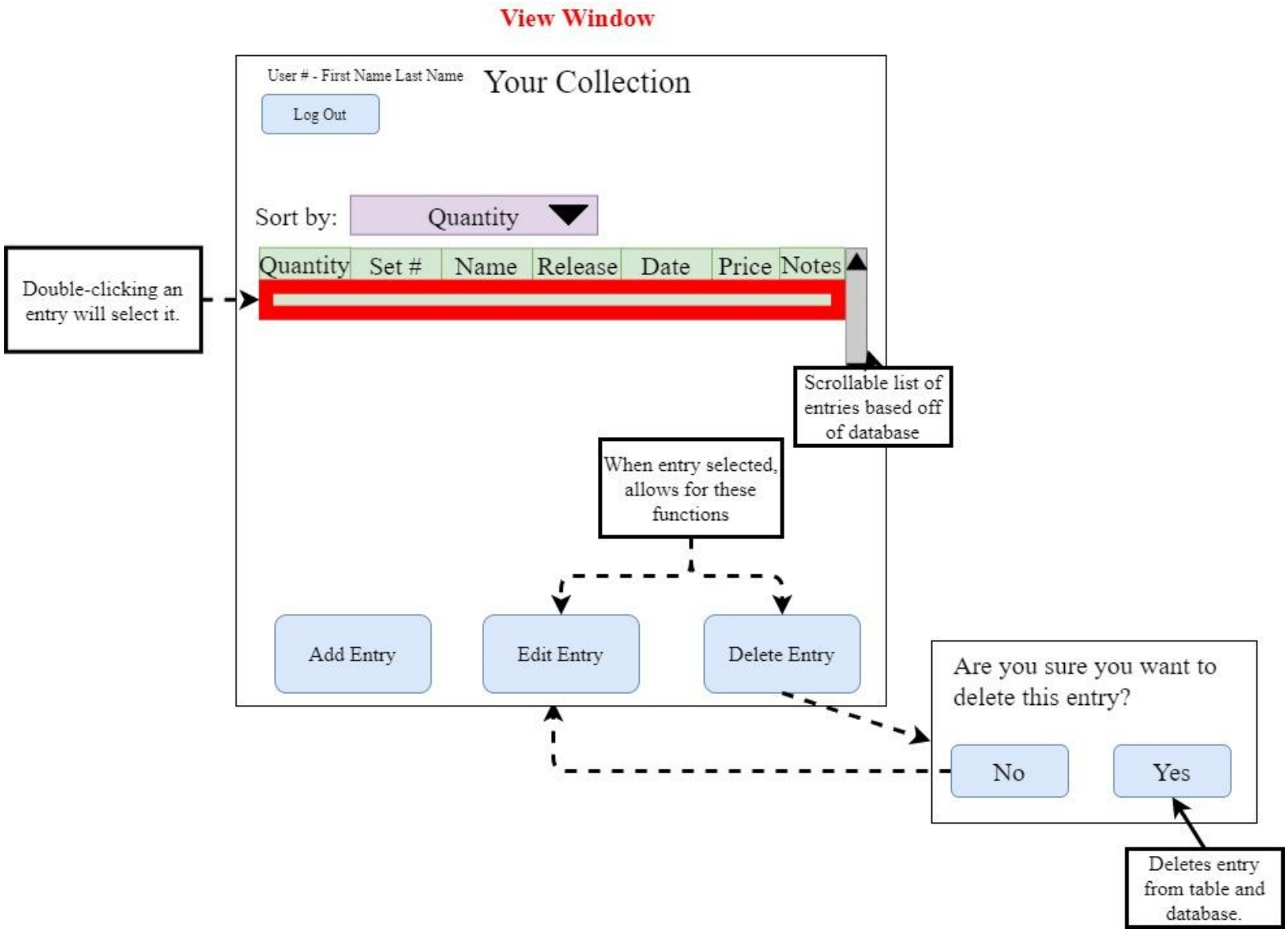
Edit Entry Diagram



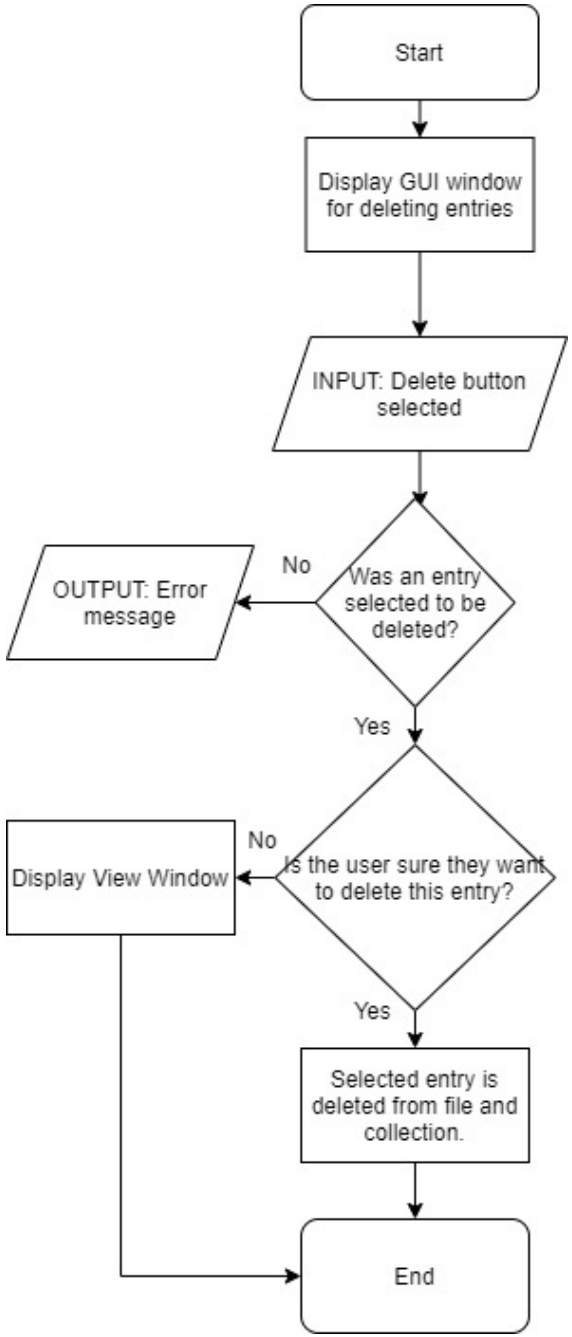
Edit Entry Flowchart



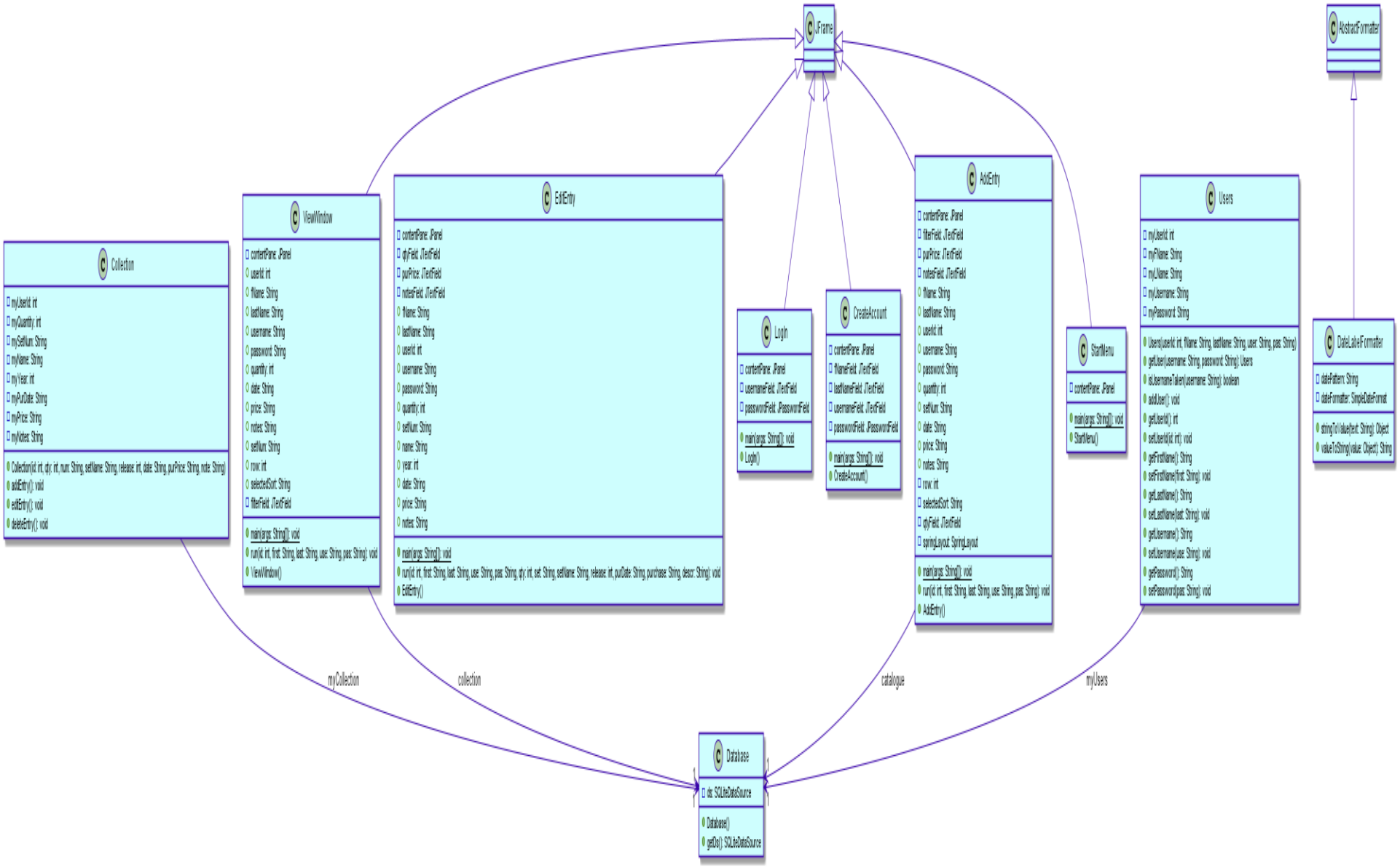
Delete Entry



Delete Entry Flowchart



UML Diagram


JFrame
AbstractFormatter
Collection
myUserId: int
myQuantity: int
mySetNum: String
myName: String
myYear: int
myPurDate: String
myPrice: String
myNotes: String
Collection(id: int, qty: int, num: String, setName: String, release: int, date: String, purPrice: String, note: String)
addEntry(): void
editEntry(): void
deleteEntry(): void
ViewWindow
contentPane: JPanel
userId: int
fName: String
lastName: String
username: String
password: String
quantity: int
date: String
price: String
notes: String
setNum: String
row: int
selectedSort: String
filterField: JTextField
main(args: String[]): void
run(id: int, first: String, last: String, use: String, pas: String): void
ViewWindow()
EditEntry
contentPane: JPanel
qtyField: JTextField
purPrice: JTextField
notesField: JTextField
fName: String
lastName: String
userId: int
username: String
password: String
quantity: int
setNum: String
name: String
year: int
date: String
price: String
notes: String
main(args: String[]): void
run(id: int, first: String, last: String, use: String, pas: String, qty: int, set: String, setName: String, release: int, purDate: String, purchase: String, descr: String): void
EditEntry()
Login
contentPane: JPanel
usernameField: JTextField
passwordField: JPasswordField
main(args: String[]): void
Login()
CreateAccount
contentPane: JPanel
fNameField: JTextField
lastNameField: JTextField
usernameField: JTextField
passwordField: JPasswordField
main(args: String[]): void
CreateAccount()
AddEntry
contentPane: JPanel
filterField: JTextField
purPrice: JTextField
notesField: JTextField
fName: String
lastName: String
userId: int
username: String
password: String
quantity: int
setNum: String
date: String
price: String
notes: String
row: int
selectedSort: String
qtyField: JTextField
springLayout: SpringLayout
main(args: String[]): void
run(id: int, first: String, last: String, use: String, pas: String): void
AddEntry()
StartMenu
contentPane: JPanel
main(args: String[]): void
StartMenu()
Users
myUserId: int
myFName: String
myLName: String
myUsername: String
myPassword: String
Users(userId: int, fName: String, lastName: String, user: String, pas: String)
getUser(username: String, password: String): Users
isUsernameTaken(username: String): boolean
addUser(): void
getUserId(): int
setUserId(id: int): void
getFirstName(): String
setFirstName(first: String): void
getLastName(): String
setLastName(last: String): void
getUsername(): String
setUsername(use: String): void
getPassword(): String
setPassword(pas: String): void
DateLabelFormatter
datePattern: String
dateFormatter: SimpleDateFormat
stringToValue(text: String): Object
valueToString(value: Object): String
Database
ds: SQLiteDataSource
Database()
getDs(): SQLiteDataSource
myCollection
collection
catalogue
myUsers

Test Plan

Action to Test	Test Procedure	Success Criteria
The program should allow the user to log in and out of the GUI program.	Log in and out of the program.	The user can log in and out of the program successfully.
The program should be able to retain all data when the user logs in and out.	Log in and out of the program. The user's collection table should be maintained.	All data is maintained correctly in the user's collection table.
The program should allow users to create new accounts.	Create an account and see if its information is stored in the account file.	The user should be able to log in and out of their account, and view their collection in a new account.

The program should be able to store information from different users in the correct account.	Add an entry to one user's collection. Log out and check another user's account to confirm that the added set is not present.	The collections of each user only contain sets that that user added.
The program should allow the user to add entries.	Add an entry to the collection. Check the collection table in the View Window.	The correct entry pops up and the new entry is successfully added to the collection file and Viewing Table.
The program should allow the user to delete entries.	Delete an entry. Check the collection table in the View Window.	The selected entry has been deleted from the user's collection table in the View Window.
The program should allow the user to edit entries.	Edit an entry. Check the collection table in the View Window.	The edited entry's new information is correctly displayed in the user's collection table in the View Window.

The program should have search bar functionality that connects to the catalogue.	Enter a value in the filter box. See if the search bar filters results from the catalogue appropriately.	All sets with names that contain the keyword should appear and the appropriate information should be pulled into its individual entry.
The program should allow the user to sort the information in their collection by the column headers of the collection table.	Click on a column header to sort by. See if the sets are sorted appropriately.	The sets in the collection should be sorted by the selected column header in ascending order.