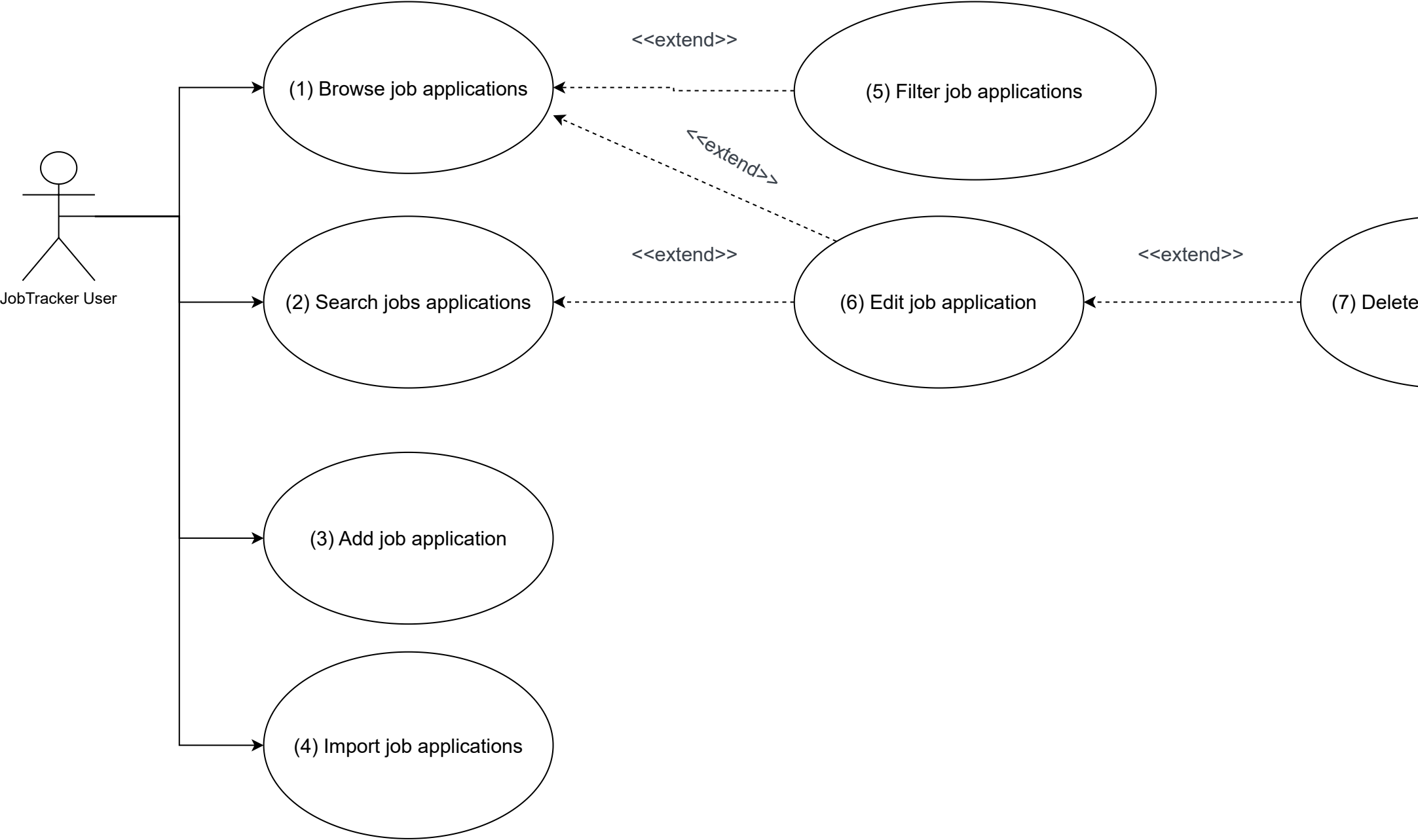
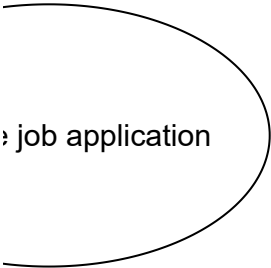


JobTracker Program Use Case Diagram



UC: Browse job applications (Ref # 1)		UC: Search jobs applications (Ref # 2)	
	The user can look and peruse through all of their job applications		1



rch job applications (Ref # 2)
The user can search by company, location, or role to find the

Overview	The user can look and peruse through all of their job applications within the tracker program	Overview	
Related use cases (if any):	<ul style="list-style-type: none"> - Extended by (5. Filter job applications) since the user can filter all of the job applications within the browse by a desired status - Extended by (6. Edit job application) because you can choose the job that you want to edit (or possibly delete) only after looking through them first 	Related use cases (if any):	<ul style="list-style-type: none"> - (- (
Actors	"JobTracker" program user	Actors	
Pre Conditions (Optional)	-----	Pre Conditions (Optional)	
Post Conditions (Optional)	-----	Post Conditions (Optional)	
UC: Add job application (Ref # 3)		UC: Imp	
Overview	The user can add a new job application to the program in order to track it for the future	Overview	T
Related use cases (if any):	-----	Related use cases (if any):	
Actors	"JobTracker" program user	Actors	
Pre Conditions (Optional)	-----	Pre Conditions (Optional)	
Post Conditions (Optional)	Job tracker now contains the new job application to track with the inputted information	Post Conditions (Optional)	
UC: Filter job applications (Ref # 5)		UC: E	
Overview	The user can filter out the job applications with a desired status out of all the applications tracked with the system	Overview	T

The user can search by company, location, or role to find the job applications they are querying for.
- Extended by (6. Edit job application) because you can choose the job that you want to edit (or possibly delete) only after looking through them first
"JobTracker" program user

Import job applications (Ref # 4)
The user can import a list of job applications into the program from a .csv file

"JobTracker" program user
The directory contains a .csv file with the inputted file name
Job tracker now contains all the job applications that were listed in the .csv file
Edit job application (Ref # 6)
The user can edit a tracked job application in order to update its information with the program
- Extends (2. Search job applications) because editing a

Related use cases (if any):	- Extends (1. Browse job applications) because a user can only filter out their desired applications if they are first browsing them	Related use cases (if any):	
Actors	"JobTracker" program user	Actors	
Pre Conditions (Optional)	-----	Pre Conditions (Optional)	
Post Conditions (Optional)	-----	Post Conditions (Optional)	S
UC: Delete job application (Ref # 7)			
Overview	The user can choose to delete a tracked job application from the system		
Related use cases (if any):	- Extends (6. Edit job application) because the user only has the option to delete a tracked job application after they choose to edit it		
Actors	"JobTracker" program user		
Pre Conditions (Optional)	-----		
Post Conditions (Optional)	Selected tracked job application is now deleted from the system		

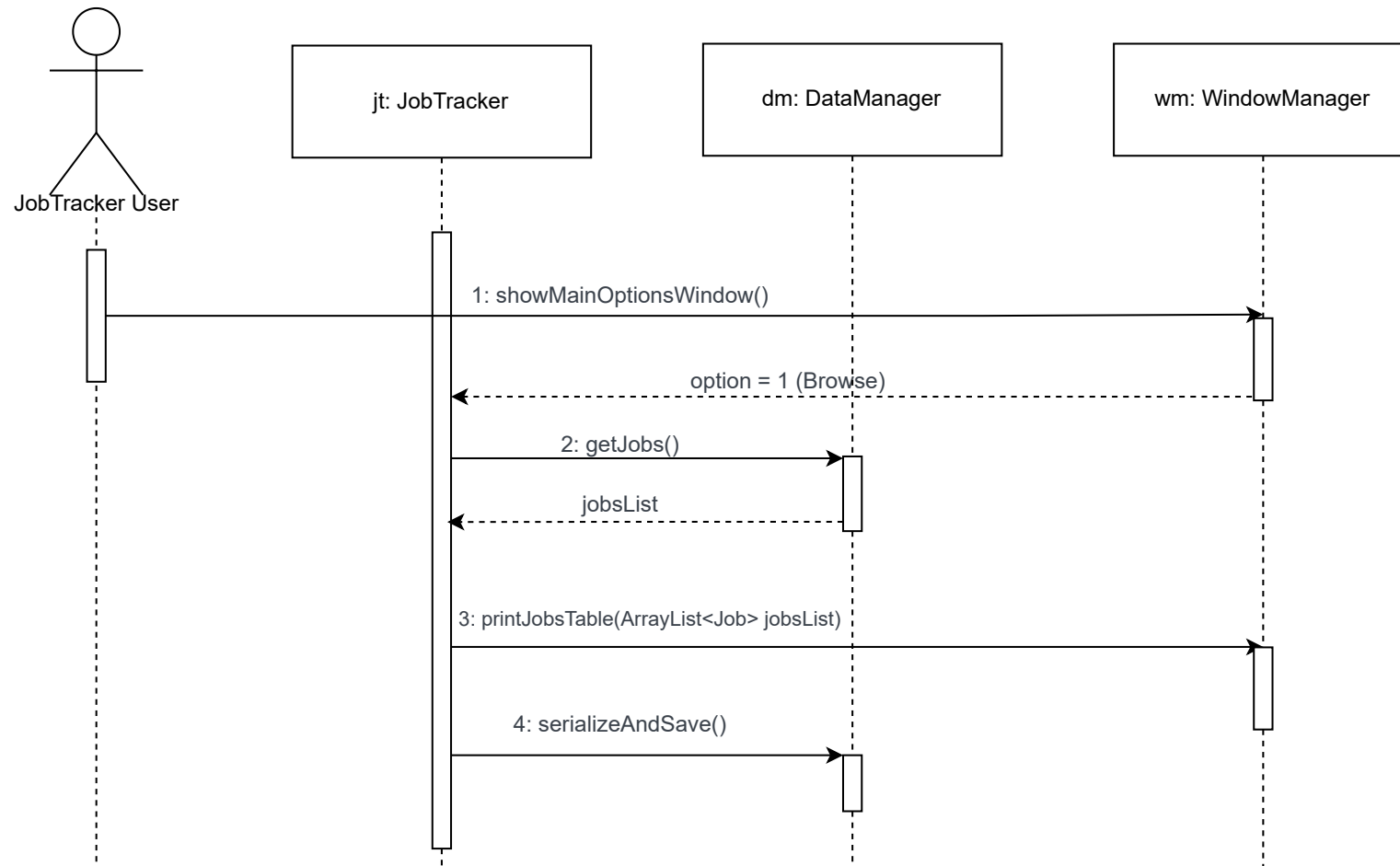
job application (or deleting it) becomes an option after only after looking through them first

- Extended by (7. Delete job application) since the user only has the option to delete a tracked job application after they choose to edit it

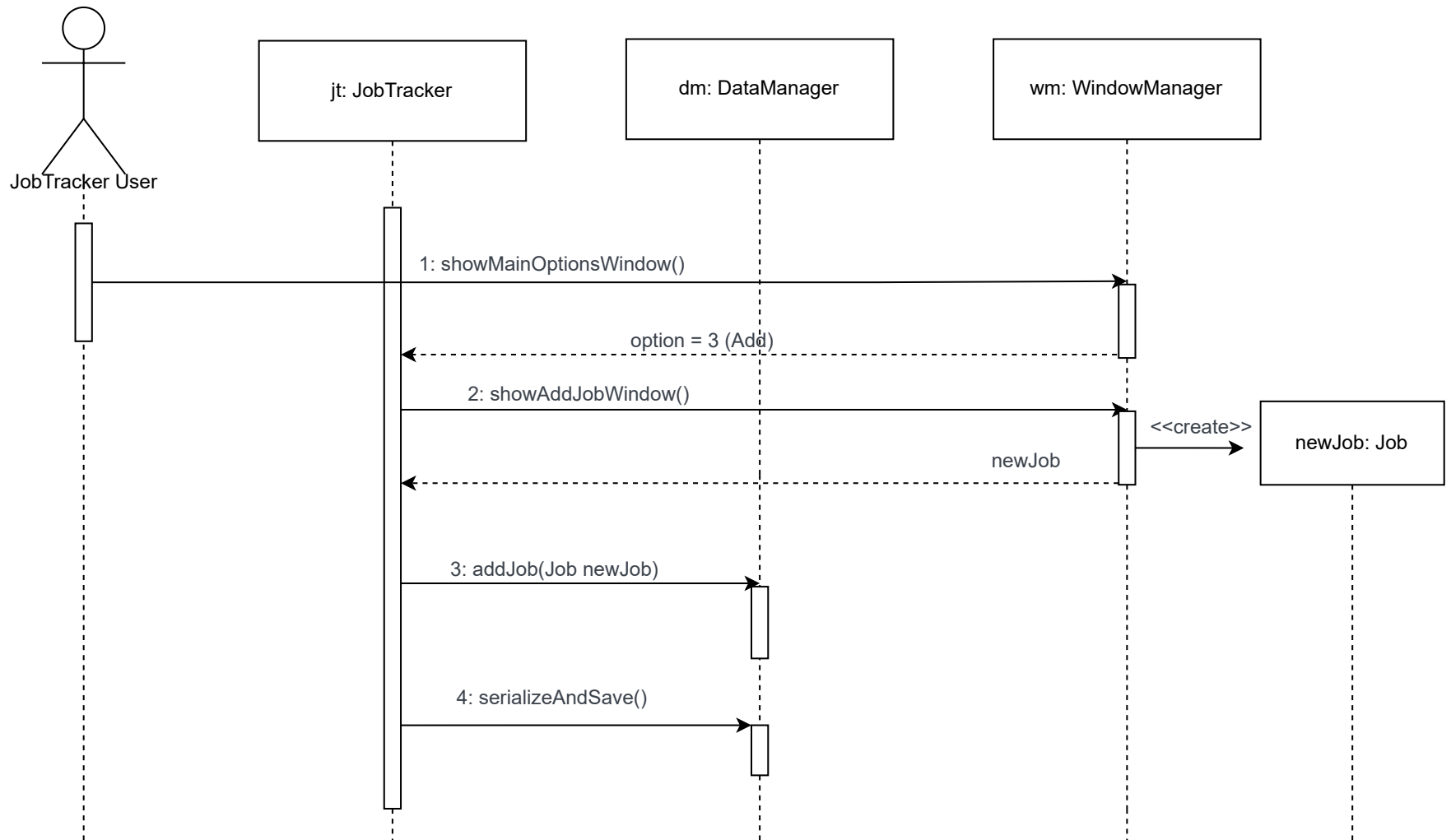
"JobTracker" program user

Selected tracked job application now has updated information

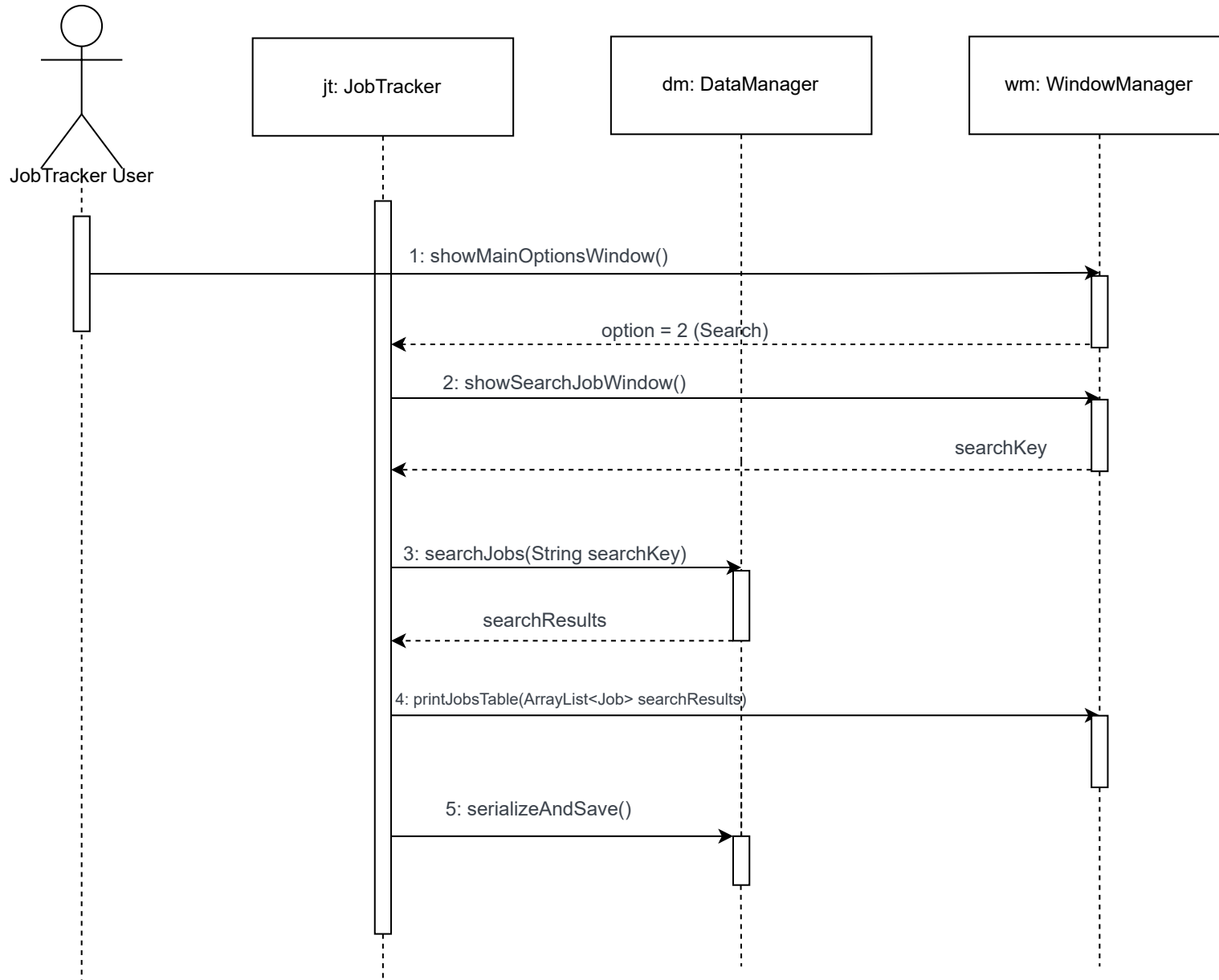
JobTracker Program Sequence Diagram: Browse



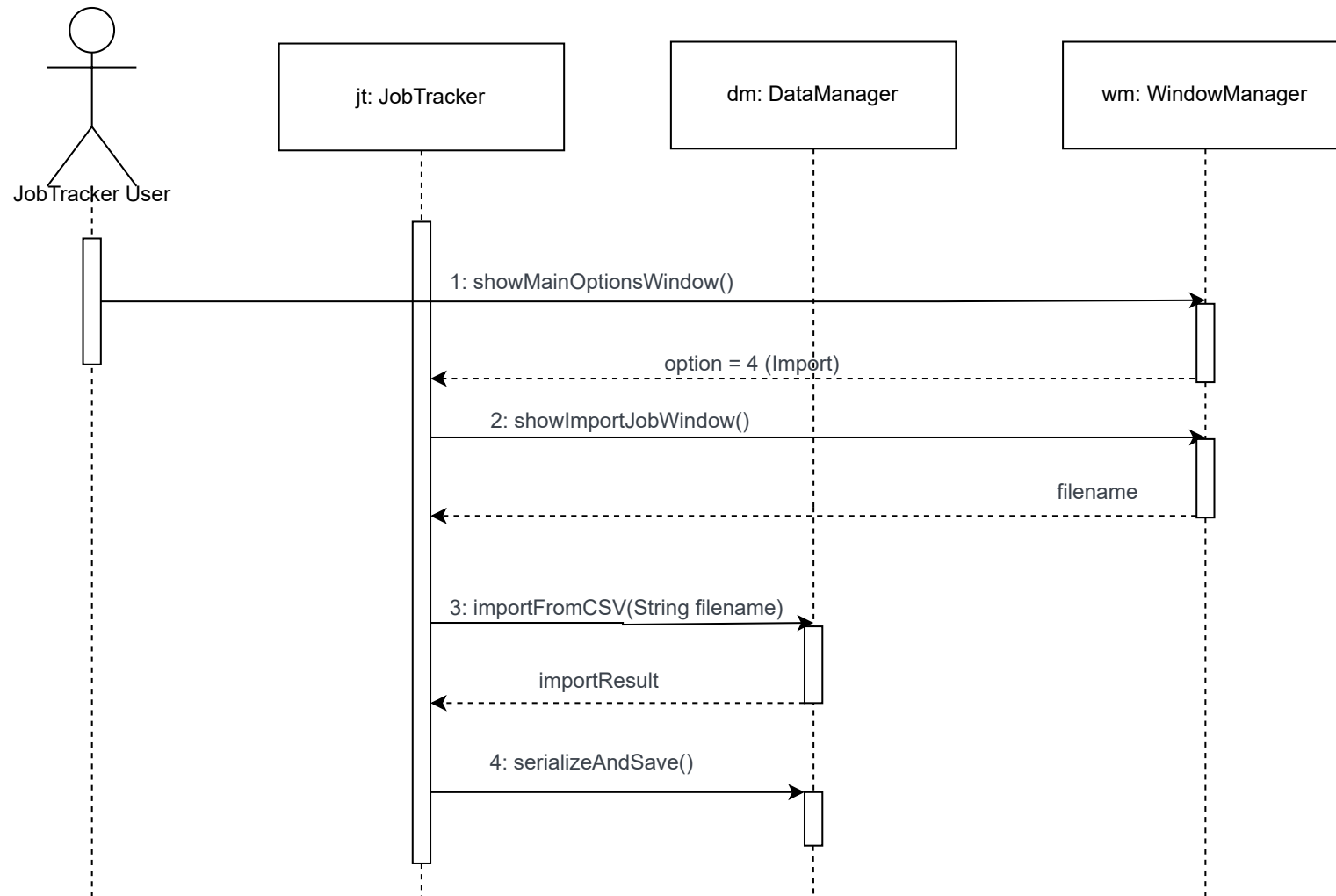
JobTracker Program Sequence Diagram: Add



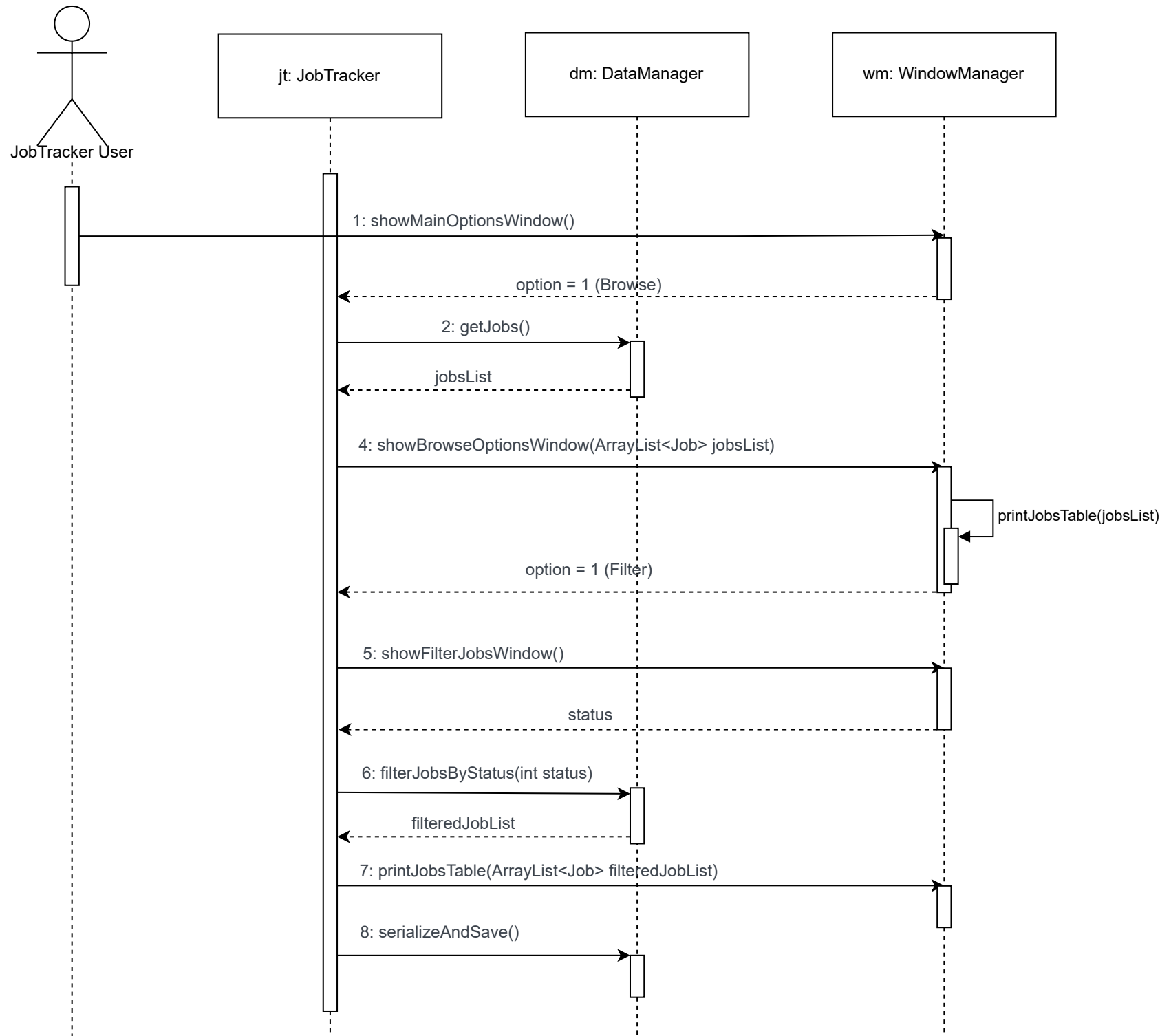
JobTracker Program Sequence Diagram: Search



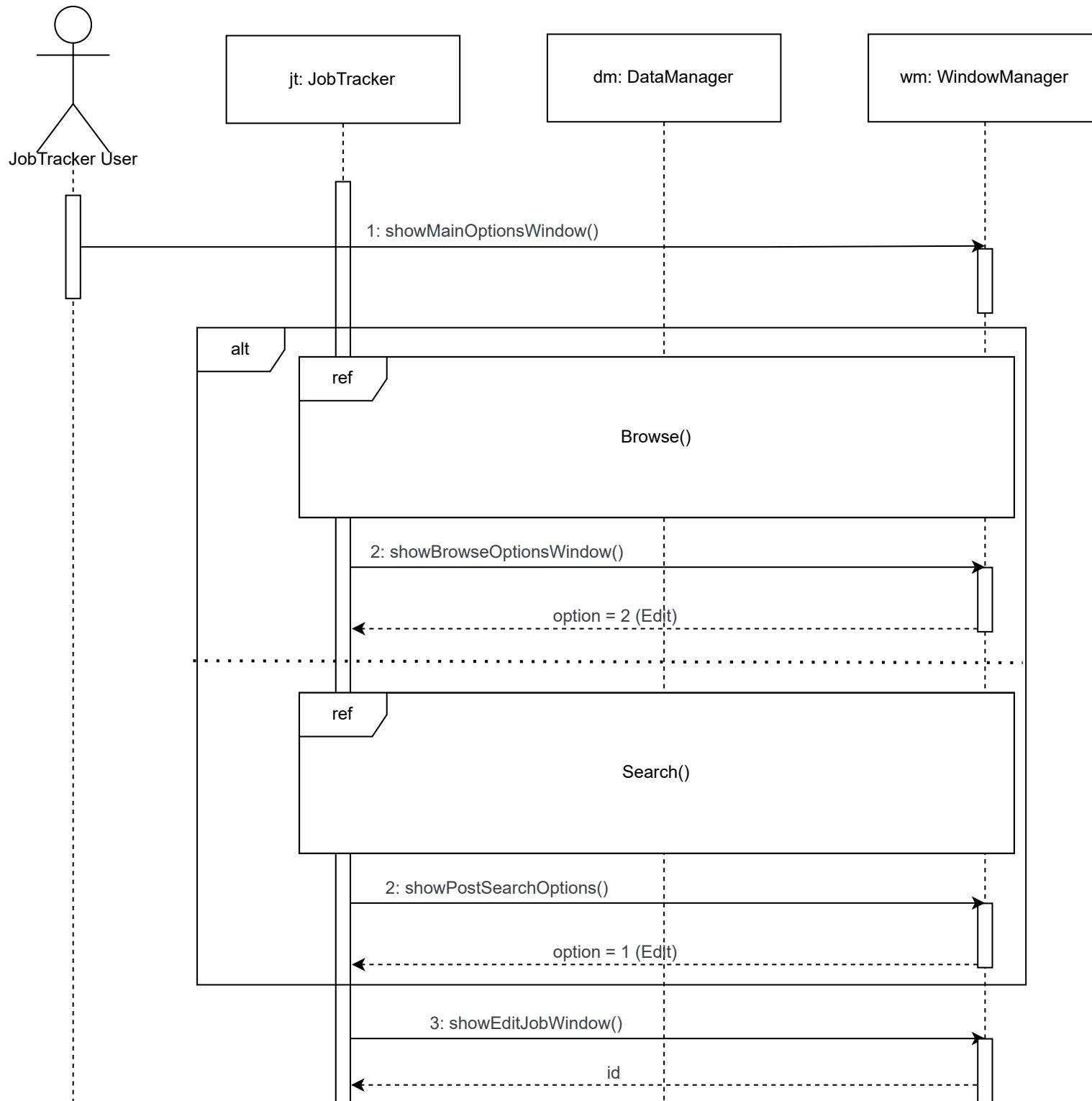
JobTracker Program Sequence Diagram: Import

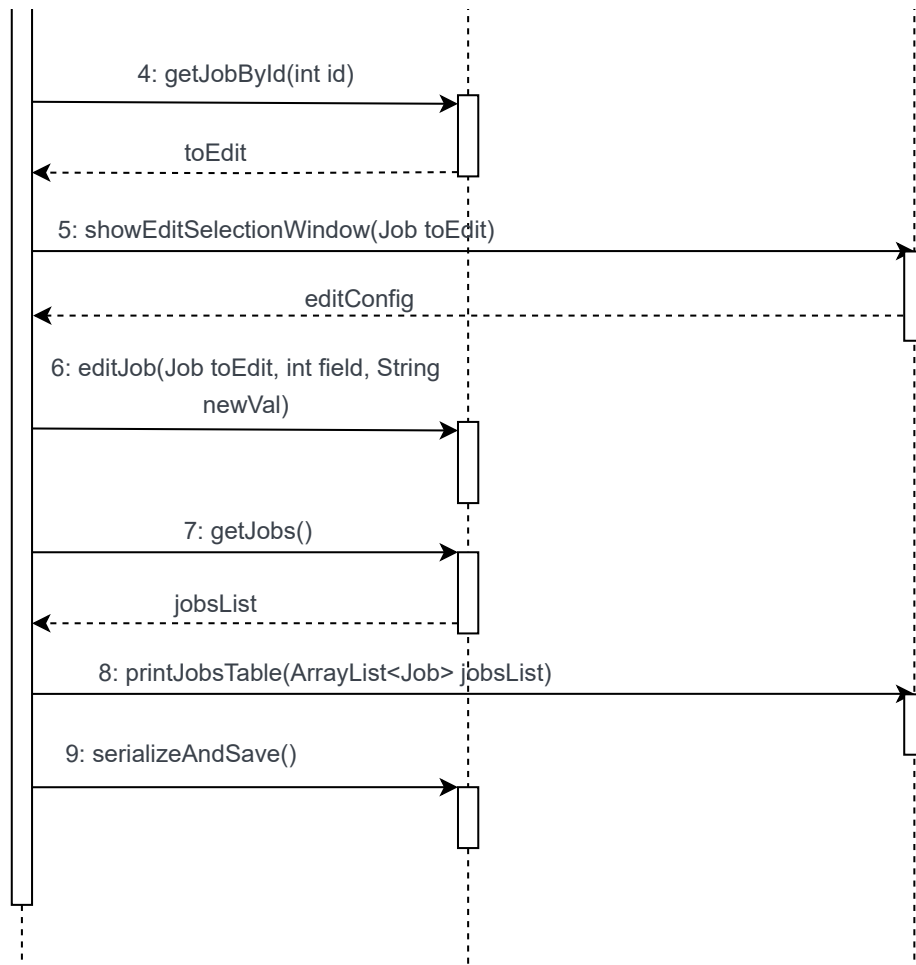


JobTracker Program Sequence Diagram: Filter

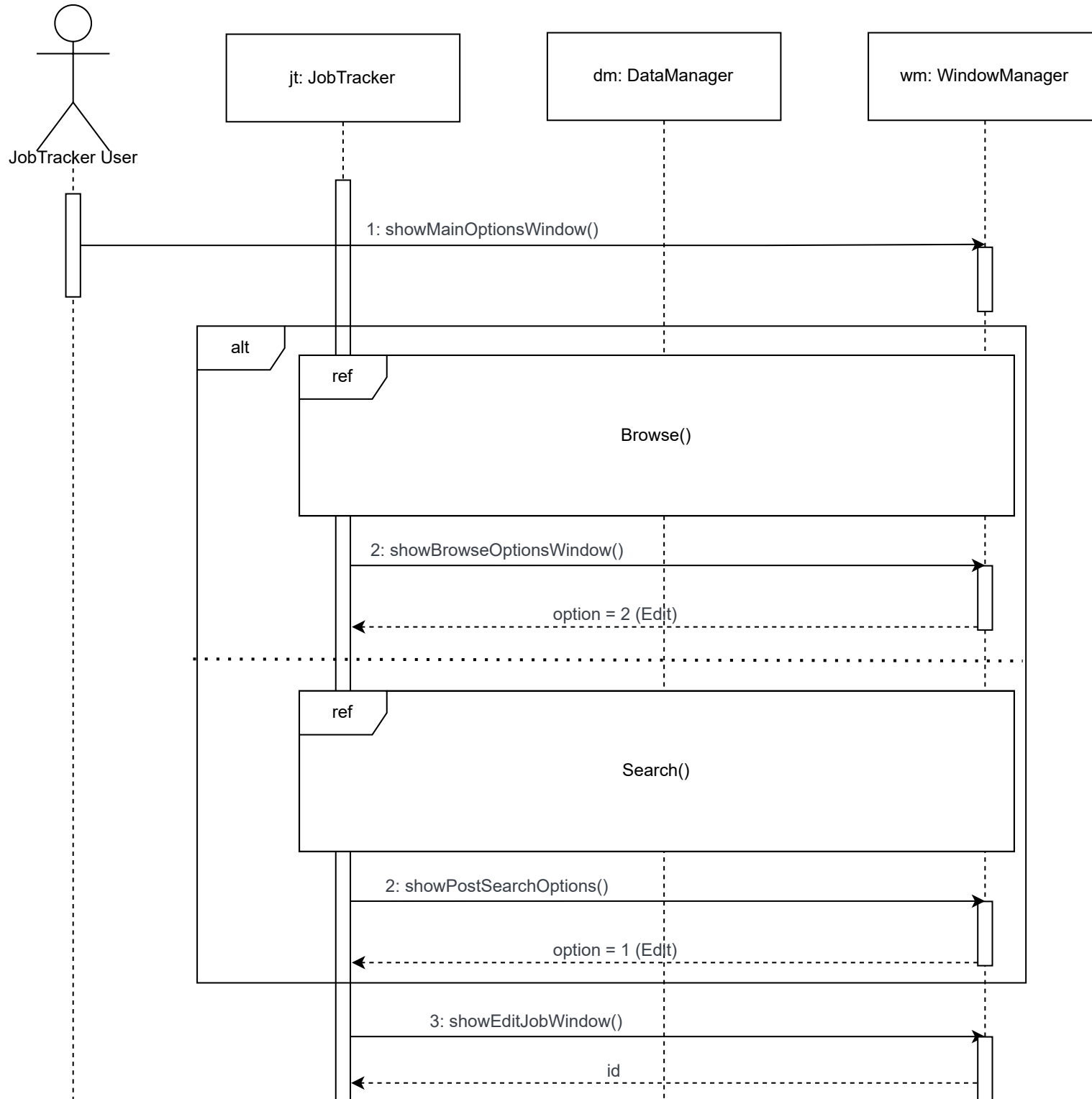


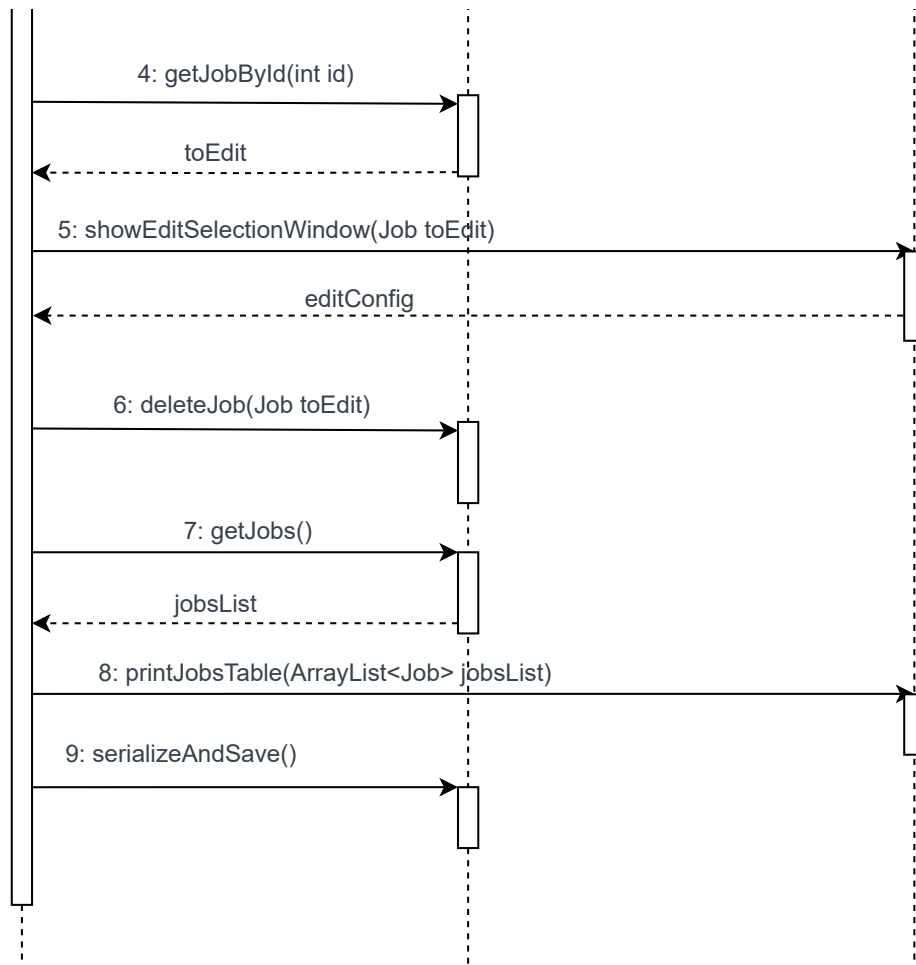
JobTracker Program Sequence Diagram: Edit



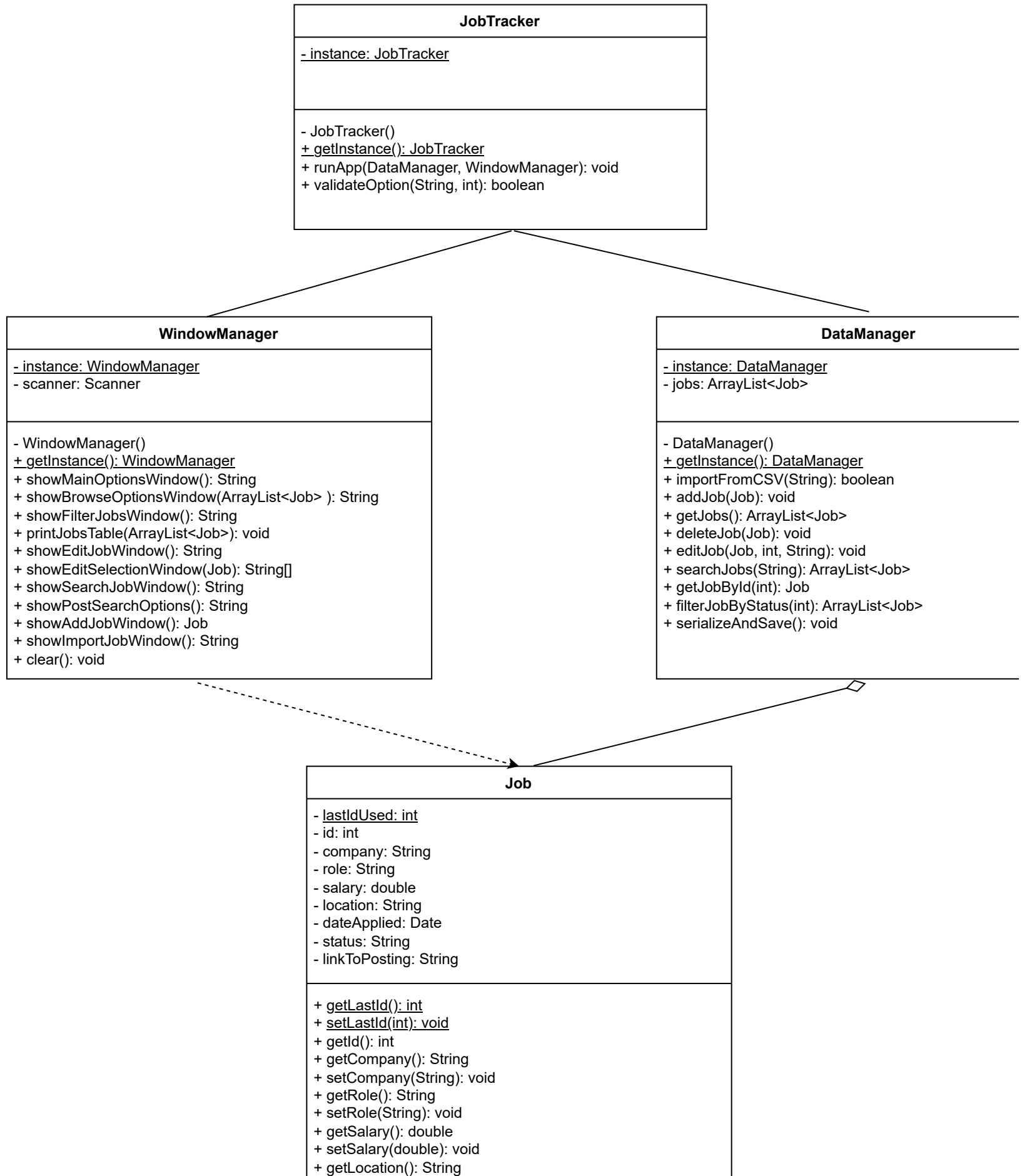


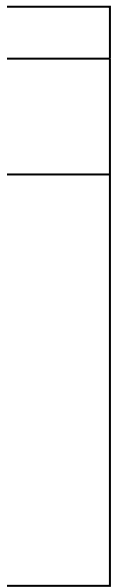
JobTracker Program Sequence Diagram: Delete





JobTracker Program Class Diagram





```
+ setLocation(String): void
+ getDateApplied(): Date
+ getDateAppliedFormatted(): String
+ setDateApplied(Date): void
+ getStatus(): String
+ setStatus(String): void
+ getLinkToPosting(): String
+ setLinkToPosting(String): void
+ toString(): void
```

