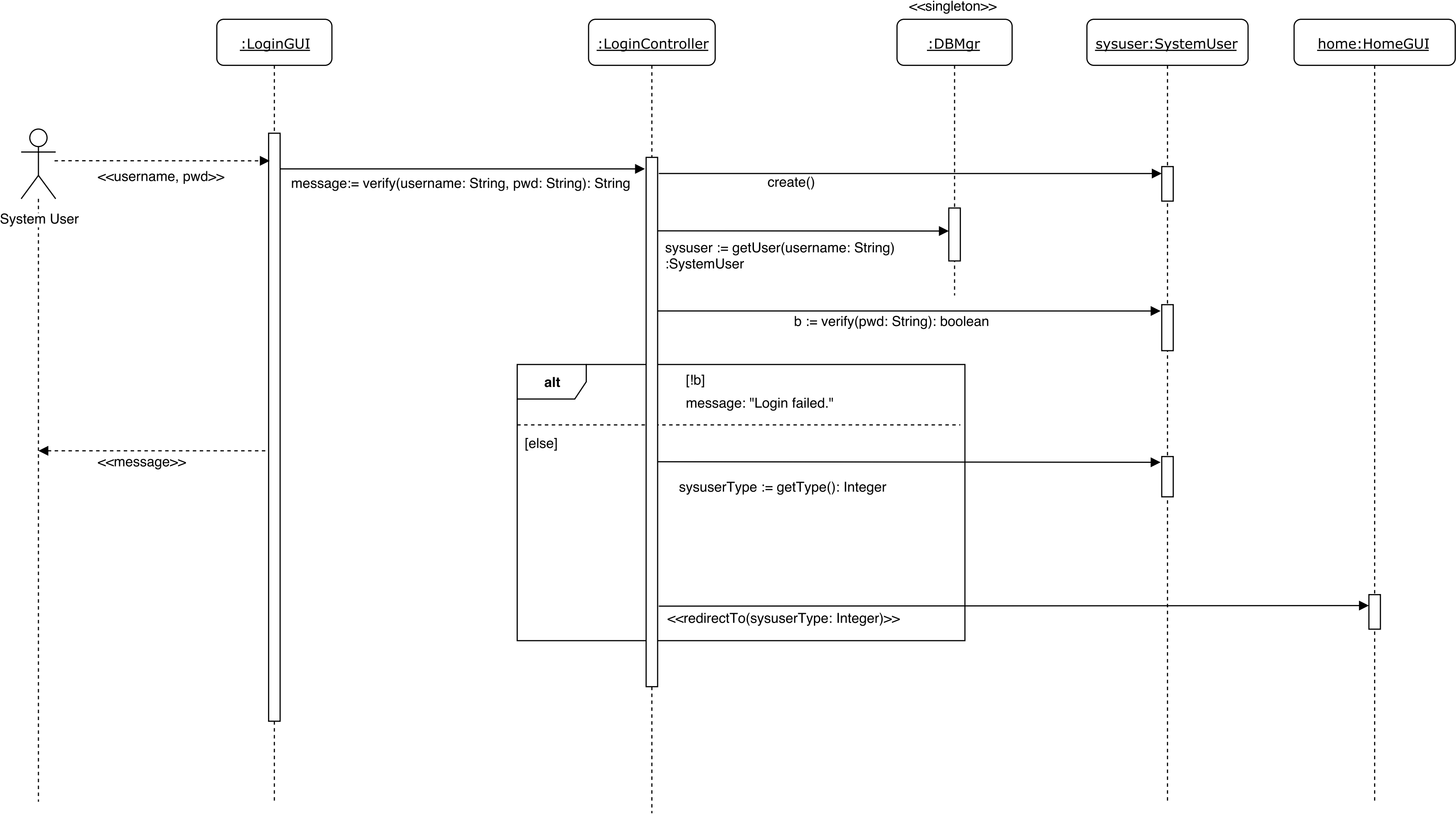
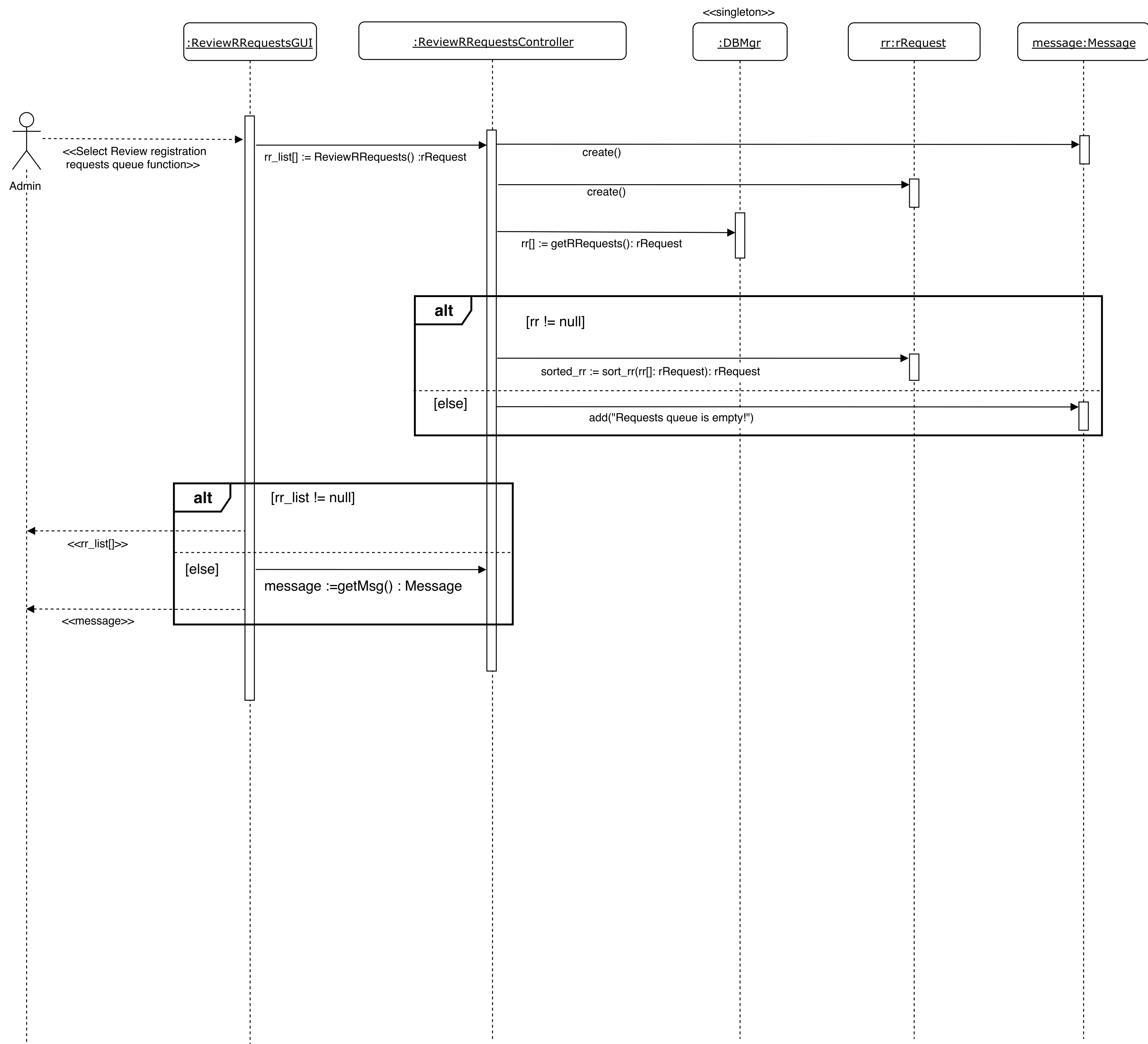


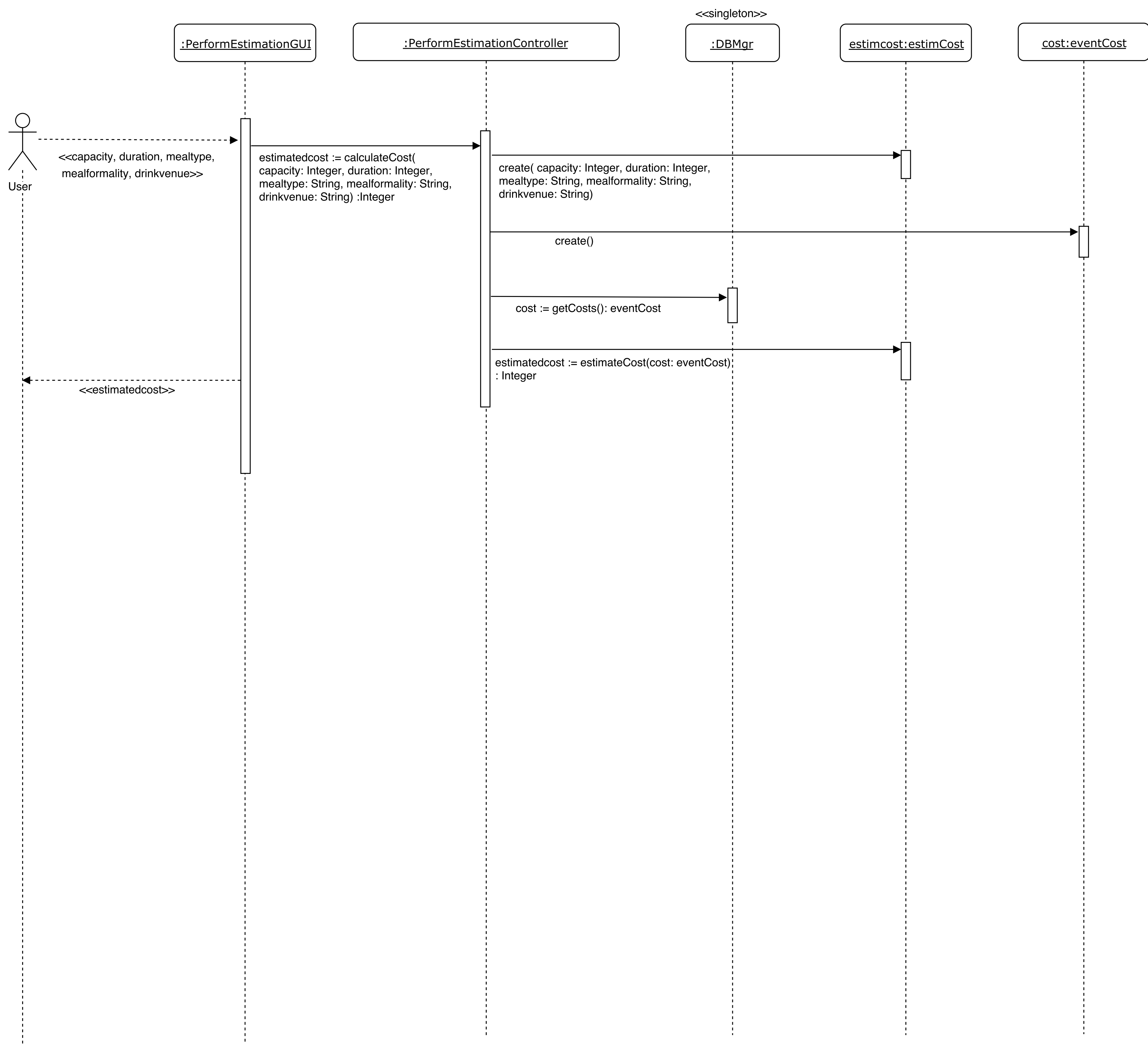
Sequence Diagram : Login



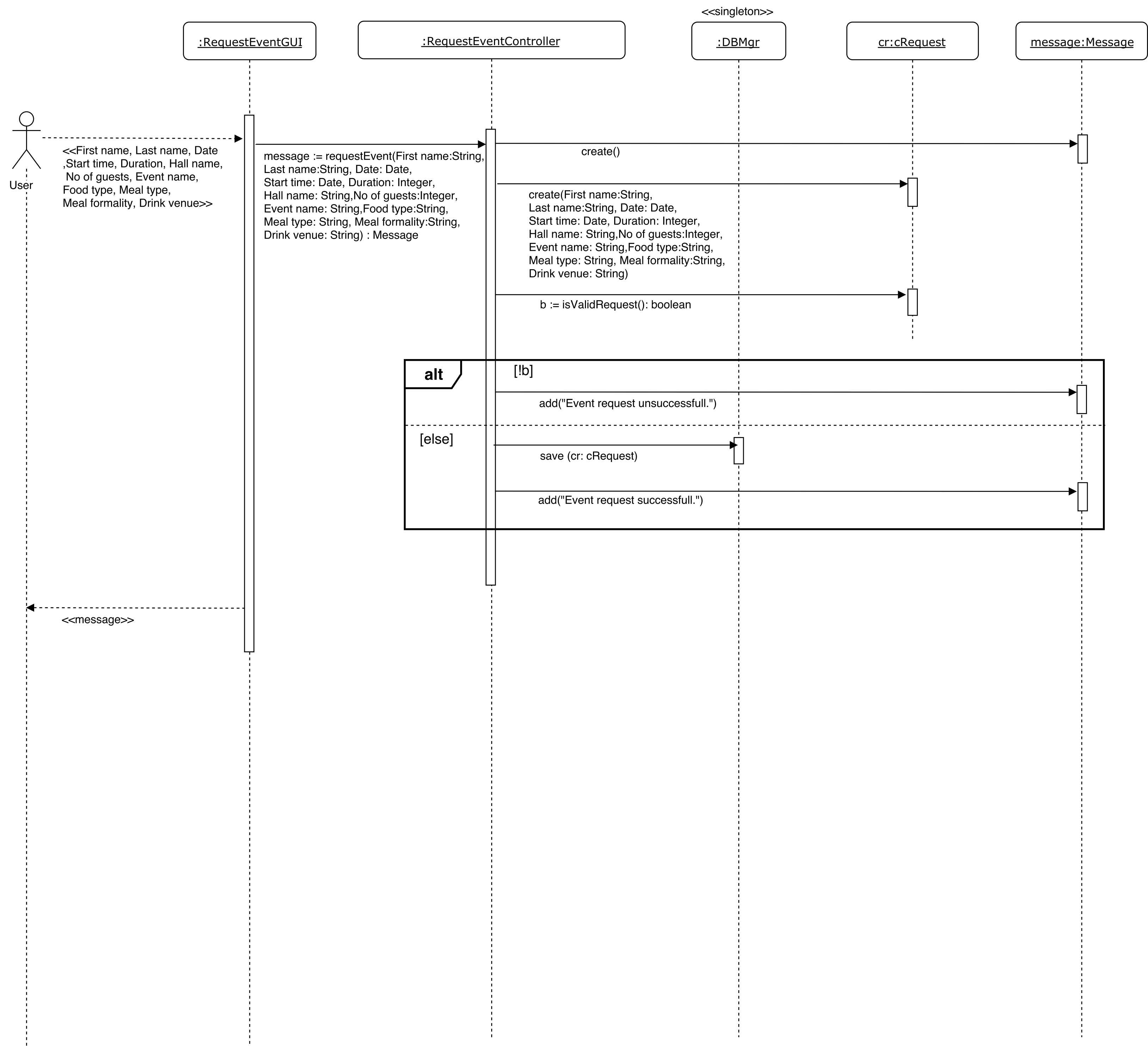
Sequence Diagram : Review registration requests



# Sequence Diagram : Perform Estimation



# Sequence Diagram : Request event



# Sequences Diagrams tradeoffs

## UC1 : Login

1- Even though the LoginGUI interacts directly with the system user, the fragment checking the validity of the system user's credentials is put on the LoginController to avoid unrelated logic in the LoginGUI and also to avoid changing the logic if we change the GUI. GUIs are known to change often.

2- The systemUser object being the one that verifies the password is a simple solution than having the LoginController ask for the password then verify it itself. The systemUser object already has all the information about the user, thus having the method verifying informations about the systemuser.

3- The same idea applies to the getType() function. The systemUser object already had this information.

4- The redirection to the new HomeGUI is made from the LoginController to avoid the same problems mentionned in previous point 1-

## UC6 : Review registration requests queue

1- The fragment at ReviewRRequestsController checking that the array of object received from database is not null to avoid overloading the ReviewRRequestsGUI with logic. The checking is split between ReviewRRequestsController and ReviewRRequestsGUI.

2- The sort\_rr method is the responsibility of the rRequest object because it has all the necessary informations about each request in order to sort them.

## UC16: Perform estimation

- 1- The estimateCost function can be put in either estimCost object or eventCost objects. The method needs both the informations at estimCost given by user and the informations about the cost of each resources that is stored in the eventCost object. The choice was made for estimCost.

## UC18: Request event

- 1- The fragment is put in the RequestEventController to make the RequestEventGUI dumb and responsible for delivering messages only.
- 2- The isValidRequest method is the responsibility of the cRequest object because it has the informations about the request and thus overloading the RequestEventController is preferred.