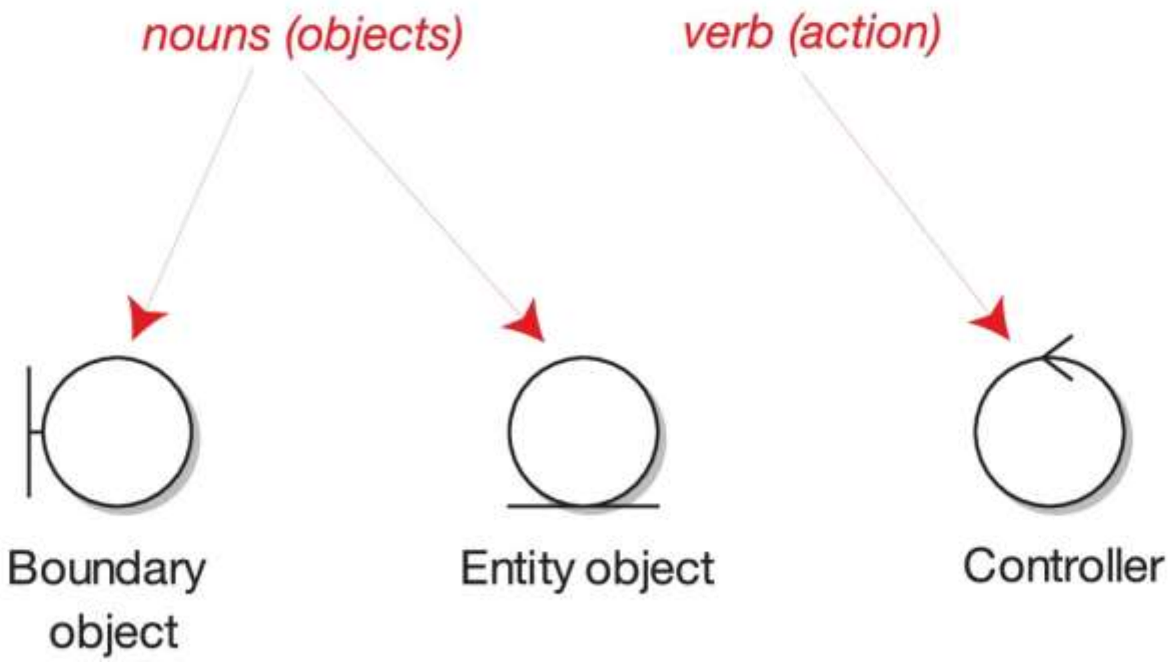
Robustness Diagrams

- To get to detailed design you need to link use cases to objects
- Robustness diagrams makes you analyze the use case text and identify a set of objects for each use case
- A robustness diagram is an object picture of a use case
- Drawing a robustness diagram ensures that the use case is written in the context of the domain model

Elementos del “Robustness Diagram”

There's a direct 1:1 correlation between the flow of action in the robustness diagram and the steps described in the use case text.



- **Boundary objects:** The “interface” between the system and the outside world Boundary objects are typically screens or web pages (i.e., the presentation layer that the actor interacts with).
- **Entity objects:** Classes from the domain model. Data that needs to be stored.
- **Controllers:** The “glue” between the boundary and entity objects.
- Nouns can talk to verbs (and vice versa).
- Nouns can’t talk to other nouns.
- Verbs can talk to other verbs.

General Notions

- Link use cases to objects
- Bridge the gap from analysis to design
- From use case text to a first draft of objets
- From use cases to object-oriented designs
- Use cases are the “what” and design is the “how”, robustness diagrams are the bridge
- Robustness diagrams are the preliminary design

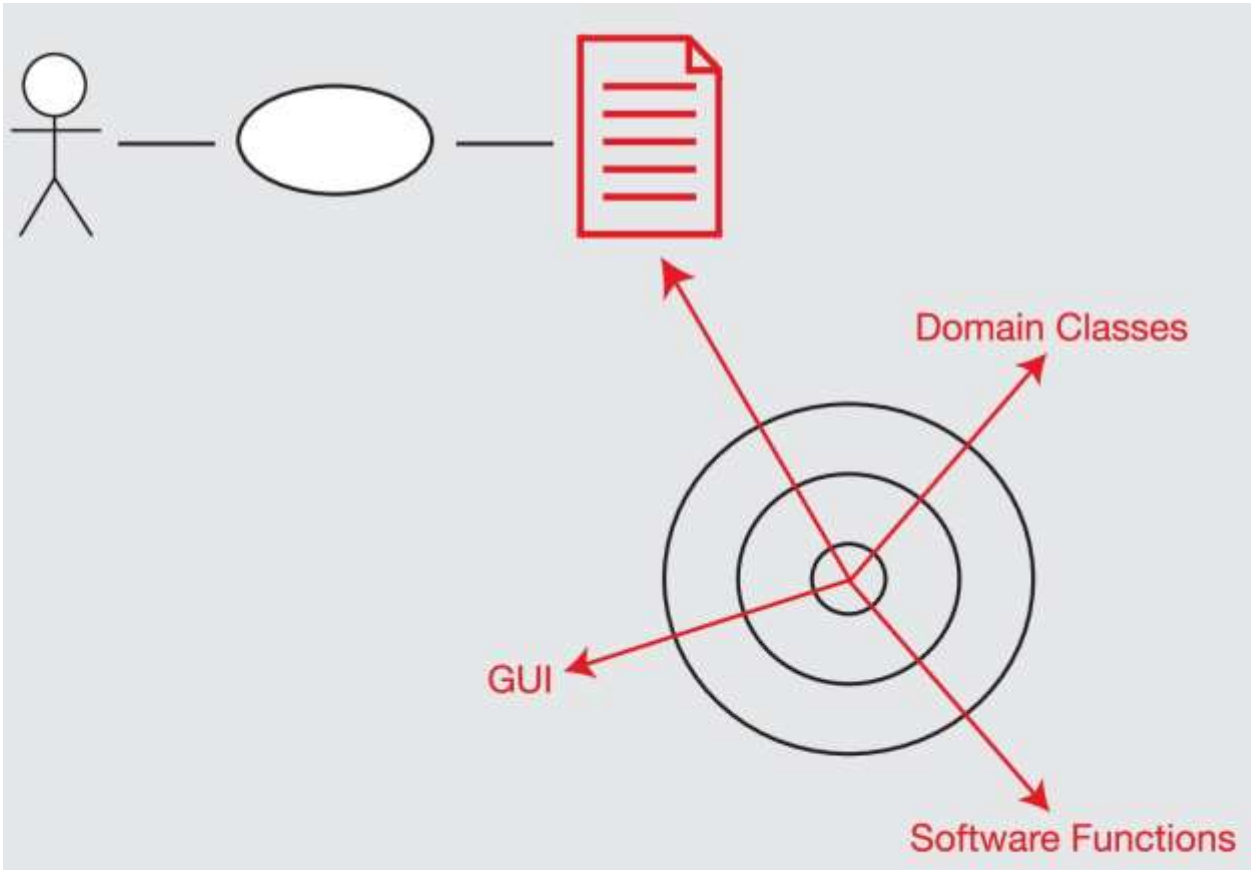
General Notions (Cont)

- Shows participating classes and software behaviour
- Behaviour is not yet linked to classes
- Flow of action is represented by a line between classes
- The entity classes come from the domain model
- The boundary classes from the use cases text

General Notions (Cont)

- Boundary and entity classes will generally beomce object instances in a sequence diagram
- The controllers will generally become the messages

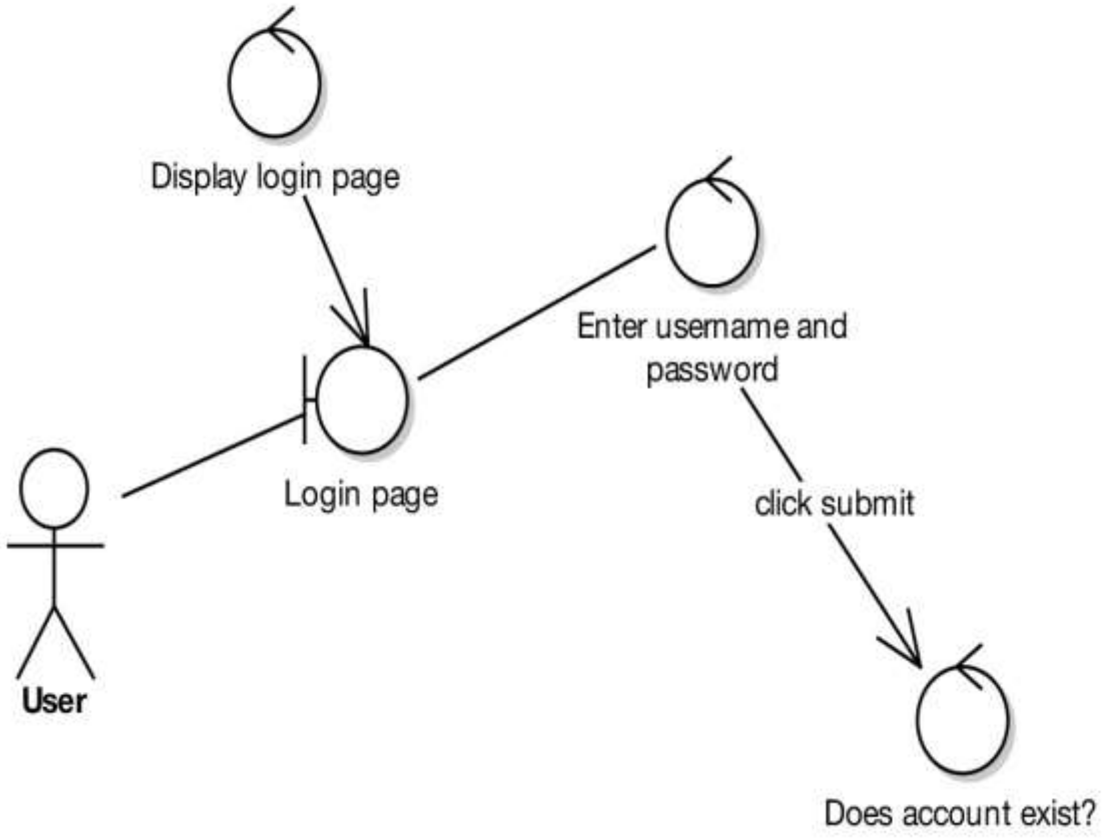
GUI, Domain Clasess, and Softwre Functions are Tied By Robustness Diagrams



- **Boundary objects:** The “interface” between the system and the outside world Boundary objects are typically screens or web pages (i.e., the presentation layer that the actor interacts with).
- **Entity objects:** Classes from the domain model
- **Controllers:** The “glue” between the boundary and entity objects.
- Nouns can talk to verbs (and vice versa).
- Nouns can’t talk to other nouns.
- Verbs can talk to other verbs.

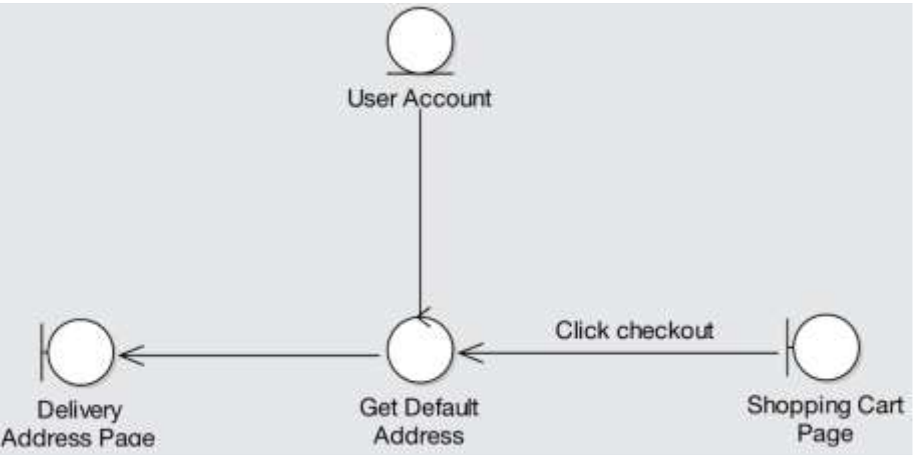
Ejemplo de "Robustness Diagram"

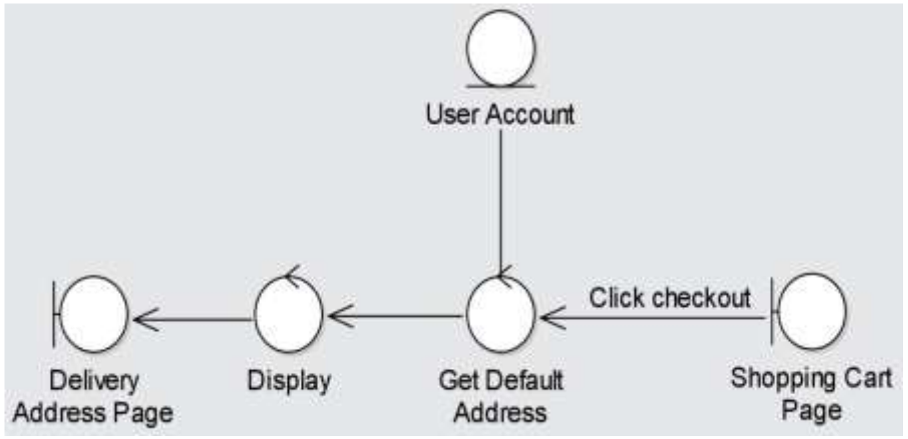
BASIC COURSE:
The user clicks the login link from any of a number of pages; the system displays the login page. The user enters their username and password and clicks Submit. The system checks the master account list to see if the user account exists. If it exists, the system then checks the password. The system retrieves the account information, starts an authenticated session, and redisplays the previous page with a welcome message.



[Full diagram](#)

Why Display Controllers





- Los “Display Controllers” nos recuerdan de hay un trabajo de inicialización en presentar un componente grafico.

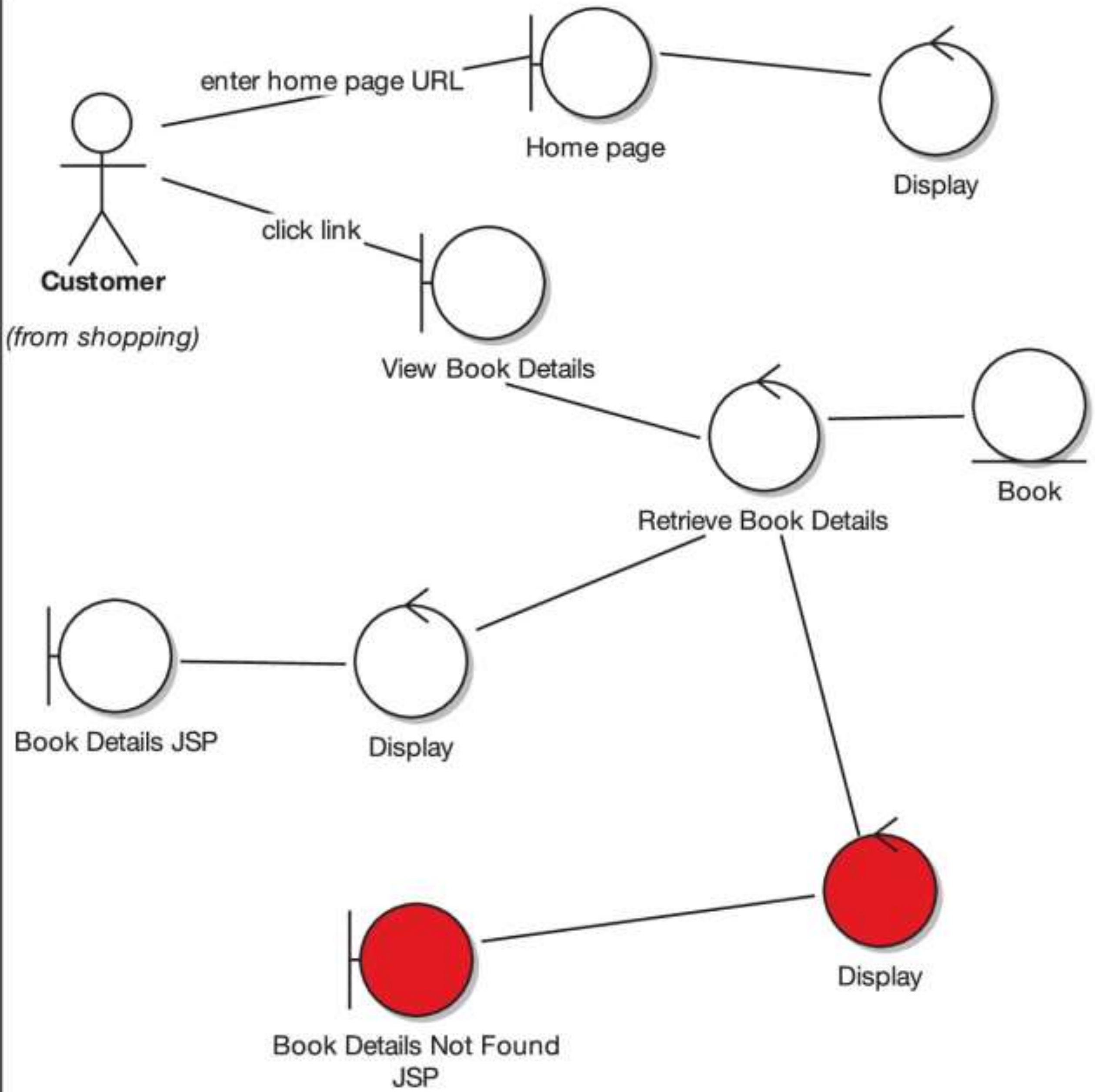
Show Book Details, First Draft

BASIC COURSE:

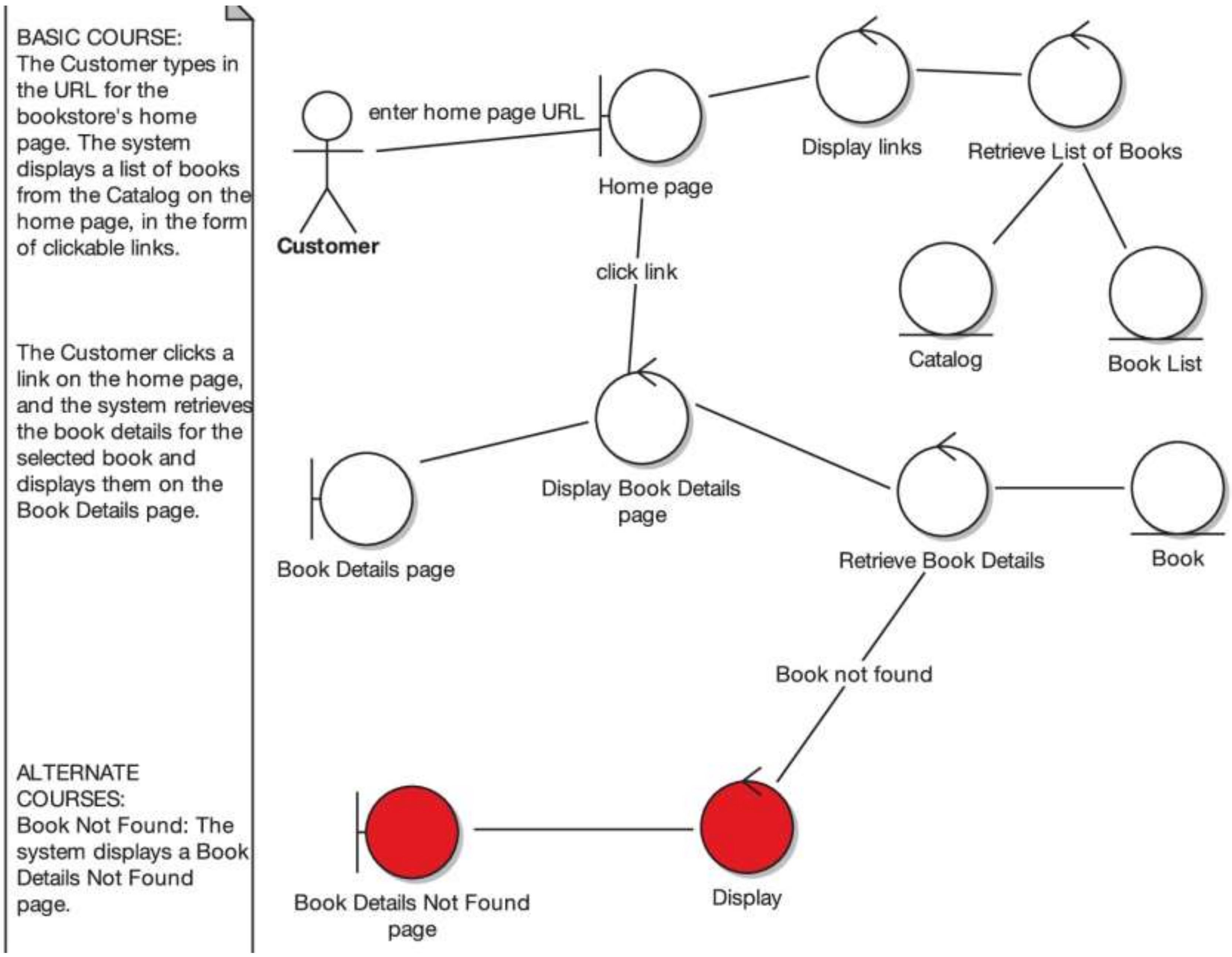
The Customer types in the URL for the bookstore's home page, which the system displays. Then the Customer clicks a link to view a Book. The system retrieves the Book details, and displays the Book Details screen.

ALTERNATE COURSES:

Book Not Found: The system displays a Book Details Not Found screen.



Show Book Details, Corrected



BASIC COURSE:
The Customer types in the URL for the bookstore's home page. The system displays a list of books from the Catalog on the home page, in the form of clickable links.

The Customer clicks a link on the home page, and the system retrieves the book details for the selected book and displays them on the Book Details page.

ALTERNATE COURSES:
Book Not Found: The system displays a Book Details Not Found page.

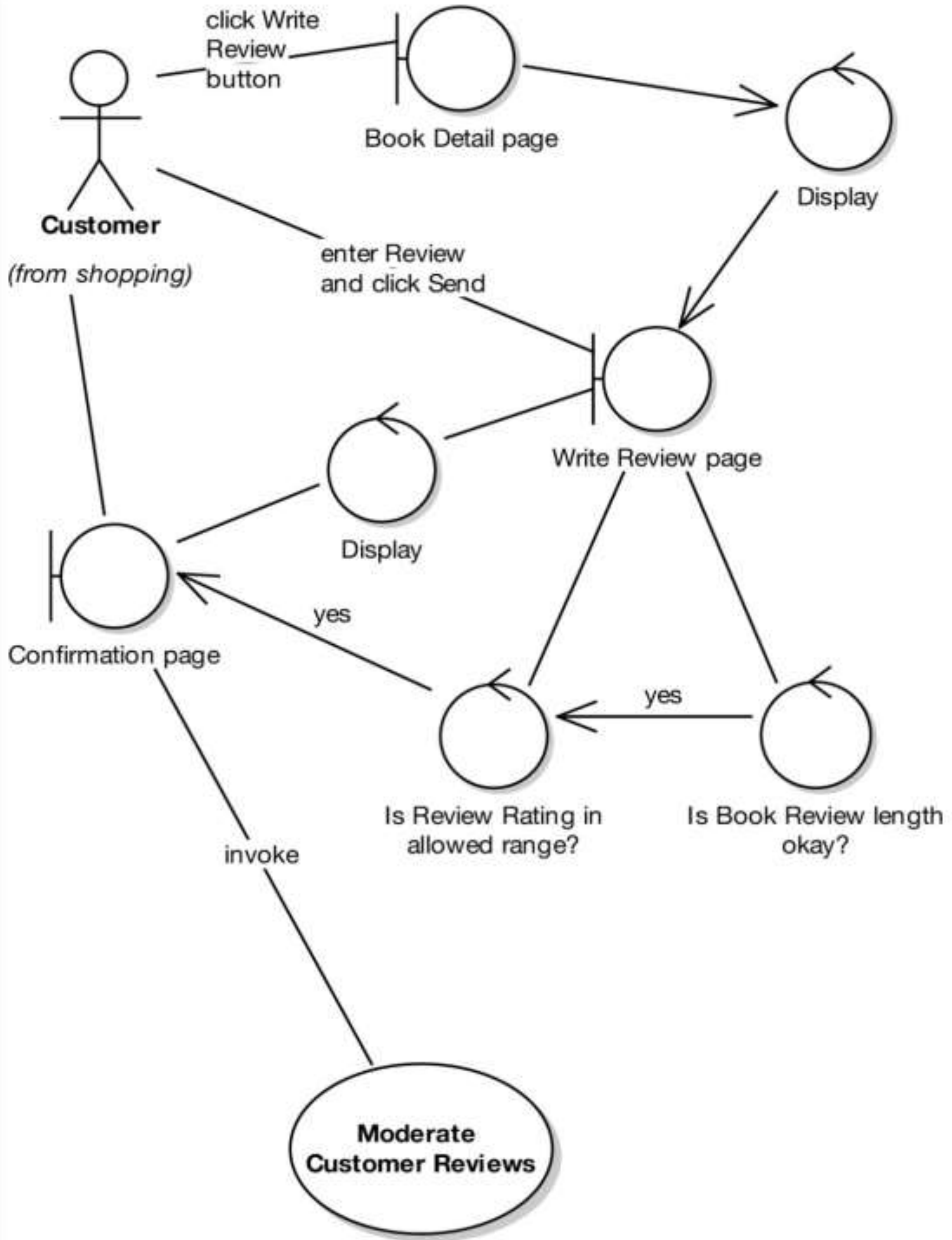
Write Customer Review-1 Iter

BASIC COURSE:
On the Book Detail page for the book currently being viewed, the Customer clicks the Write Review button, and the system shows the Write Review page. The Customer types in a Book Review, gives it a Book Rating out of 5 stars, and clicks the Send button. The system ensures that the Book Review isn't too long or short, and that the Book Rating is within 1-5 stars. The system then displays a confirmation page, and the review is sent to a Moderator ready to be added.

ALTERNATE COURSES:
User not logged in: The user is first taken to the Login page, and then to the Write Review page once they've logged in.

The user enters a review which is too long (text > 1MB): The system rejects the review, and responds with a message explaining why the review was rejected.

The review is too short (< 10 characters): The system rejects the review.


click Write Review button
Book Detail page
Display
Customer
(from shopping)
enter Review and click Send
Write Review page
Display
Confirmation page
yes
yes
Is Review Rating in allowed range?
Is Book Review length okay?
invoke
Moderate Customer Reviews

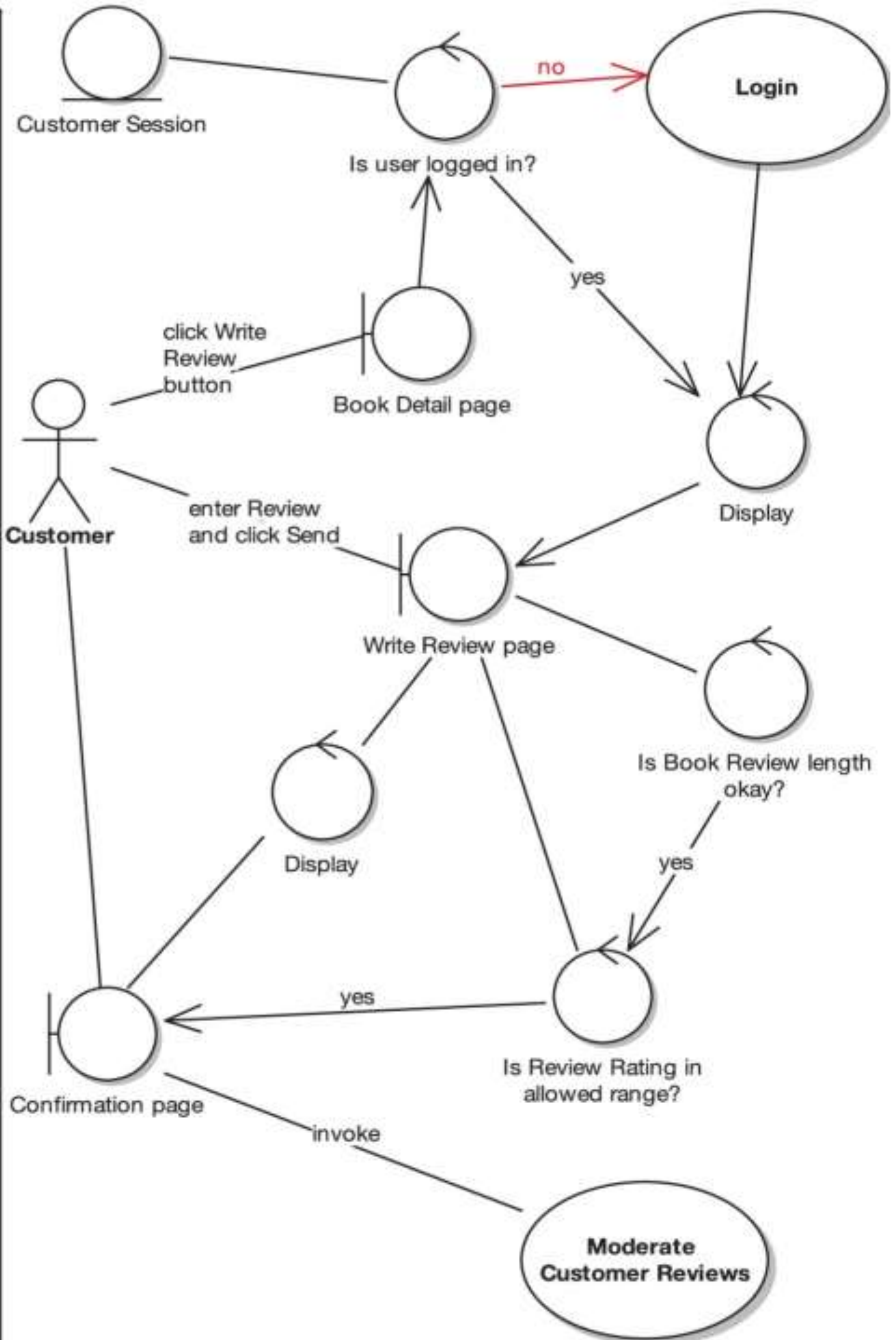
Write Customer Review-2 Iter

BASIC COURSE:
On the Book Detail page for the book currently being viewed, the Customer clicks the Write Review button. The system checks the Customer Session to make sure the Customer is logged in, and then displays the Write Review page. The Customer types in a Book Review, gives it a Book Rating out of 5 stars, and clicks the Send button. The system ensures that the Book Review isn't too long or short, and that the Book Rating is within 1-5 stars. The system then displays a confirmation screen, and the review is sent to a Moderator ready to be added.

ALTERNATE COURSES:
User not logged in: The user is first taken to the Login page, and then to the Write Review page once they've logged in.

The user enters a review which is too long (text > 1MB): The system rejects the review, and responds with a message explaining why the review was rejected.

The review is too short (< 10 characters): The system rejects the review.



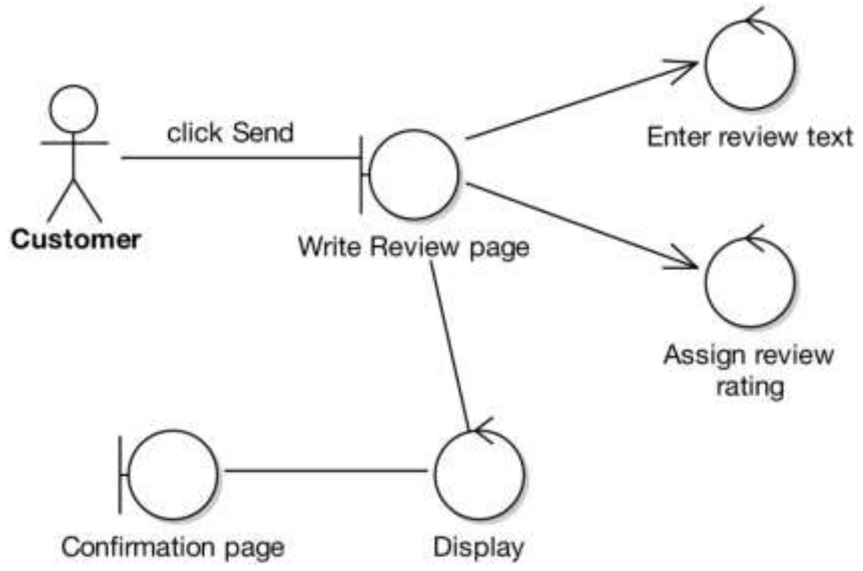
Customer Session
Is user logged in?
no
Login
yes
click Write Review button
Book Detail page
Customer
Display
enter Review and click Send
Write Review page
Is Book Review length okay?
Display
yes
yes
Is Review Rating in allowed range?
Confirmation page
invoke
Moderate Customer Reviews

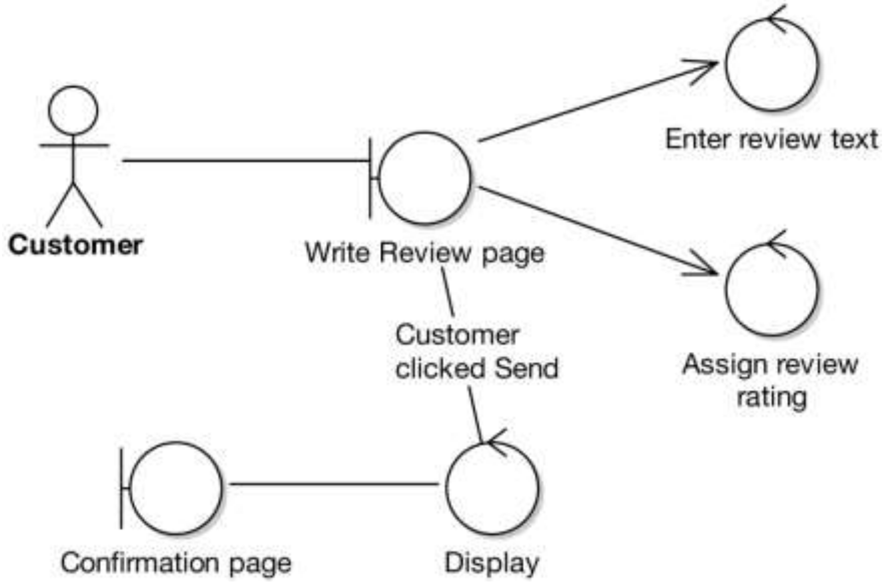
3-iter

Login Status in “Write Customer Review”

- What class can we ask about what the login status of the customer is?
- Looks like we have to add a new domain class
  - Diag. with new Customer Session Class
- Note that when the customer enters the review and clicks send it is unclear which controller gets activates

Write Customer Review-Probs (Cont)

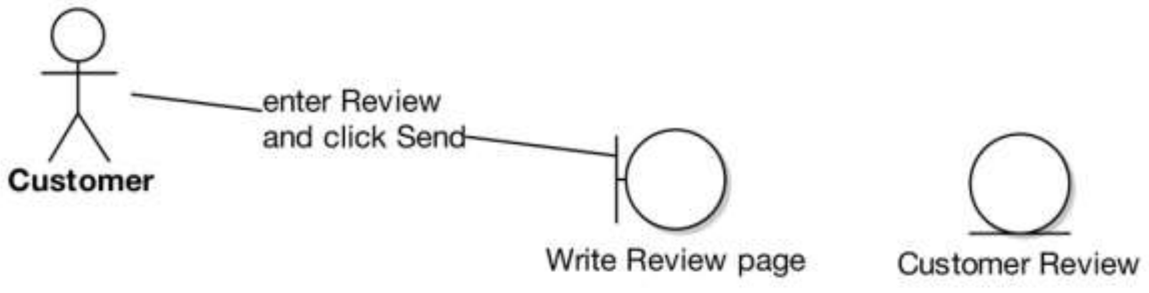




- Define where each action goes after the boundary object.

Write Customer Review Problems

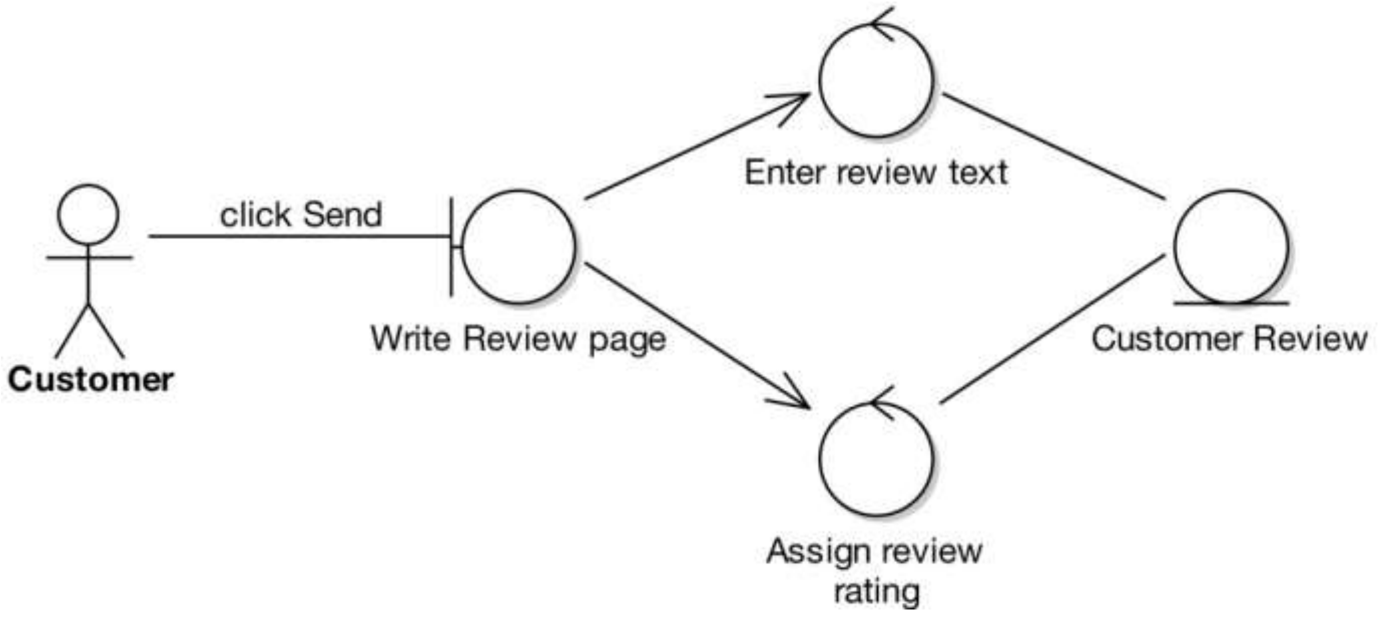
- The Customer Review entity class should be the container for the review information being entered by the Customer



- “enter Review” should be a controller.
- “click Send” should also be a controller.

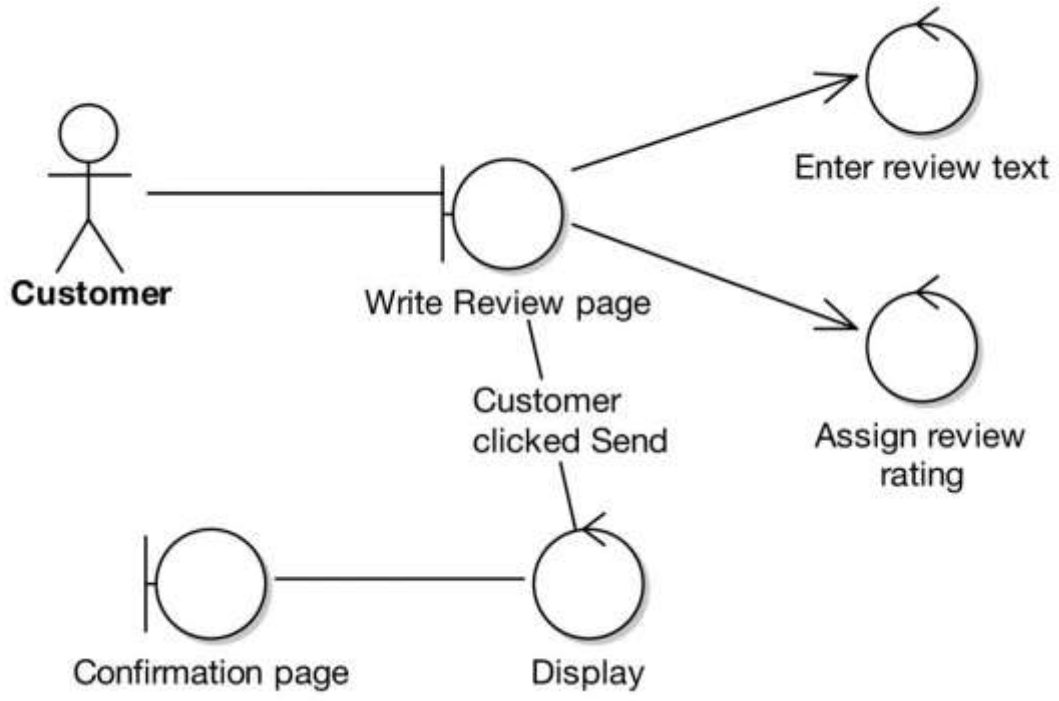
Write Customer Review Problems (Cont)

- Here we add the two controllers necessary represent the Customer’s actions in the Write Review Page making the Customer Review class the container of the customer review information



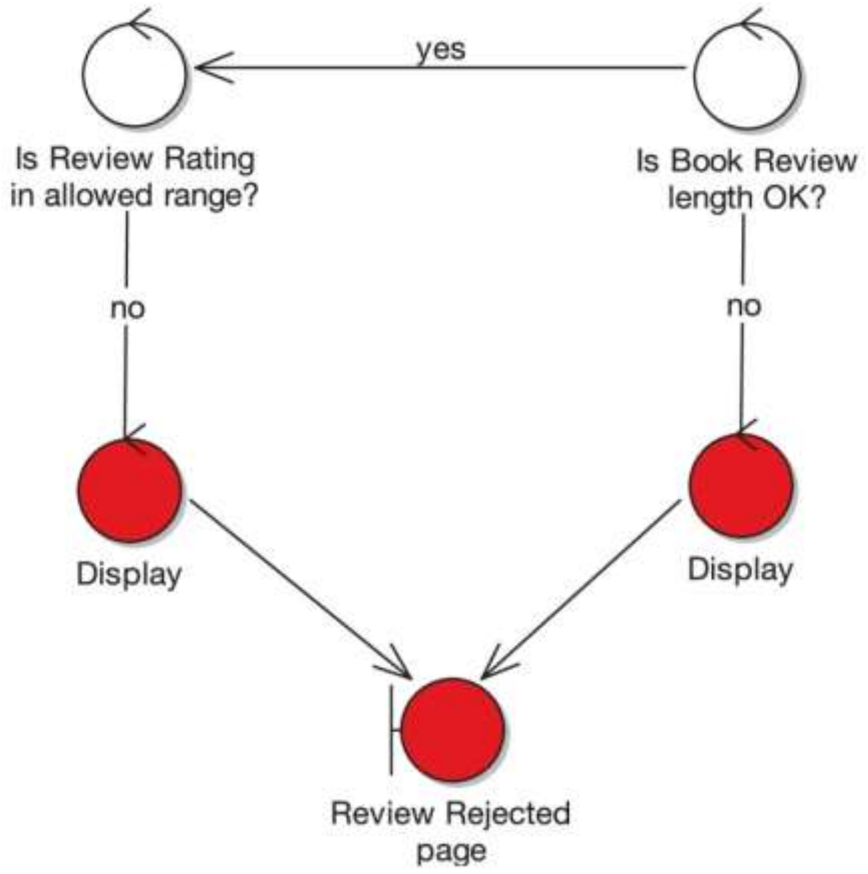
Write Customer Review Problems (Cont)

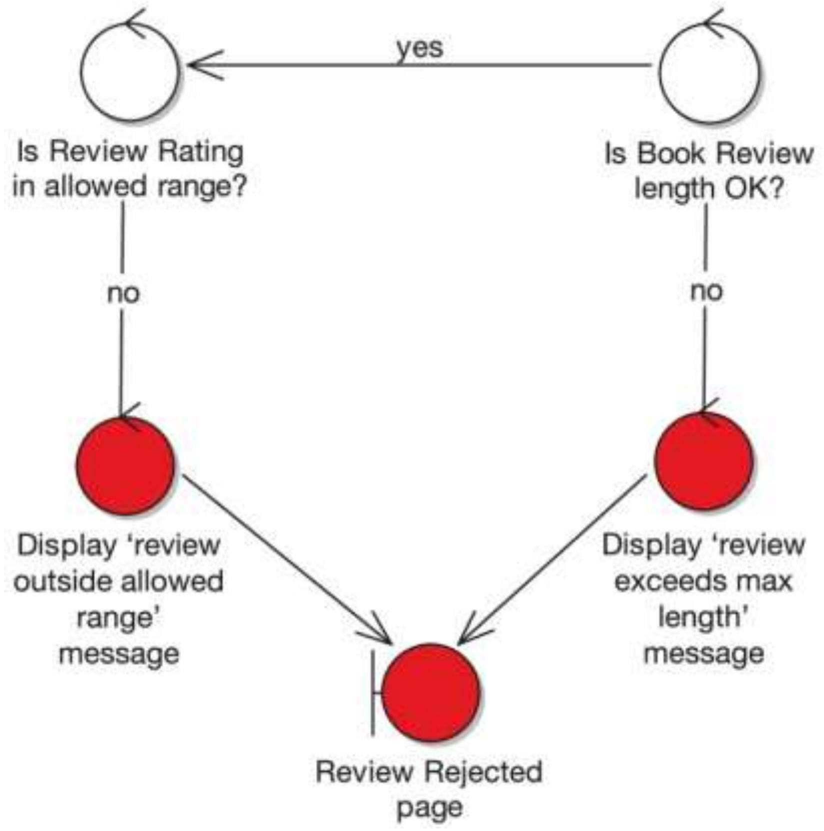
- The other problem is that we cannot link the Customer “send” action with the response of the system



2-16-2023

Write Customer Review-Probs (Cont)




yes
Is Review Rating in allowed range?
Is Book Review length OK?
no
no
Display 'review outside allowed range' message
Display 'review exceeds max length' message
Review Rejected page

- **Diag. Right:** Insufficient detail as to what gets displayed to the user.
- **Diag. Left:** Corrected
- Final Complete Diagram

General Concerns

- Always show the users action coming off the boundary objects, linking the previous screen with the next screen via a controller.
- Only one book details page is really necessary.
- ASP or JSP, etc. is too technology specific.
- Use a noun to name a page.
- Be able to trace the data flow from the boundary classes (usually screens) to the entity classes and vice-versa

General Concerns (Cont.)

- Make sure that the robustness diagrams are in the context of the domain model and the GUI
- All entity classes should appear in the updated domain model

Puntos Importantes de Casos de Usos

- Write the use case in the context of the object model.
- Follow the two-paragraph rule. (No mas de dos párrafos.)
- Reference boundary classes (e.g., screens) by name
- Reference domain classes by name.
- Write your use case using an event/response flow, describing both sides of the user/system dialogue.
- Write your use cases in active voice.
- Organize your use cases with actors and use case diagrams

Robustness Diagrams Checks

- Make sure the entity classes have been populated with attributes.
- All the screens should have names.
- Make sure that the data flow can be traced from the entity classes to the screens and vice versa
- Make sure the use case text matches the robustness diagrams.

Robustness Diagrams Checks (Cont.)

- The entities in the robustness diagrams should appear in the domain classes diagram.
- Don’t yet assign behavior to classes, just include all the logical functions or controllers needed in the robustness diagrams.
- The controllers will become methods of classes when the sequence diagrams get created.

Robustness Diagrams Checks (Cont.)

- Some groups of controllers may become management classes.
- The robustness diagram should focus on the logical flow of the use case.