Appendix B

iCoot Case Study

B.1 BUSINESS REQUIREMENTS

This section documents the business requirements modeling carried out during the requirements phase of the iCoot development, in terms of project mission statement and business use case model. The business use case model also applies to the full Coot system.

B.1.1 Customer's Mission Statement

Below is the mission statement delivered by Nowhere Cars at the start of the Coot project:

> Since we automated the tracking of cars at our stores – using bar codes, counter-top terminals and laser readers – we have seen many benefits: the productivity of our rental assistants has increased 20%, cars rarely go missing and our customer base has grown strongly (according to our market research, this is at least partly due to the improved perception of professionalism and efficiency.
>
> The management feels that the Internet offers further exciting opportunities for increasing efficiency and reducing costs. For example, rather than printing catalogs of available cars, we could make the catalog available to every Internet surfer for browsing on-line. For privileged customers, we could provide extra services, such as reservations, at the click of a button. Our target saving in this area is a reduction of 15% in the cost of running each store.
>
> Within two years, using the full power of e-commerce, we aim to offer all of our services via a Web browser, with delivery and pick-up at the customer's home, thus achieving our ultimate goal of the virtual rental company, with minimal running costs relative to walk-in stores.

Working with the customer, this mission statement was expanded into business use cases.

B.1.2 Actor List

- Assistant: An employee at one of our stores who helps a Customer to rent a Car and reserve a CarModel.
- Customer: A person who pays us money in return for one of our standard services.

- Member: A Customer whose identity and credit-worthiness have been validated and who, therefore, has access to special services (such as making a Reservation by phone or over the Internet).
- NonMember: A Customer whose identity and credit-worthiness have not been checked and who, therefore, must provide a deposit to make a Reservation and surrender a copy of their License to rent a Car.
- Auk: The existing system that handles Customer details, Reservations, Rentals and the Catalog of available CarModels.
- DebtDepartment: The department that deals with unpaid fees.
- LegalDepartment: The department that deals with accidents in which a rented Car has been involved.

B.1.3 Use Case List

- B1:Customer Rents Car: Customer rents a Car that they have selected from those available.
- B2:Member Reserves CarModel: Member asks to be notified when a CarModel becomes available.
- B3:NonMember Reserves CarModel: NonMember pays a deposit to be notified when a CarModel becomes available
- B4:Customer Cancels Reservation: Customer cancels an unconcluded Reservation, by phone or in person.
- B5:Customer Returns Car: Customer returns a Car that they have rented.
- B6:Customer Told CarModel Is Available: Customer is contacted by an Assistant when a Car becomes available.
- B7:Car Reported Missing: Customer or Assistant discovers that a Car is missing.
- B8:Customer Renews Reservation: Customer renews a Reservation that has been outstanding for more than a week.
- B9:Customer Accesses Catalog: Customer browses the catalog, in-Store or at home.
- B10: Customer Fined for Uncollected Reservation: Customer fails to collect a Car that they have reserved.
- B11:Customer Collects Reserved Car: Customer collects a Car that they have reserved.
- B12:Customer Becomes a Member: Customer provides CreditCard details and proof of address to become a Member.
- B13:Customer Notified Car Is Overdue: Assistant contacts Customer to warn them that a Car they have rented is more than a week overdue.
- B14:Customer Loses Keys: Replacement keys are provided for a Customer who has lost them.
- B15:MembershipCard Is Renewed: Assistant contacts Member to renew membership when their CreditCard has expired.
- B16:Car Is Unreturnable: A Car is wrecked or breaks down.

B.1.4 Use Case Communication Diagrams

Communication diagrams were not used widely during business requirements modeling (although they were used extensively during system requirements gathering). However, one diagram (see Figure B.1) was produced to illustrate the external and internal actors involved in B3:NonMember Reserves CarModel.

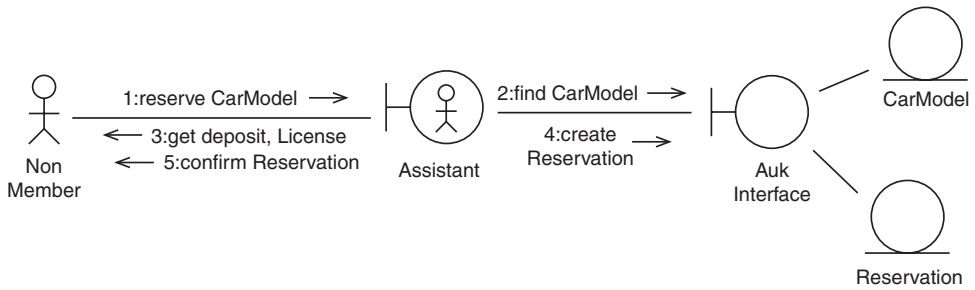

Figure B.1: Communication diagram for B3:NonMember Reserves CarModel

B.1.5 Use Case Activity Diagrams

Activity diagrams were not used widely during the business requirements modeling. However, one diagram (see Figure B.2) was produced to illustrate the finer points of the B3:NonMember Reserves CarModel use case.

B.1.6 Use Case Details

B1:Customer Rents Car.

1. Customer tells Assistant which CarModel they'd like to rent.
2. If Auk indicates no such Car is available, Customer is offered an alternative.
3. If there is a Car available, Assistant marks the Car as taken in Auk.
4. Assistant asks for Customer's License to confirm their identity.
5. For a Member, Assistant takes their number from their MembershipCard and checks that they have no outstanding fees and that they have not been barred.
6. For a NonMember, Assistant checks whether they're already in Auk; if they're not, Assistant scans a copy of their License into Auk, and records their name, phone number and license number.
7. If Customer's details are satisfactory and they have paid any outstanding fees, they're charged for the Rental.
8. If the payment fails, the Car is released in Auk.

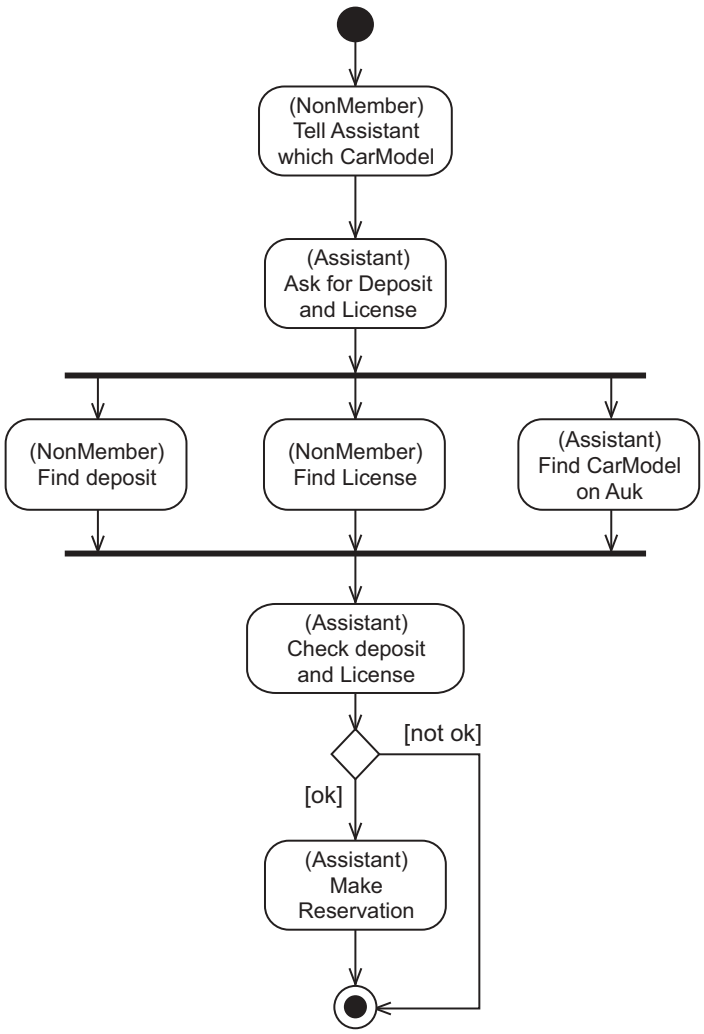

Figure B.2: Activity diagram for B3:NonMember Reserves CarModel

9. If the payment does not fail, the Customer is given the keys and directed to the display area.

B2:Member Reserves CarModel.

1. Member tells Assistant their membership number (over the phone or in person).
2. Member tells Assistant which CarModel to reserve.
3. If Member has not been barred, their CreditCard has not expired, and they have no outstanding fees, a Reservation is made on Auk.

4. If the Reservation is being made over the phone, Member can pay outstanding fees by confirming their CreditCard details, which must match those stored in Auk and must not have expired.
5. Member is told the reservation number.

B3:NonMember Reserves CarModel.

1. NonMember tells Assistant which CarModel to reserve.
2. Assistant finds CarModel on Auk.
3. Assistant asks for a deposit for the Reservation.
4. Assistant asks for NonMember's License and phone number.
5. Assistant checks License visually.
6. If License looks valid, Assistant creates a new Reservation, recording the License number, phone number and a scan of the License in Auk.
7. Assistant gives NonMember a Reservationslip containing the unique reservation number.

B4:Customer Cancels Reservation.

1. At any time, Customer can cancel a Reservation.
2. Member objects can do this over the phone or in person, by providing their membership number.
3. NonMembers must cancel in person: they present their License to an Assistant, who checks that it matches the scan in Auk, and refunds their deposit.
4. If a Car has already been moved to the reserved area, a matching Car is moved back to the display area.

B5:Customer Returns Car.

1. When a Car is returned to the check-in area, Assistant scans bar code to confirm the return and checks that the tank is full.
2. Car is returned to the display area by an Assistant.
3. If Customer returns an overdue Car or a Car with a tank that is not full, Customer must pay the appropriate amount – Members can do this using their existing credit card details, if they have not expired.
4. If the Customer refuses to pay, their details are passed to the DebtDepartment.

B6:Customer Told Car Model is Available.

1. When a Car is returned, Auk tells Assistant whether it matches any Reservation objects.
2. If so, Assistant moves Car to the reserved area.
3. On a first-come-first-served basis, Assistant will try to contact a matching Customer by phone.
4. If a Customer can't be reached within two days, their Reservation is canceled and the Car is moved out of the reserved area to the display area.

B7:Car Reported Missing.

1. If a Car that Auk indicates is in the display area can't be found when it is needed or during a stock check, Car is reported stolen to the police.
2. If a Car is reported missing by a Customer, it is reported stolen to the police, along with License details of the Customer (as the last known keeper of the vehicle).
3. In both cases, the date of loss is recorded on Auk.

B8:Customer Renews Reservation.

1. If a Reservation can't be satisfied within seven days, the Reservation must be renewed.
2. Assistant has two days to contact Customer by phone to see if they wish to renew the Reservation for a further seven days.
3. If the Customer doesn't wish to renew, the Reservation is canceled; Customer must return to the Store and present their License to retrieve their deposit.

B9:Customer Accesses Catalog.

1. Customers can come into the Store to browse a paper catalog.
2. For a fee, they can take a copy of the catalog home with them.
3. If they choose to join the mailing list, they will receive a free copy of the catalog by mail every six months.

B10: Customer Fined for Uncollected Reservation.

1. If a CarModel has become available for a particular Reservation and an Assistant told the Customer by phone that it's available, Customer has two days to collect.
2. If Customer fails to collect, the Reservation is concluded and an Assistant moves a matching Car from the reserved area back to the display area.
3. For NonMembers, their deposit is forfeited.
4. For Members, a fine is recorded on Auk and their details are passed to the DebtDepartment.

B11:Customer Collects Reserved Car.

1. Customer comes to the Store to collect a Car from the reserved area.
2. Customer presents License.
3. If the License matches the details on Auk, the Reservation is marked as concluded.
4. An Assistant gives the keys to the Customer and directs them to the reserved area.

B12:Customer Becomes Member.

1. In order to become a Member, Customer must offer their License, further proof of address, and a credit card.
2. Assistant checks License and proof of address.
3. Assistant checks CreditCard with CreditCardCompany.

4. If okay, Assistant records License number, address, phone number and CreditCard details in Auk.
5. Auk issues new MembershipCard with unique membership number.
6. If the CreditCard expires, no further member actions are permitted unless the member returns to the Store to show a new CreditCard.

B13:Customer Notified Car is Overdue.

1. Since a Rental is paid up-front, Customer is warned if they have forgotten to return a Car.
2. If Car is more than one week overdue, an Assistant will attempt to contact Customer by phone.
3. If Customer can't be contacted for two weeks, Car is reported missing (see B7).

B14:Customer Loses Keys.

1. If Customer notifies Assistant that they have lost keys, replacement keys are provided, by courier if necessary.
2. Cost of replacement is added to Customer's details in Auk.

B15:MembershipCard is Renewed.

1. Auk records that Member whose CreditCard has expired is not in good standing.
2. Auk informs Assistant that Member's Credit card has expired.
3. Assistant contacts Member by phone to tell them that they must renew their membership.
4. Member returns to Store with fresh CreditCard and details are entered into Auk.
5. Auk records that Member is in good standing.

B16:Car is Unreturnable.

1. If Customer tells Assistant that Car is wrecked or breaks down, Assistant arranges recovery.
2. If Car is wrecked, details are passed to LegalDepartment.

B.2 SYSTEM REQUIREMENTS

This section documents the results of system modeling during the requirements phase of the iCoot development, in terms of user interface sketches and a system use case model.

B.2.1 User Interface Sketches

The user interface sketches for iCoot, produced with the help of the customer, are shown in Figures B.3 through B.10.

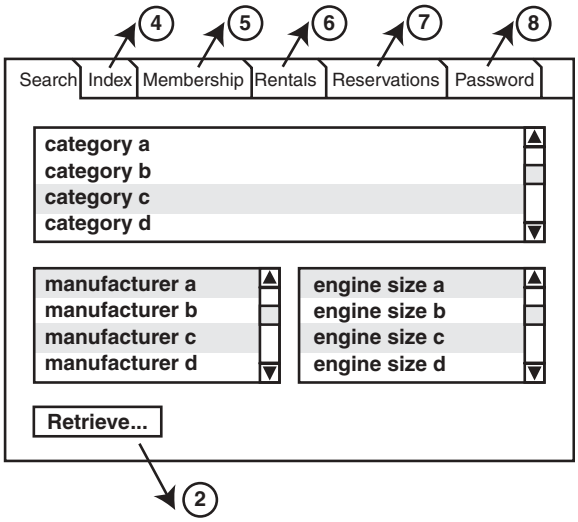

Figure B.3: User interface sketch 1 (creating a query)

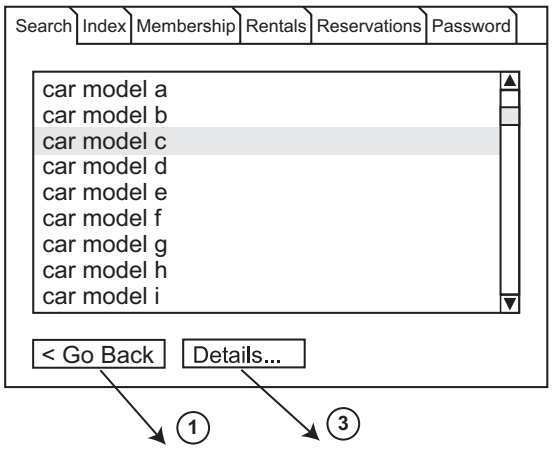

Figure B.4: User interface sketch 2 (viewing results)

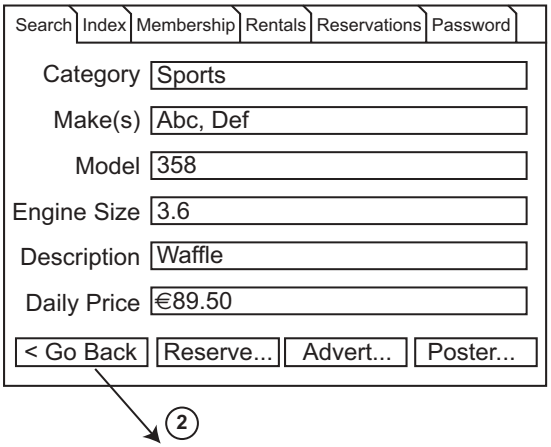

Figure B.5: User interface sketch 3 (viewing car model details)

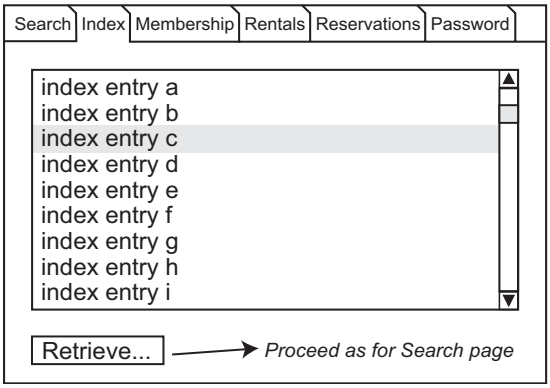

Figure B.6: User interface sketch 4 (selecting an index heading)

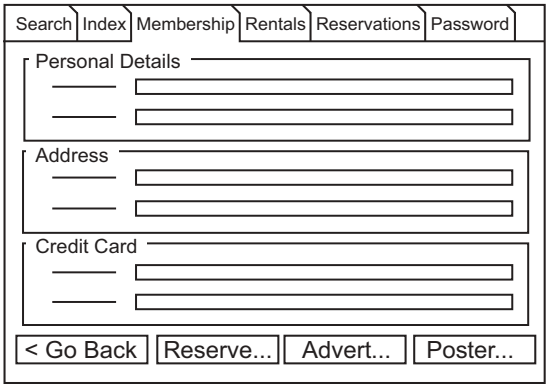

Figure B.7: User interface sketch 5 (viewing membership details)

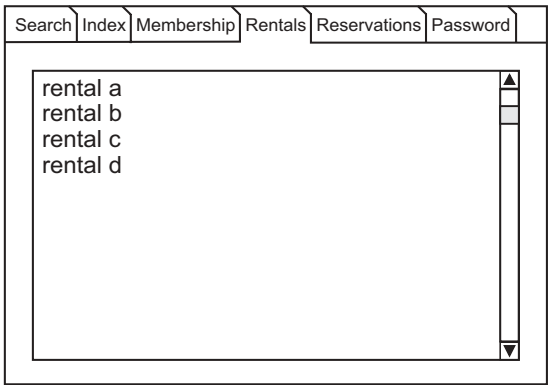

Figure B.8: User interface sketch 6 (viewing rentals)

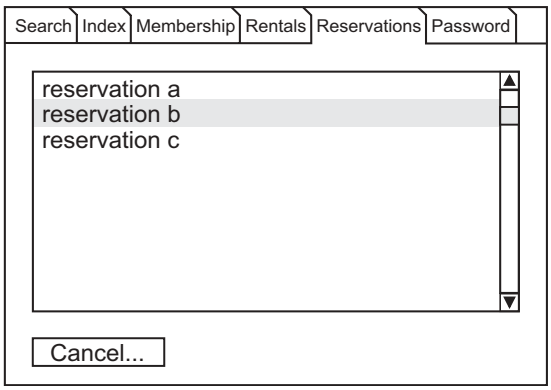

Figure B.9: User interface sketch 7 (viewing reservations)

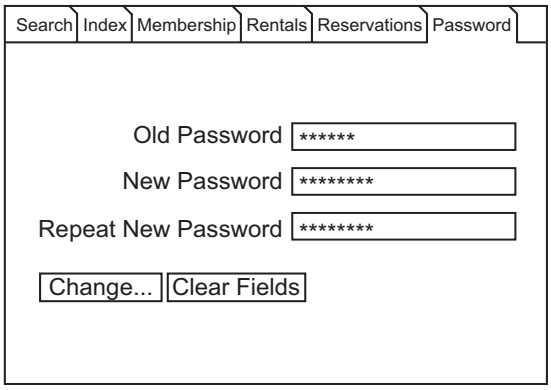

Figure B.10: User interface sketch 8 (changing a password)

B.2.2 Actor List

- Customer: A person using a Web browser to access iCoot.
- Member: A Customer who has presented their name, address and CreditCard details at one of our Stores; each Member is given an Internet password to accompany their membership number. (Specializes Customer.)
- NonMember: A Customer who is not a Member. (Specializes Customer.)
- Assistant: An employee at a Store who contacts Members to tell them about the progress of their Reservations.

B.2.3 Use Case List

- U1:Browse Index: A Customer browses the index of CarModels. (Specializes U13, includes U2.)
- U2:View Results: A Customer is shown the subset of CarModels that were retrieved. (Included by U1 and U4, extended by U3.)
- U3:View CarModel Details: A Customer is shown the details of a retrieved CarModel, such as description and advert. (Extends U2, extended by U7.)
- U4:Search: A Customer searches for CarModels by specifying Categories, Makes and engine sizes. (Specializes U13, includes U2.)
- U5:Log On: A Member logs on to iCoot using their membership number and current password. (Extended by U6, U8, U9, U10 and U12.)
- U6:View Member Details: A Member views some of the details stored by iCoot, such as name, address and CreditCard details. (Extends U5.)
- U7:Make Reservation: A Member reserves a CarModel when viewing its details. (Extends U3.)
- U8:View Rentals: A Member views a summary of the Cars they're currently renting. (Extends U5.)
- U9:Change Password: A Member changes the password that they use to log on. (Extends U5.)
- U10: View Reservations: A Member views summaries of their unconcluded Reservations, such as date, time and CarModel. (Extends U5, extended by U11.)
- U11:Cancel Reservation: A Member cancels an unconcluded Reservation. (Extends U10.)
- U12:Log Off: A Member logs off from iCoot. (Extends U5.)
- U13:Look for CarModels: A Customer retrieves a subset of CarModels from the catalog. (Abstract, generalized by U1 and U4.)

B.2.4 Use Case Diagram

The use case diagram for iCoot is shown in Figure B.11.

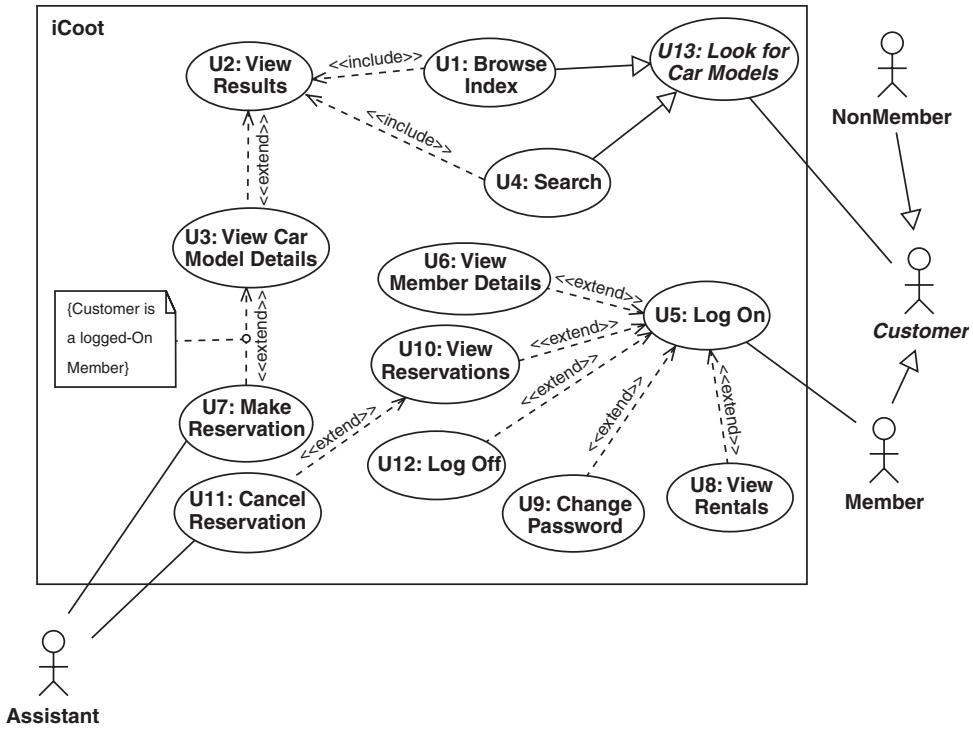

Figure B.11: Use case diagram for iCoot

B.2.5 Use Case Survey

The use case survey for iCoot, describing how the use cases fit together, is:

Any Customer can look for CarModels in the catalog, by browsing the CarModel index (U1) or by searching (U4). In the latter case, the Customer specifies the Categories, Makes and engine sizes that they're interested in. Either way, after each retrieval, the Customer is shown the results as a collection of matching CarModels (U2), along with basic information such as CarModel name. The Customer can then choose to view extra information about particular CarModel objects such as a description and an advert (U3).

A Customer who has become a Member can log on (U5) and gain access to extra services. The extra services are: making a Reservation (U7), canceling a Reservation (U11), checking membership details (U6), viewing outstanding Reservations (U10), changing their log-on password (U9), viewing their outstanding Rentals (U8) and logging off (U12).

Assistants are involved in the life cycle of Reservations, moving Cars to and from the reserved area, for example.

Customers come in two varieties, Members and NonMembers.

Browsing the index and searching for CarModels are two different ways of looking for CarModels (U13). In order to view CarModel details, a Customer must be viewing the results of looking for models (via the browsing or searching route).

In order to reserve a CarModel, a Member must be viewing its details (NonMembers can't make reservations, even when they're viewing details).

In order to cancel a Reservation, a Member must be viewing their outstanding Reservations.

B.2.6 Use Case Details

U1:Browse Index. (Specializes U13, includes U2.)
Preconditions: None.
1. Customer selects an index heading.
2. Customer elects to view CarModels for the selected index heading.
3. Include U2.

Postconditions: None.

U2:View Results. (Included by U1 and U4, extended by U3.)
Preconditions: None.
1. iCoot presents Customer with a summary of each retrieved CarModel, including model number and price.
2. Extend with U3.

Postconditions: None.

U3:View CarModel Details. (Extends U2, extended by U7.)
Preconditions: None.
1. Customer selects one of the matching CarModels.
2. Customer requests details of the selected CarModel.
3. iCoot displays details for the selected car model (makes, engine size, price, description, advert and poster).
4. If Customer is a logged-on Member, extend with U7.

Postconditions: iCoot has displayed details of selected CarModels.
Non-Functional Requirements: r1. Adverts should be displayed using a streaming protocol rather than requiring a download.

U4:Search. (Specializes U13, includes U2.)
Preconditions: None.
1. Customer selects required categories (if any).
2. Customer selects required Makes (if any).

3. Customer selects required engine sizes (if any).
4. Customer initiates the search.
5. Include U2.

Postconditions: None.

Abnormal paths: a1. If Customer specifies no categories, makes or engine sizes, rather than retrieving the entire catalog, iCoot should not allow the search to be initiated.

U5:Log On. (Extended by U6, U8, U9, U10 and U12.)

Preconditions: Member has obtained a password from their local Store.

1. Member enters the membership number.
2. Member enters the password.
3. Since iCoot must enforce one logon for a Member, Member can choose to steal (invalidate and thus take over from) an existing session.
4. Member elects to log on.
5. Extend with U6, U8, U9, U10, U12.

Postconditions: Member is logged on.

Abnormal Paths: a1. If the membership number/password combination is incorrect, iCoot informs Member that one of the two is incorrect (for security, they're not told which one).

a2. If the membership number/password combination is correct, but Member is already logged on and they have not elected to steal, iCoot informs Member.

U6:View Member Details. (Extends U5.)

1. Member elects to view membership details.
2. Member is presented with membership details (name, address, status, amount owing, CreditCard details).
3. For security reasons, iCoot must display only the last four digits of the Member's CreditCard number.
4. iCoot informs Member that to correct details, they must contact their local Store.

Postconditions: Member has been presented with membership details.

U7:Make Reservation. (Extends U3.)

Preconditions: Customer is a Member who has logged on.

1. Member elects to reserve CarModel for the details on display.
2. iCoot asks Member for confirmation, issuing a warning that failure to collect a reserved CarModel will result in a fine.
3. Member confirms Reservation.
4. iCoot shows Member the Reservation number and indicates that Assistant will be in touch when a Car is available.
5. When an Assistant logs on to Coot, Assistant is given a list of Reservations that require action.

6. Assistant takes necessary action to progress Reservations (e.g. promoting to Collectable if a Car is available and moving the Car to the reserved area).

Postconditions: Any requested Reservations have been made.

Abnormal Paths: a1. If Member declines Reservation conditions, no Reservation is made.

U8:View Rentals. (Extends U5.)

Preconditions: None. Relationships: U5.

1. Member elects to view their Rentals.
2. iCoot presents Member with summary of each Car they currently have out for rent (including number plate and due date).

Postconditions: iCoot has presented Member with summaries of Cars currently rented.

U9:Change Password. (Extends U5.)

Preconditions: None.

1. Member elects to change password.
2. Member enters old password (which is obscured on screen).
3. Member enters new password (obscured).
4. Member enters new password again (for confirmation, also obscured).
5. Member initiates the change.
6. iCoot asks for confirmation (warning that new password must be memorable).
7. If Member confirms, password is changed.

Postconditions: Password has been changed.

Abnormal Paths: a1. If old password is incorrect or new passwords don't match, Member is informed (but not given details of the error, for security) and password is unchanged.

a2. If old passwords match but new password doesn't follow password rules (a mix of at least six letters and digits), Member is informed and password is unchanged.

U10: View Reservation objects. (Extends U5, extended by U11.)

Preconditions: None.

1. Member elects to view Reservations.
2. iCoot displays summaries of the Member's outstanding (unconcluded) Reservations (including number, state, timestamp and CarModel number).
3. Extend with U11.

Postconditions: Member has been presented with summary of outstanding Reservations.

U11:Cancel Reservation. (Extends U10.)

Preconditions: None.

1. Member selects a Reservation.
2. Member elects to cancel the Reservation.
3. iCoot asks for confirmation.

4. Member confirms that they wish to cancel the Reservation.
5. iCoot marks the Reservation as Concluded and updates Assistants' terminals accordingly.

Postconditions: Any canceled Reservations that were confirmed have been marked as Concluded.
Abnormal Paths: a1. If Member doesn't confirm a cancellation, iCoot takes no action.

U12:Log Off.
Preconditions: None.
1. Member elects to log off.
2. iCoot ends current session.
3. iCoot makes Member-only functions unavailable to Member.

Postconditions: Member is logged off.
Abnormal Paths: a1. For security reasons, a logged-on Member is logged off automatically if they do not interact with iCoot for ten minutes.

U13:Look for CarModels (Abstract, specialized by U1 and U4.)
Preconditions: None.
Postconditions: Customer has been presented with summaries of retrieved CarModels.

B.2.7 Supplementary Requirements

s1. The client applet must run in Java PlugIn 1.2 (and later versions).
s2. iCoot must be able to cope with a catalog of 100,000 CarModels.
s3. iCoot must be able to serve 1,000,000 Customers simultaneously with no significant degradation in performance.

B.2.8 Use Case Priorities

Below is the list of use case priorities for iCoot, with the scores that were used for the first increment.

- **Green**:
  - U1: Browse Index
  - U4: Search
  - U2: View Results
  - U3: View CarModel Details
  - U5: Log On
- **Amber**:
  - U12: Logoff.
  - U6: View Member Details
  - U7: Make a Reservation
  - U10: View Reservations

- **Red**:
  - U11: Cancel Reservation
  - U8: View Rentals
  - U9: Change Password

During the first increment, U6 was also completed. The other use cases were completed during the second increment.

B.3 ANALYSIS

This section documents the results of the analysis phase of the iCoot development, in terms of analysis class model, a state machine for a Reservation and use case realization (communication diagrams). The Reservation state machine also applies to the full Coot system. The class model includes a few pieces from the full Coot schema, such as NonMember and dateLost.

B.3.1 Class Diagram

The analysis class diagram for iCoot is shown in Figure B.12. Most of these classes also appear in the design class model (Section B.5), so their descriptions have been placed in the Glossary (Section B.8), to avoid repetition.

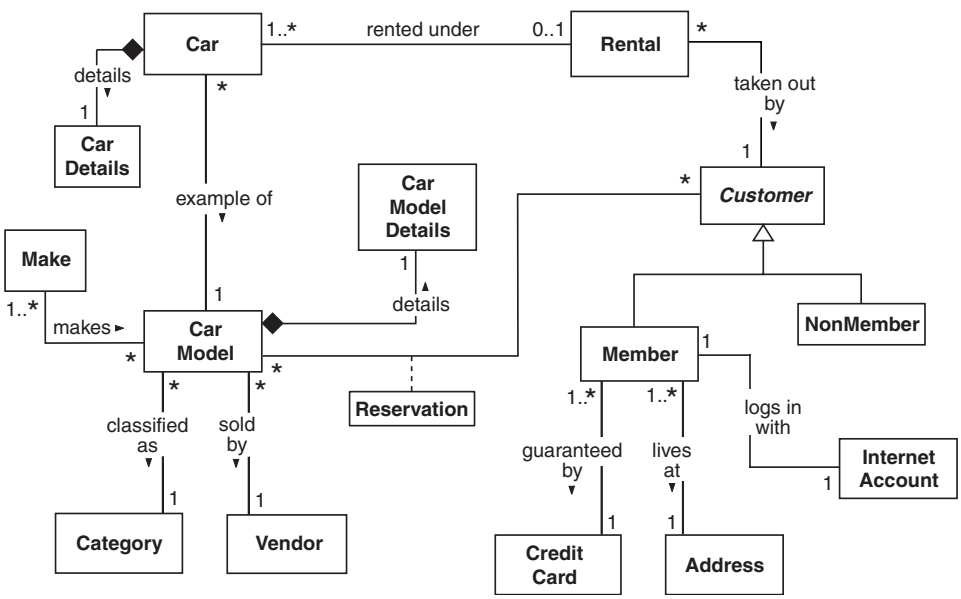

Figure B.12: Analysis class diagram

B.3.2 Attributes

The class attributes for iCoot are shown in Figure B.13. These attributes also appear in the design class model as fields, where they're given types and descriptions – refer to the design documentation (Section B.5) for details.

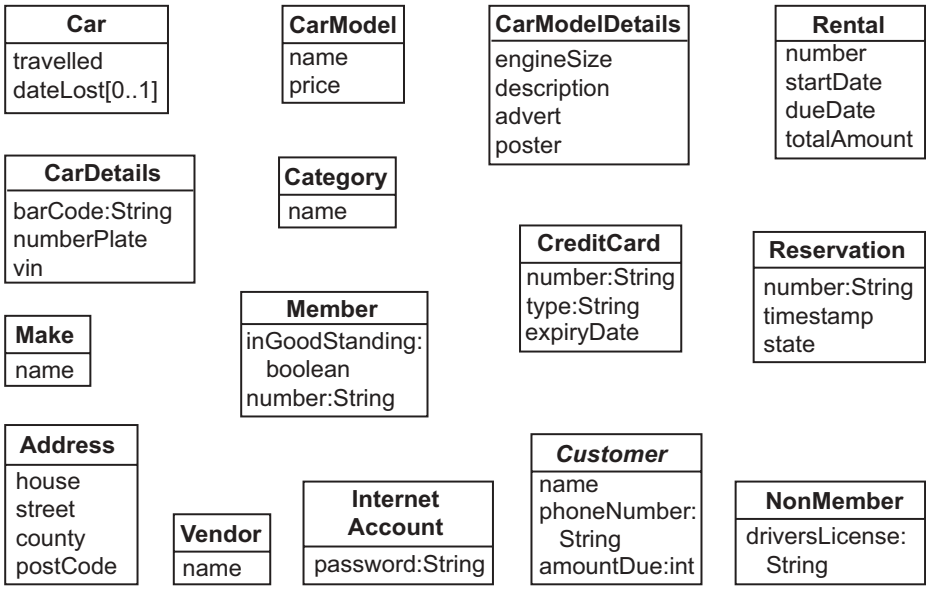

Figure B.13: Analysis attributes

B.3.3 Operation List

- CarModel:
  - getSummary() – Fetch a summary of the receiver, including model number and price.
  - getDetails() – Fetch the receiver's details, including makes, engine size, price, description, advert and poster.
- CarModelHome:
  - findByIndexHeading(h:String) – Search for CarModel objects under index heading h.
  - findByQuery(categories,makes,sizes) – Search for CarModel objects with Category from categories, a Make in makes and engine size in sizes.
- LogonController:
  - logon(n:String,p:String,s:boolean) – Log on the Member with membership number n and password p, specifying whether or not to steal any existing session with s.

  - changePassword(m:Member,o:String,n1:String,n2:String) – Change the password for m to n1, as long as n2 matches and the current password is o.
  - logoff() – Log off the logged-on Member.
- Member:
  - getPassword():String – Fetch the receiver's password.
  - isLoggedOn():boolean – True if the receiver is logged on.
  - logon() – Log the receiver on.
  - logoff() – Log the receiver off.
  - getDetails() – Fetch the receiver's details, including name, address, status, amount owing and (concealed) credit card details
  - setPassword(p:String) – Set the receiver's password to p.
- MemberHome: findByMembershipNumber(n:String):Member – Find the Member with membership number m.
- MemberUI:
  - search(categories,makes,sizes) – Search for CarModel objects with a Category from categories, a Make in makes and engine size in sizes.
  - index(h:String) – Search for CarModel objects under index heading h.
  - logon(n:String,p:String,s:boolean) – Log on the Member with membership number n, password p, specifying whether or not to steal any existing session with s.
  - setMember(m:Member) – Set the logged-on Member to m.
  - showMemberDetails() – Show details for the logged-on Member.
  - showRentals() – Show Rental objects for the logged-on Member.
  - showReservations() – Show unconcluded Reservation objects for the logged-on Member.
  - changePassword(o:String,n1:String,n2:String) – Change the password for the logged on Member to n1, as long as n2 matches and the current password is o.
  - confirmChange() – Confirm that the password really should be changed.
  - reserve(c:CarModel) – Reserve c for the logged-on Member.
  - confirmReserve() – Confirm that the Reservation really should be made.
  - cancel(r:Reservation) – Cancel r.
  - confirmCancel() – Confirm that the Reservation really should be canceled.
  - showDetails(c:CarModel) – Show details for c.
  - logoff() – Log off the logged-on Member.
- NonMemberUI:
  - search(categories,makes,sizes) – Search for CarModel objects with a Category from categories with a Make in makes and engine size in sizes.
  - index(h:String) – Search for CarModel objects under index heading h.
- Rental: getSummary() – Fetch a summary of the receiver, including number plate and due date.
- RentalHome: findByMember(m:Member – Fetch the Rental objects for member m.

- Reservation:
  - getSummary() – Fetch a summary of the receiver, including number, timestamp, state and CarModel.
  - getNumber() – Fetch the receiver's number.
  - setState(s) – Set the receiver's state to s.
- ReservationHome:
  - findByMember(m:Member) – Fetch the reservations for m.
  - create(c:CarModel,m:Member) – Reserve c for m, with the current date and time.

B.3.4 State Machine for a Reservation

Figure B.14 shows the state machine diagram for a Reservation, produced to model its complex life cycle. The accompanying state machine survey is:

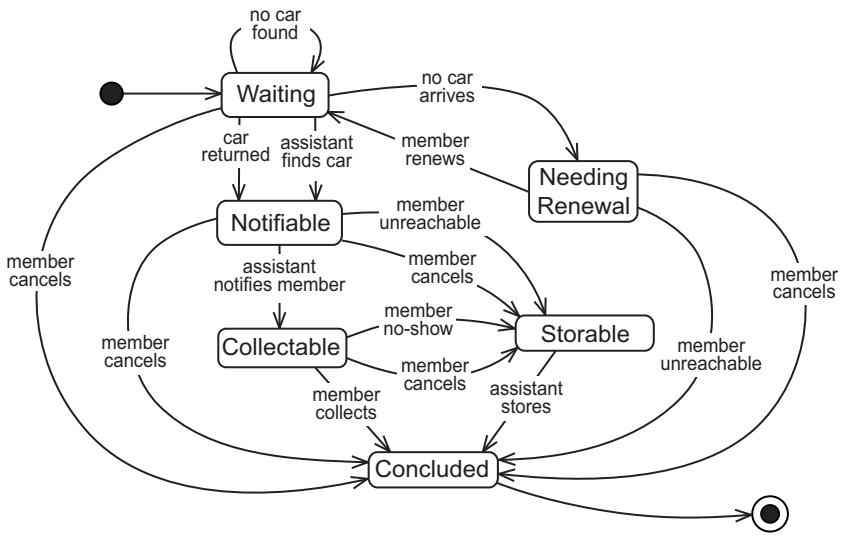

Figure B.14: State machine diagram for a Reservation

When a Member reserves a CarModel over the Internet, the Reservation is initially Waiting to be processed by an Assistant (this is so the Customer can make a Reservation without the intervention of an Assistant). The Reservation becomes Notifiable if, some time later, an Assistant finds a suitable unreserved Car in the display area of the car park, or if one is returned by a Customer. In this case, the Car is moved to the reserved area.

If no Car becomes available for a particular Reservation within a week, the Reservation becomes NeedingRenewal: the Member must be contacted, by phone or in person, so

that they can cancel the Reservation, or ask for it to be renewed for another week. If the Member cancels or can't be contacted within five days, the Reservation is Concluded.

Once a Reservation is Notifiable, the Member must be notified by an Assistant, in person or by phone, within three days; if the Customer can be reached, the Reservation is Collectable otherwise it becomes Displayable (a Car that was moved to the reserved area must be returned to the display area).

Once a Reservation is Collectable, the Member must collect the Car within three days: if they do collect, the Reservation is Concluded; otherwise, the Reservation becomes Displayable.

Once a Displayable Reservation's Car has been put back in the display area, the Reservation is Concluded.

At any time, the Member may cancel the Reservation over the Internet, by phone or in person.

The system will keep Assistants informed as to the state of current (not yet concluded) reservations, so that they can take appropriate action.

B.3.5 Use Case Realization

The communication diagrams for iCoot, verifying the analysis class model, are shown in Figures B.15 through B.26, one per system use case. Note the use of **guards** (arbitrary conditions in brackets), to specify conditional messages and * to specify iteration (iteration guards can be used to control iteration, but these would have made the diagrams more complex).

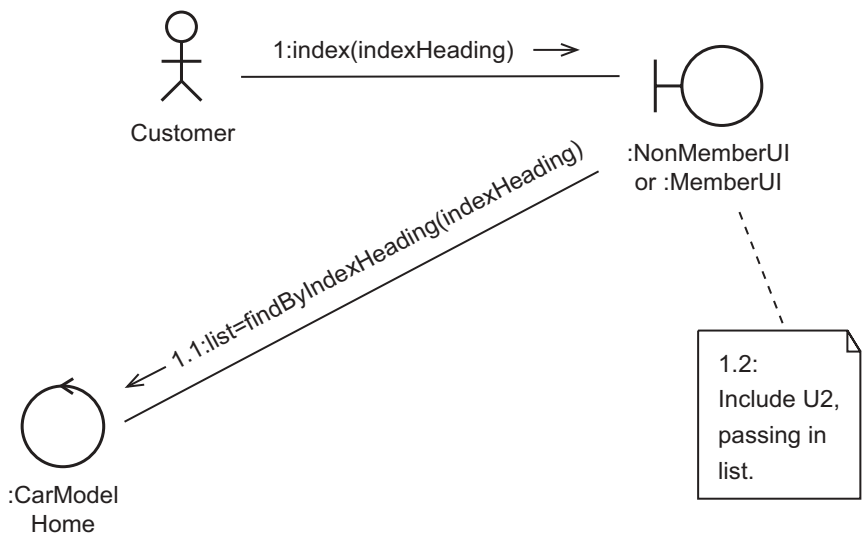

Figure B.15: Communication diagram for U1:Browse Index

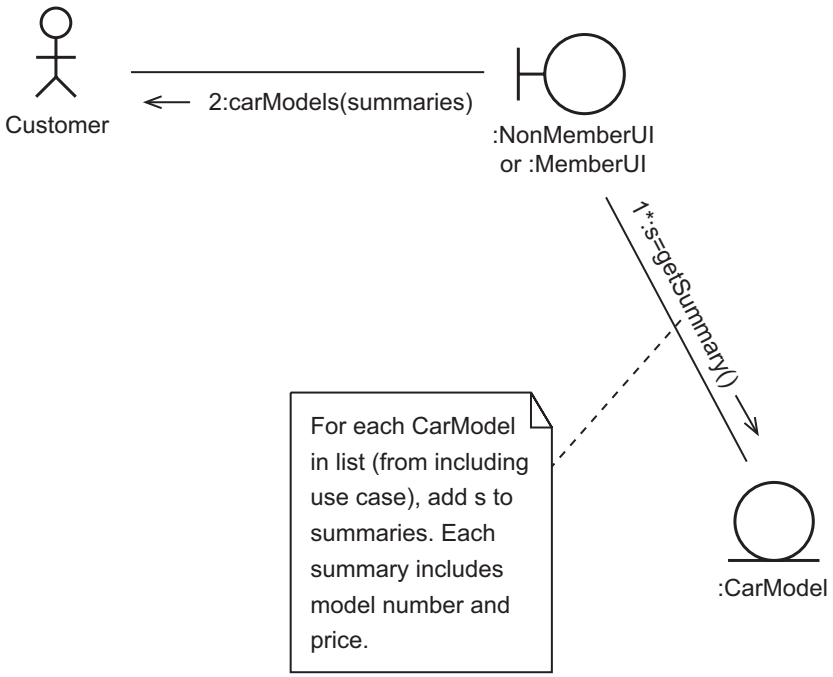

Figure B.16: Communication diagram for U2:View Results

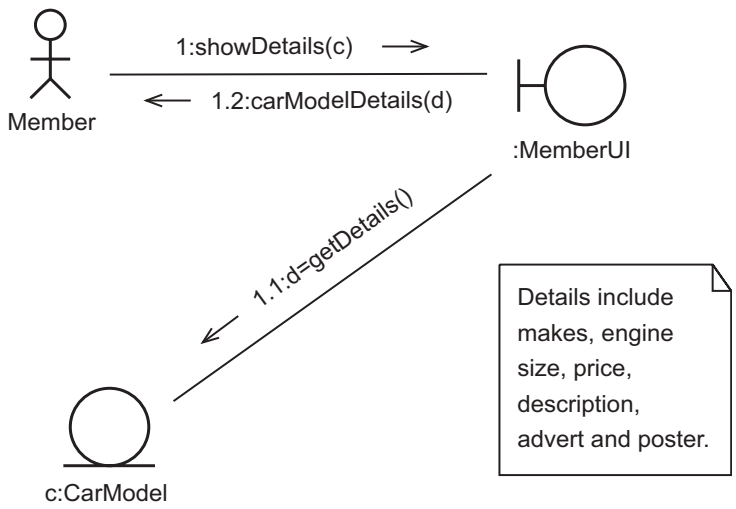

Figure B.17: Communication diagram for U3:View CarModel Details

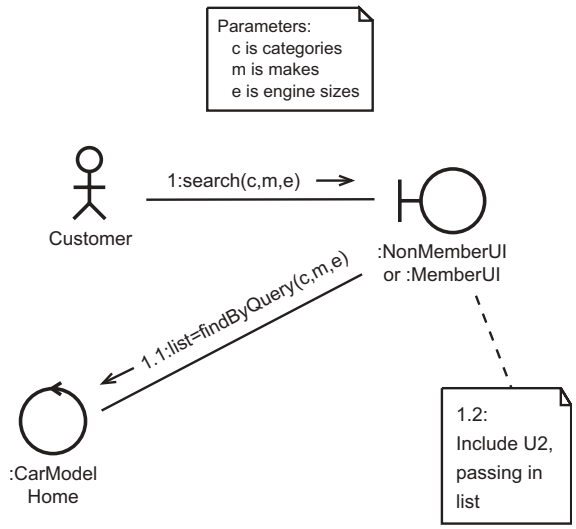

Figure B.18: Communication diagram for U4:Search

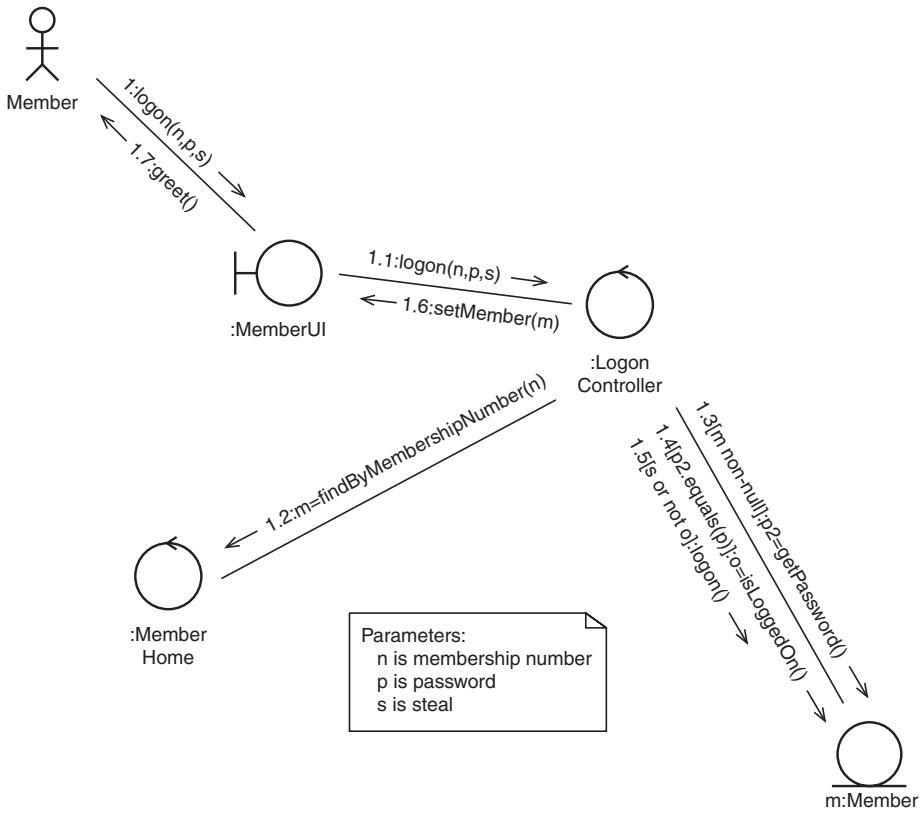

Figure B.19: Communication diagram for U5:Log On

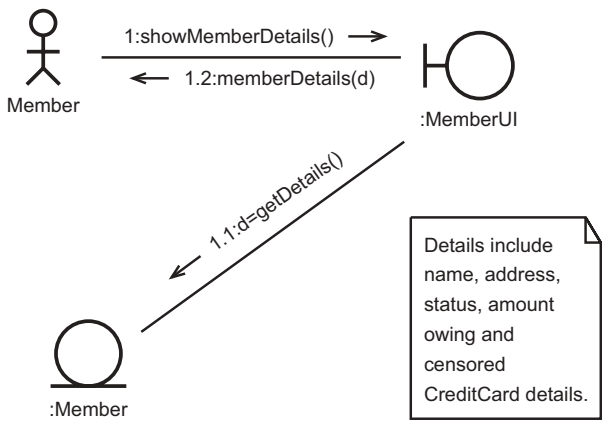

Figure B.20: Communication diagram for U6:View Member Details

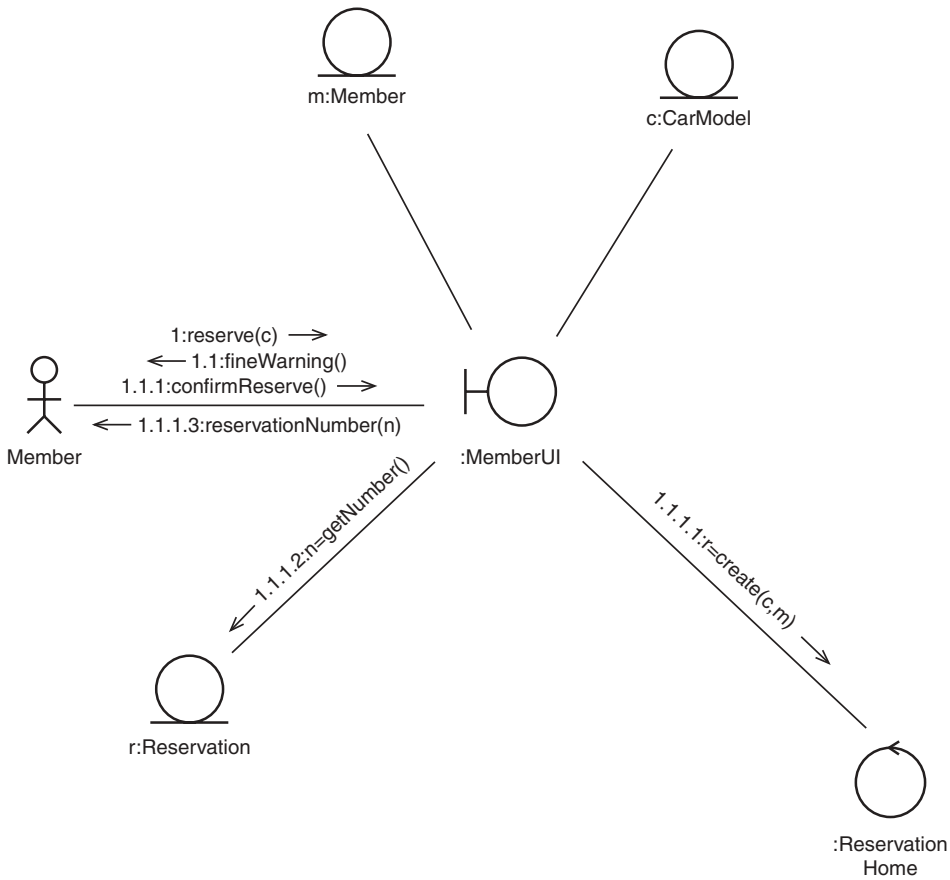

Figure B.21: Communication diagram for U7:Make Reservation

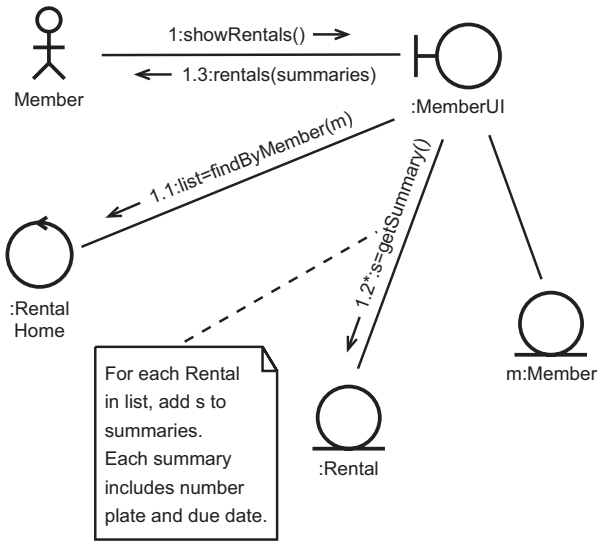

Figure B.22: Communication diagram for U8:View Rentals

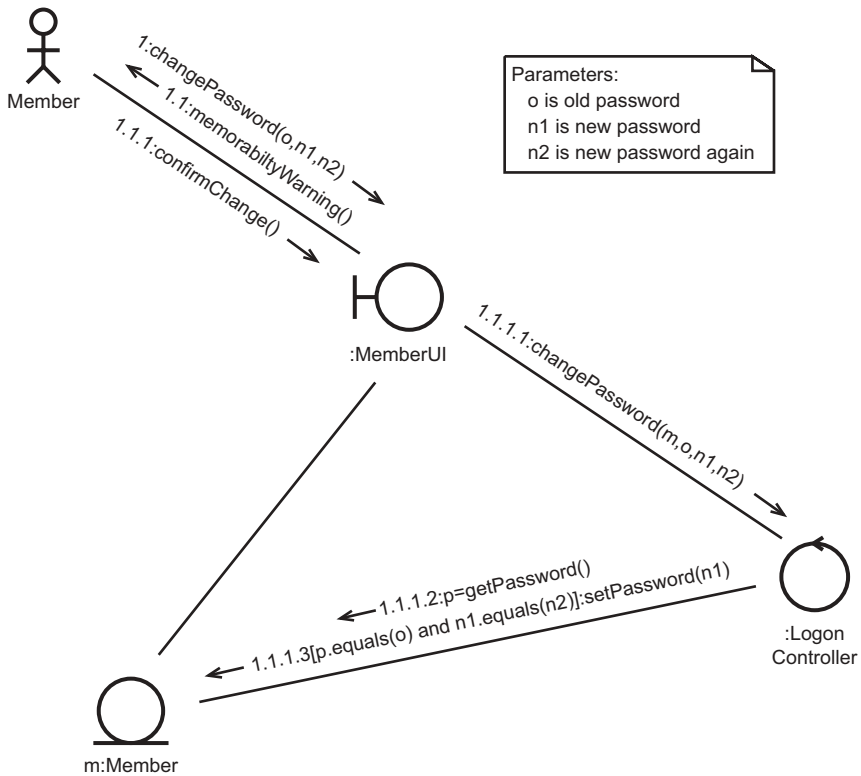

Figure B.23: Communication diagram for U9:Change Password

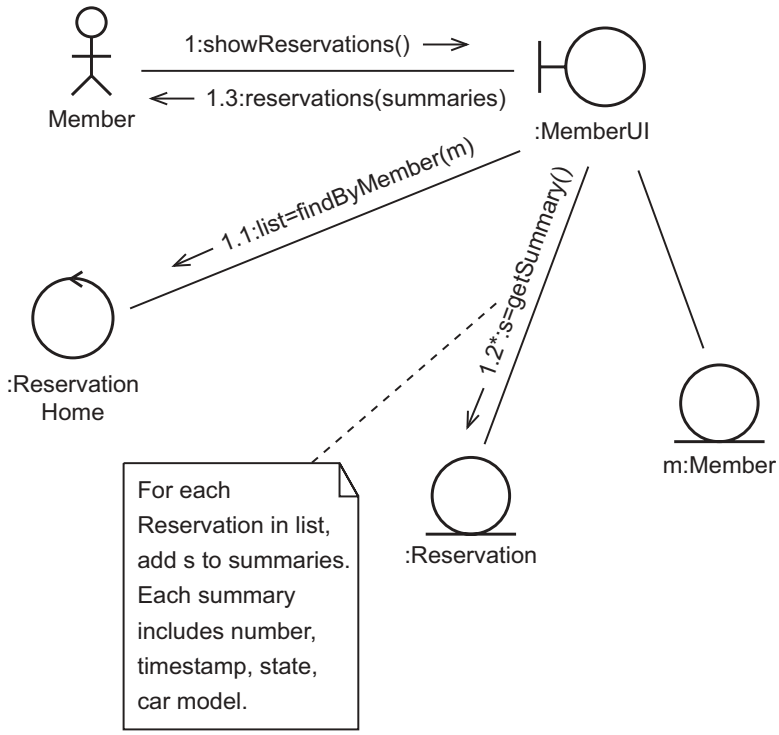

Figure B.24: Communication diagram for U10: View Reservations

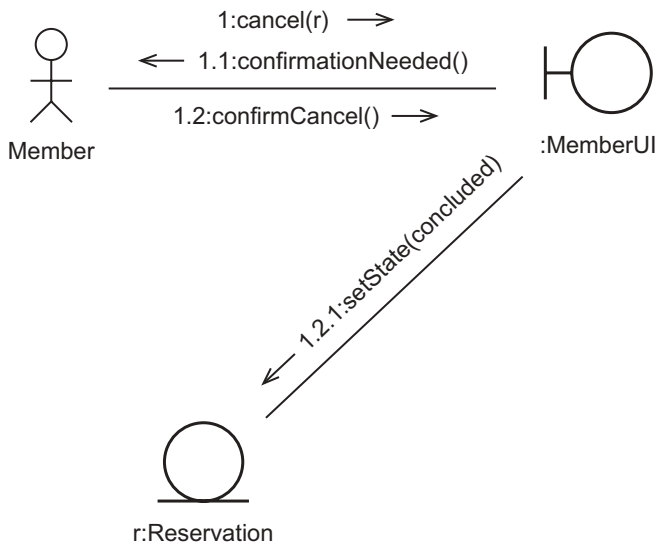

Figure B.25: Communication diagram for U11:Cancel Reservation

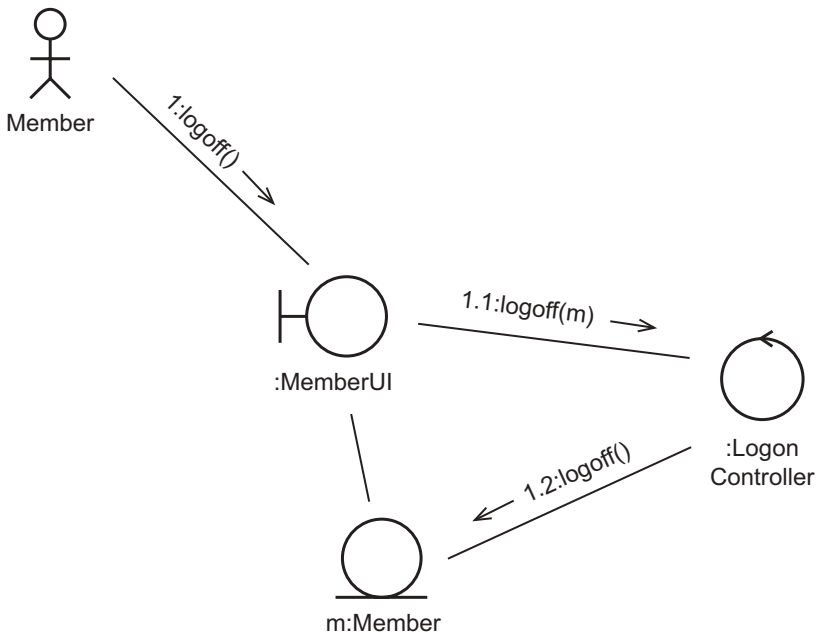

Figure B.26: Communication diagram for U12:Log Off

B.4 SYSTEM DESIGN

This section documents the results of system design carried out during the design phase of the iCoot development, in terms of technology choices, layers, packages, deployment diagram, security policy and concurrency policy.

B.4.1 Technology Choices

On the client side, the choice of technology is driven by convenience for the customer – we do not want customers to have to install any software in order to access our services. Also, we want them to be able to use any desktop machine, regardless of the operating system they have installed. The obvious choice for the client environment, therefore, is a web browser. Since the user interface must be interactive, in order to make a `Reservation` for example, we have to choose between technologies such as HTML/CGI, Java applets, ActiveX controls and Flash. Due to the need for portability (and client security), we can discount ActiveX controls. We would also prefer our customers to have almost-instant access to the user interface when browsing our site. This effectively discounts applets and Flash, both of which typically