CS 5704: Software Engineering

OO Methodologies

Dr. Pardha S. Pyla

Modeling the world

- Objects
- Classes
- Relationships
- Example

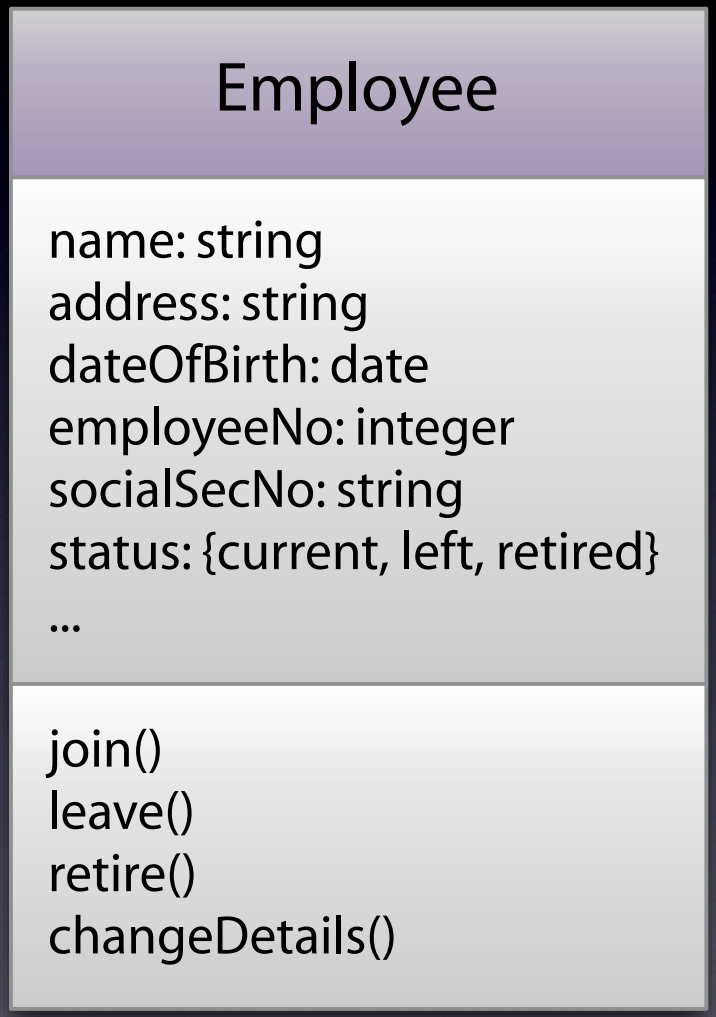
Objects

- An object is
  - “an entity that has a state”
  - “a defined set of operations that operate on that state”
- State is represented as a set of object attributes
- Operations provide services to other objects

Classes

- Abstractions to describe sets of objects
- “a type specification and a template for creating objects”
- Includes declarations of all attributes and operations

UML representation



OO development process

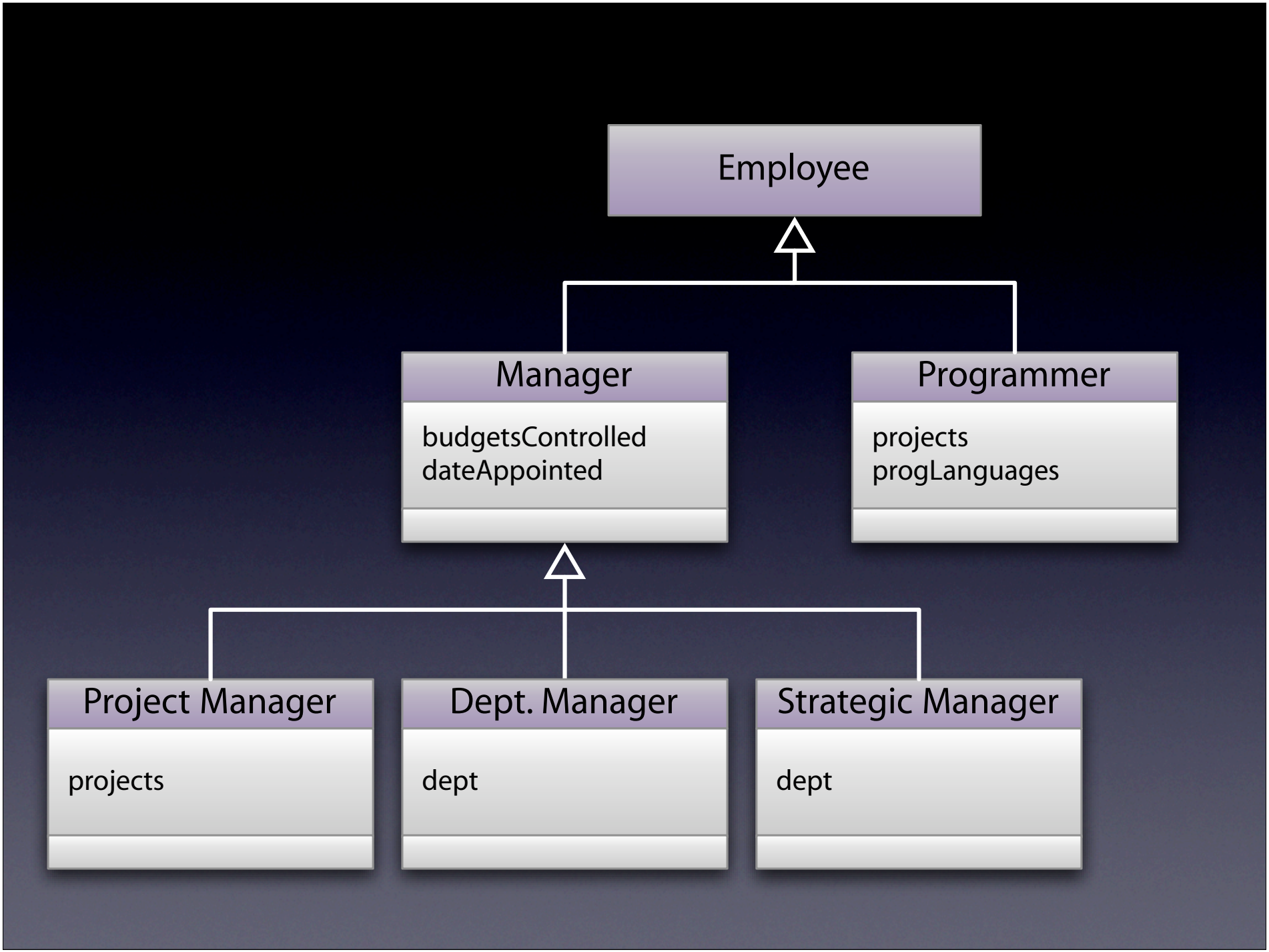
- Object-oriented analysis
  - modeling the problem domain using objects and associated operations
- Object-oriented design
  - modeling the solution domain using objects and associated operations
- Object-oriented programming
  - realizing the OO designs using OO languages

Overarching strategy

- Describe and extract relationships among objects in the problem domain
- Translate, elaborate, add to these descriptions
- Seamlessly move from analysis to design to implementation
  - Adding complexity

OO terminology

- Instantiation
- Inheritance
- Encapsulation
- Polymorphism (and dynamic binding)
- Association


Employee
Manager
budgetsControlled
dateAppointed
Programmer
projects
progLanguages
Project Manager
projects
Dept. Manager
dept
Strategic Manager
dept

Why OO?

- Abstracts complexity
  - any complex domain is an aggregate of different objects and associated services
- Contains change
  - description encapsulated at the granularity of a class
- Affords progressive specification
  - overview first, details later approach
- Facilitates reuse

Unified Modeling Language

- Language using graphical notations for modeling
- Aids in visualizing and specifying a software system
- Includes both static and behavioral structures
- Example

Use cases

- A narrative of interaction among actors and system
- Actors include users, external systems, databases
  - anyone or anything that interact with the system

Use case template - I

- Name (high-level description of the narrative)
- Use case diagram
- Actors
- Preconditions (what conditions must be true for this use case to start)
- Primary scenario (most probable series of steps that occur as part of this narrative)

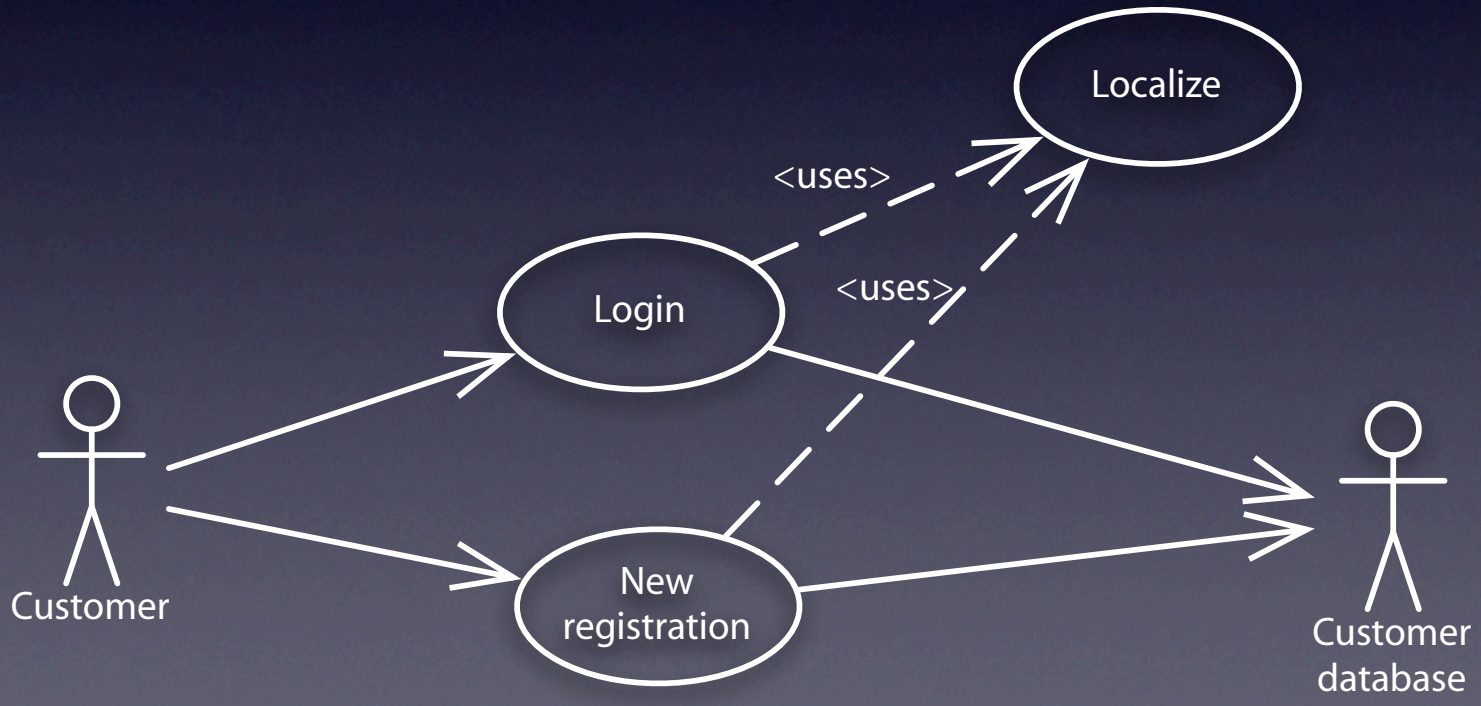
Use case template -2

- Alternative scenarios (less probable sequence of events that could occur as part of the narrative)
- Excepting scenarios (errors or exceptions to primary and alternative scenarios)
- Extension points (special steps that could occur beyond the regular sequence)
- “Used” use cases (other use cases used in current use case)
- Post conditions (state of system after use case executes)

Login use case - I

- Use case name: Login

In this use case, details of user authentication are covered. Based on the type of user, the access privileges are set for transactions that are possible after the execution of this use case



Login use case -2

- Precondition

  The user needs authentication and the login screen is displayed on the UI

- Primary scenario

1. The use case initiates when the user attempts to log into the system
2. The user enters the login name and password in the so marked fields
3. The user specifies security level of the computer from available options
4. The username and password combination is authenticated; if successful associated access privileges are set
5. The authentication success or failure is returned to the calling component
6. The user information is passed to “localize” use case
7. Use case ends

Login use case -3

- Secondary scenarios

  The use case is invoked during a browse, after a checkout, ... use cases

- Exception scenarios

1. The user types in an invalid username
2. The user types in an invalid password
3. User is new to the system and hence has no username or password
4. User gets disconnected during login due to network connection failure

- Extension points
  - None

Login use case -4

- "Used" use cases
  1. Localize
  2. Checkout
  3. Modify account
  4. ...
- Post condition
  - A function is selected, user explicitly logs out, or the login session times out

Login functional requirements - I

- Login screen description
  - The user shall be shown a Username field and a password field with a submit button using which the user shall be able to login.
  - There shall be two additional security level choices below the submit button. These security levels shall be “public/shared computer" and “personal computer".
  - The user shall be given the option to select one of these two options.
  - The choice shall be defaulted to public/shared option.

Login functional requirements -2

- Online account creation option
  - The system shall have and option on the login screen using which the user shall be able to create a new account. If this option is selected, the Registration Use Case shall be used to handle this request
- Password display description
  - The password entry field shall have character masking to prevent the display of entered password in plain text
- Authentication description
  - Upon submission of the login information by the user, the login system shall query the Customer Database and validate the entered username and password

Login functional requirements -3

- Authentication failure description
  - In case of a failed authentication because of wrong username or password, a feedback message shall be provided to the user. Moreover, the screen shall prompt the following messages: "register for a new login name and password if the user is new", "check the caps lock status on the keyboard and reenter password". The Login screen shall be shown again for the user to retry login.
- Post authentication description
  - In case of a successful authentication, the Localize Use Case shall be used. The user shall be shown only that information that his permissions allow him to view. In case of an alternate invocation of the Use Case, the system shall return result of authentication to calling module

Login functional requirements -4

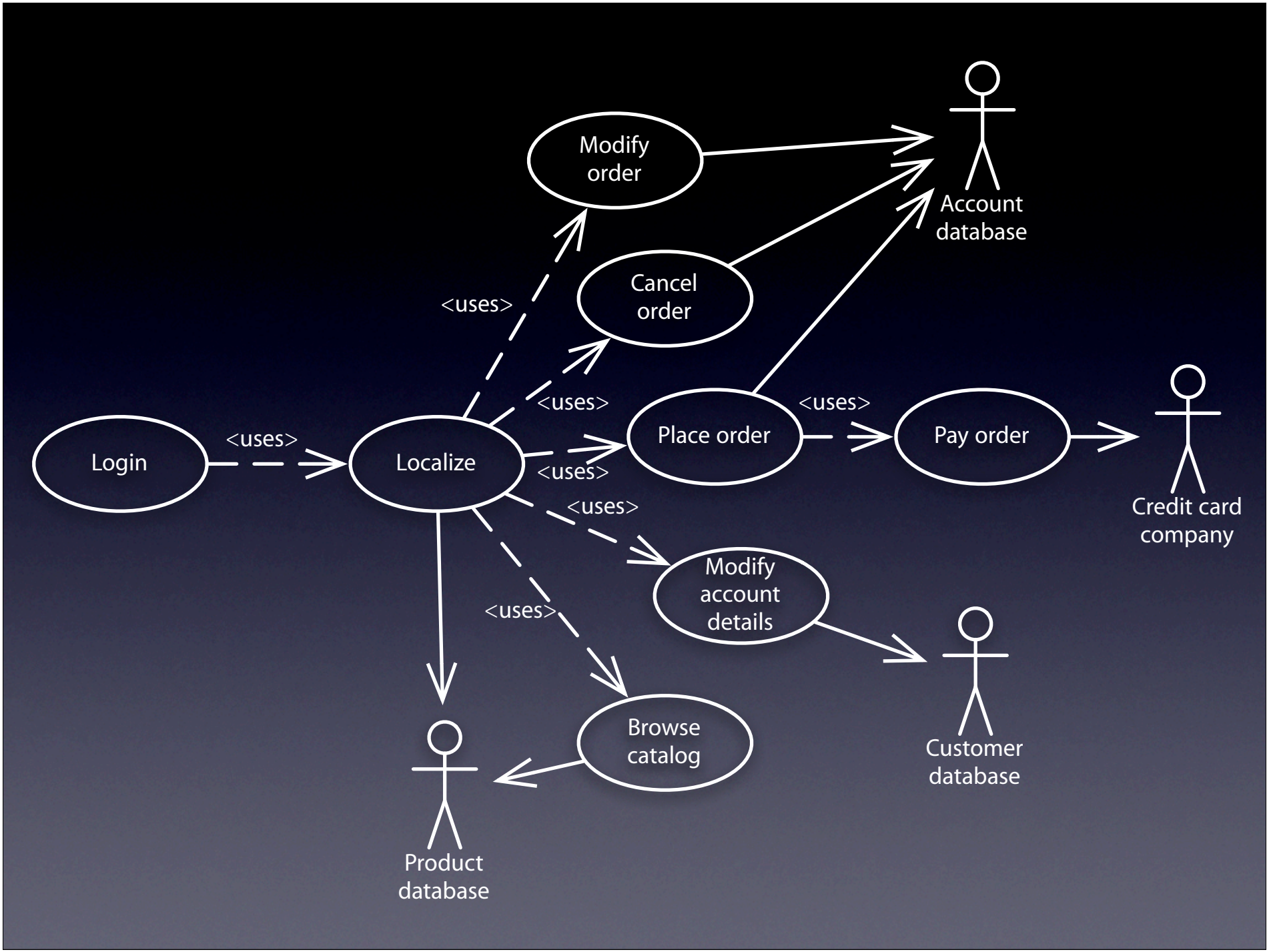
- Session list maintenance
  - As and when a user logs into the system a new entry shall be made into an internally maintained session list on the customer database. This list shall record the username, security level of the connection (shared/public vs. private), and the access privileges of the user.
- Localizing the user
  - Localize use case shall be used to automatically set the zone of interest to the user

Localize use case

- Use case name: Localize

  This Use Case describes how a user gets localized to a particular zone in the country. This is to show the user only information that is relevant to that zone. This use case also facilitates providing specialized services to users based on locality.

- Actors: Customer


Modify order
Account database
Cancel order
<uses>
<uses>
<uses>
Login
<uses>
Localize
Place order
<uses>
Pay order
Credit card company
<uses>
<uses>
Modify account details
Customer database
Browse catalog
Product database

Localize use case -2

- Precondition: User is able to access web site and/ or logged in
- Primary Scenario
  - The use case starts with the user logging into the website successfully
  - The system queries the customer database to identify user details
  - The system then localizes the user by querying the customer database to determine user address and computing location zone

Localize use case -3

- Alternate scenario
  - The user accesses the website without logging in.
  - The system provides user list of countries, and based on selection, corresponding states, and based on selection, the available zones
  - The banner use case is used to update the UI to show user zone status
- Exception scenario
  - The user address does not fall under any recognized zones

Localize use case -4

- Extension points
  - If any special sales, discounts, rebates are marked for the user selected zone, the system automatically gives a visual prompt to the user about those options
- “Used” use cases
  - ...
- Post conditions
  - The user is localized to a particular zone
- Functional requirements ...

pause

UML Diagrams

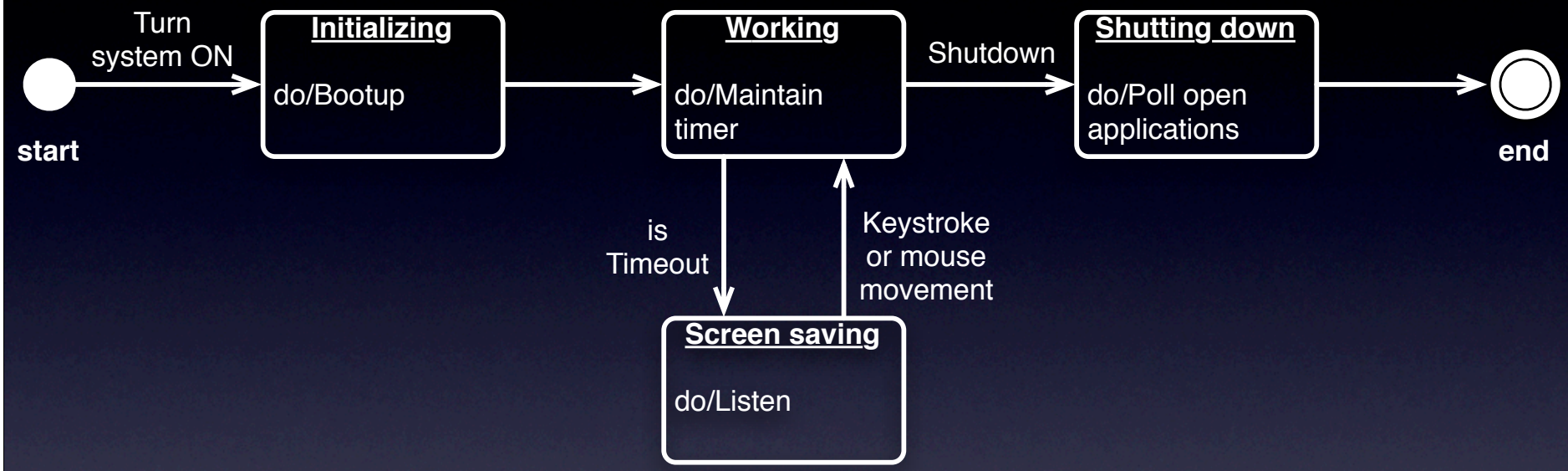
- Structural diagrams
  - Class diagram
  - Object diagram
  - Component diagram
  - Deployment diagram

UML Diagrams -2

- Behavioral diagrams
  - Use case diagram
  - Sequence diagram
  - Collaboration diagram
  - Statechart diagram
  - Activity diagram

State transition diagrams

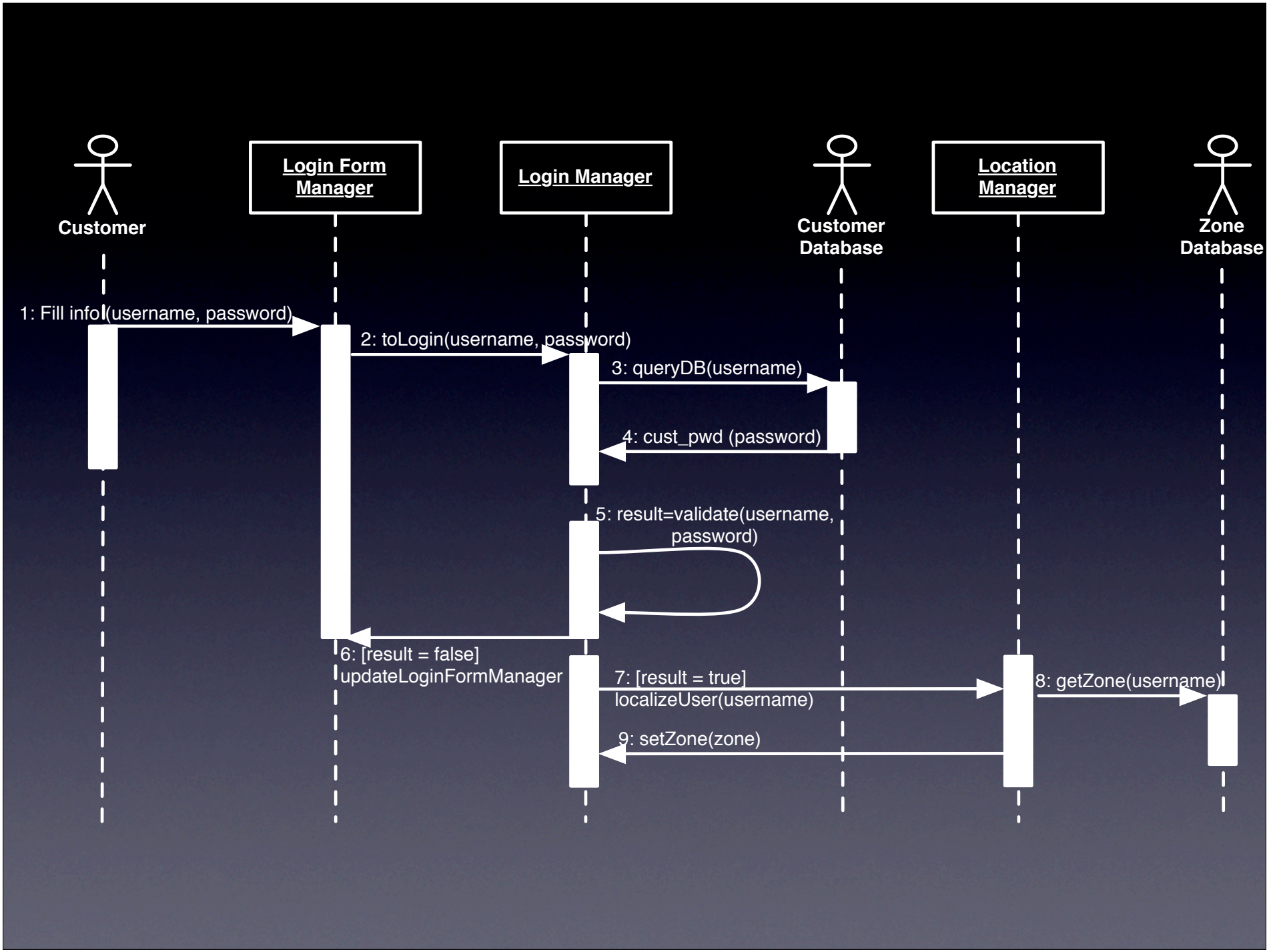
- Life of an object, as it transitions from one state to another
- Results of the changes the object goes through
- Important for developers to visualize what happens as the object changes states


Turn
system ON
start
Initializing
do/Bootup
Working
do/Maintain
timer
Shutdown
Shutting down
do/Poll open
applications
end
is
Timeout
Keystroke
or mouse
movement
Screen saving
do/Listen

* Schmuller

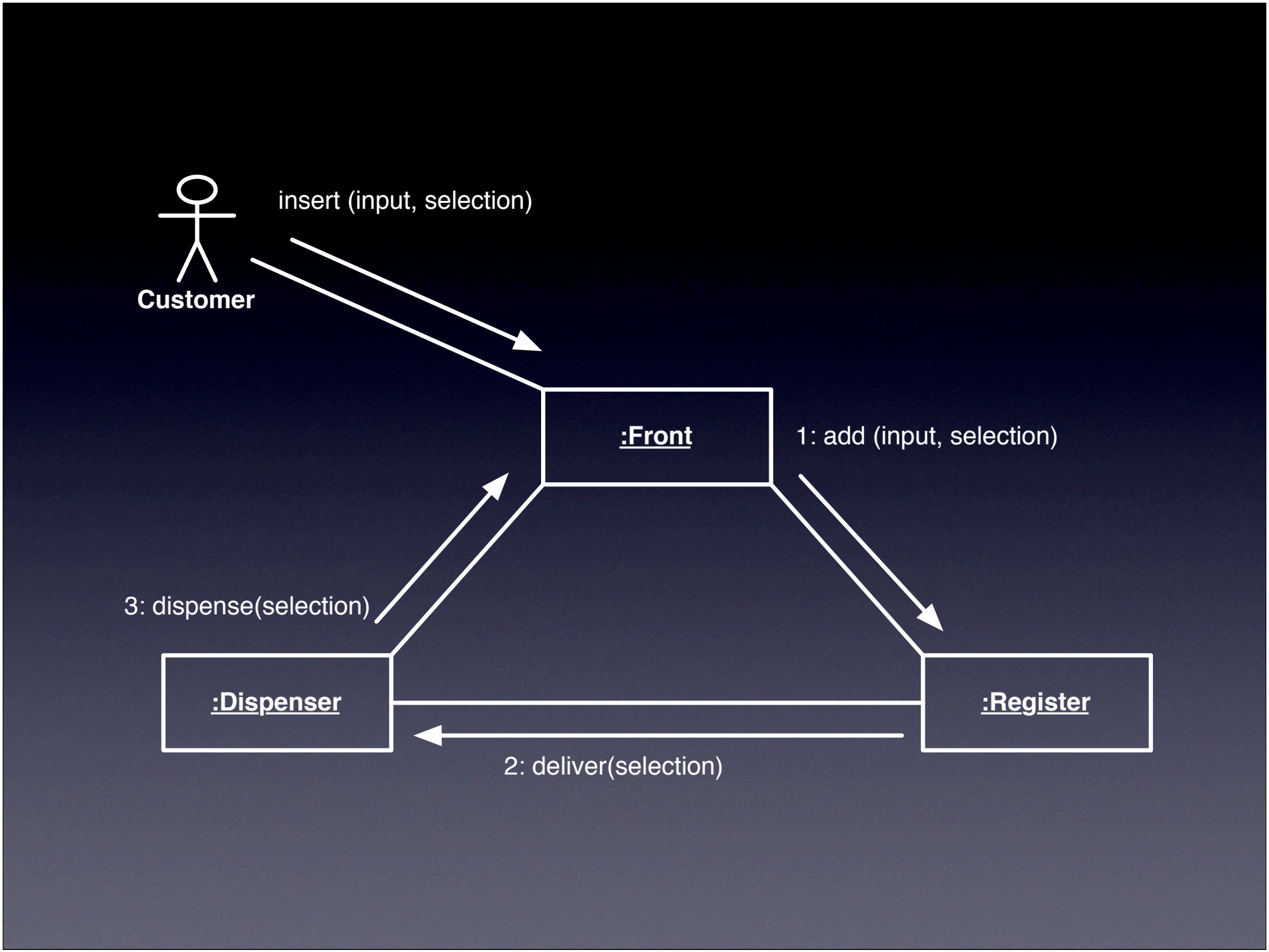
Sequence diagrams

- From state of one object in *state diagrams* to communication among objects in time
- Objects, their lifelines, and execution of operations


Customer
Login Form Manager
Login Manager
Customer Database
Location Manager
Zone Database
1: Fill info(username, password)
2: toLogin(username, password)
3: queryDB(username)
4: cust_pwd (password)
5: result=validate(username, password)
6: [result = false] updateLoginFormManager
7: [result = true] localizeUser(username)
8: getZone(username)
9: setZone(zone)

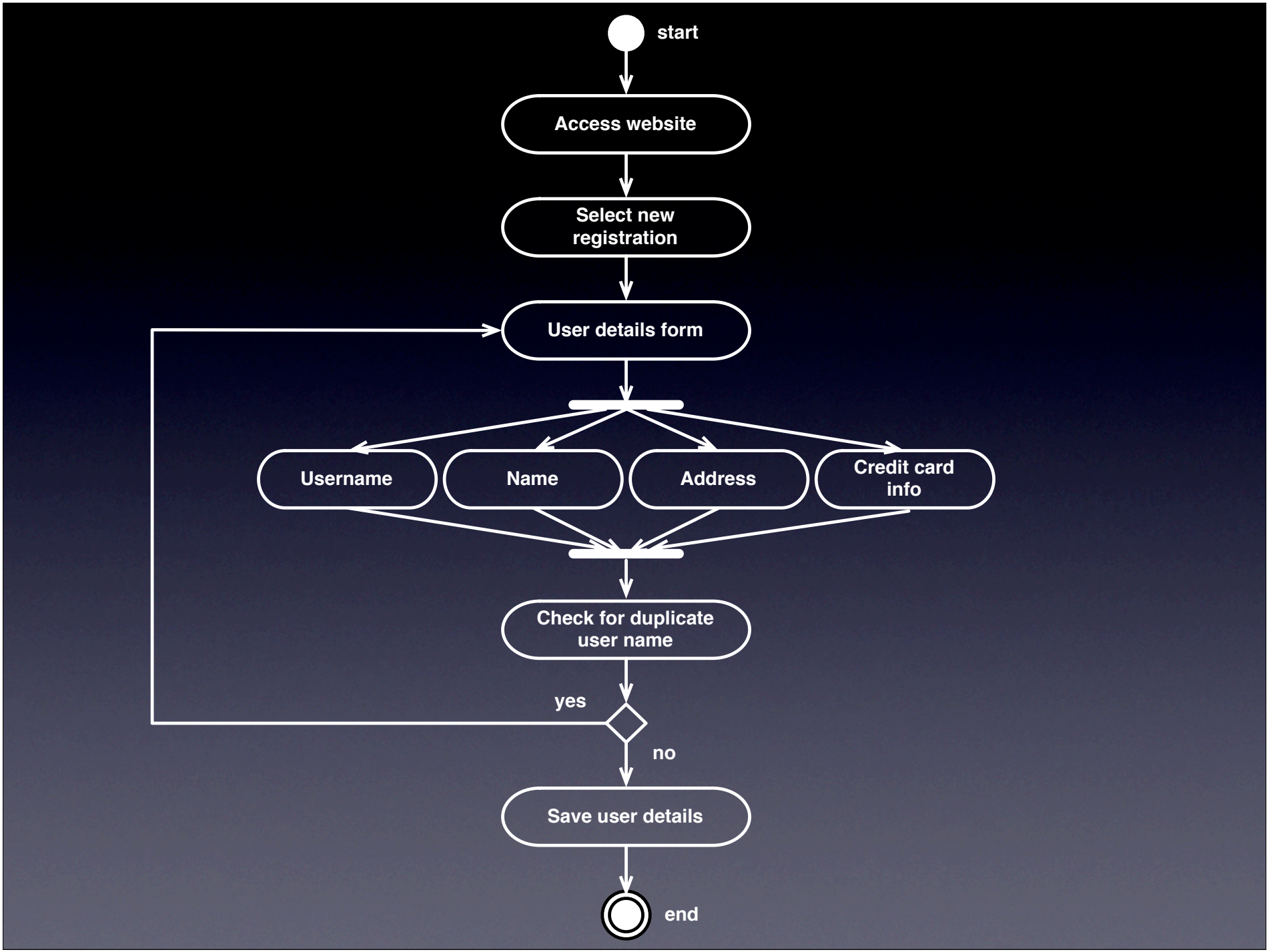
Collaboration diagrams

- Semantically equivalent to sequence diagrams
- Sequence diagrams represent temporal aspects
- Collaboration diagrams represent context and overall organization of object interactions
- Also shows messages and arguments between objects


insert (input, selection)
Customer
:Front
1: add (input, selection)
3: dispense(selection)
:Dispenser
:Register
2: deliver(selection)

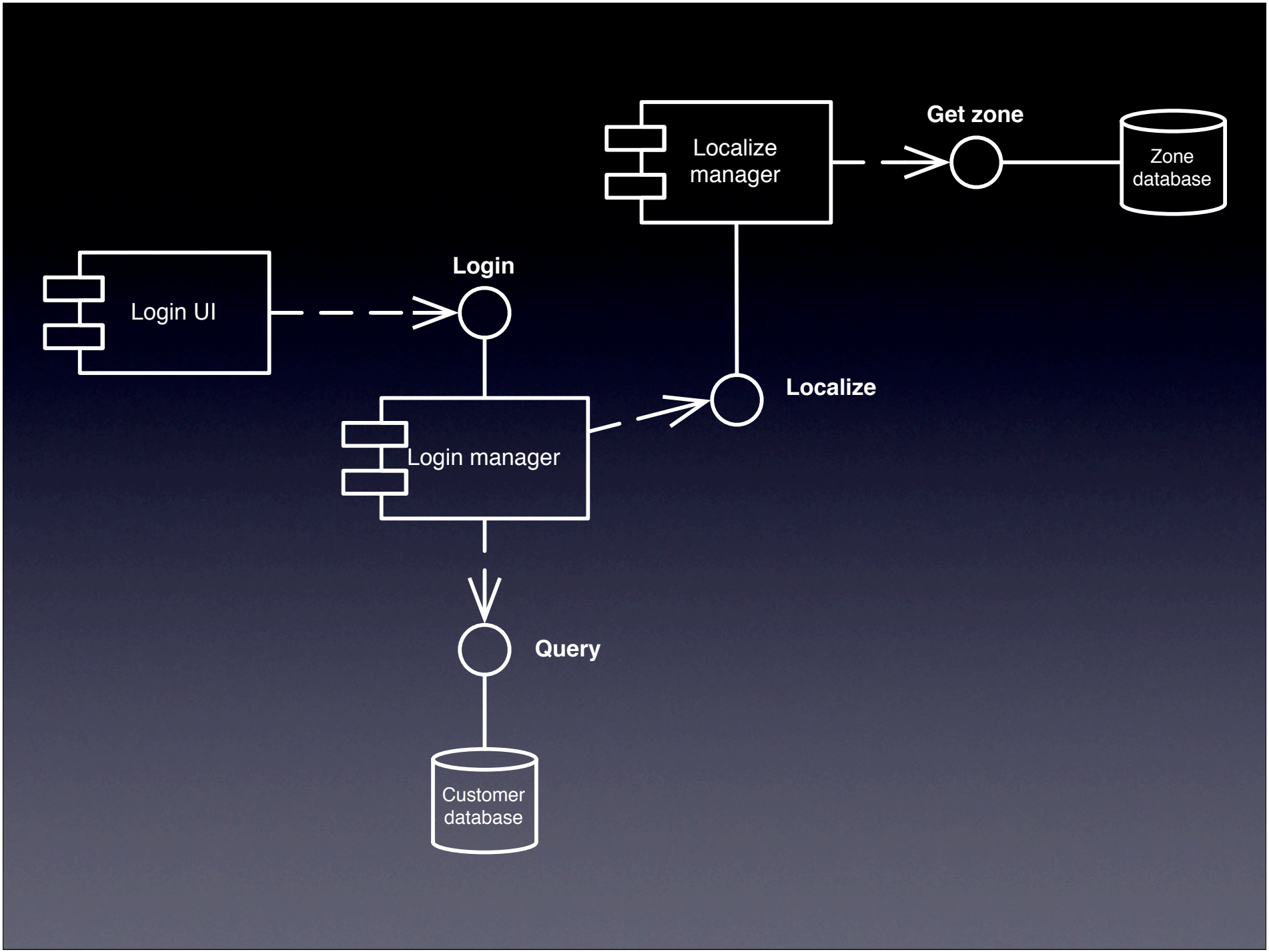
Activity diagrams

- A simplified look at the sequence of events
  - flowchart
- Shows decision points, if any


start
Access website
Select new registration
User details form
Username
Name
Address
Credit card info
Check for duplicate user name
yes
no
Save user details
end

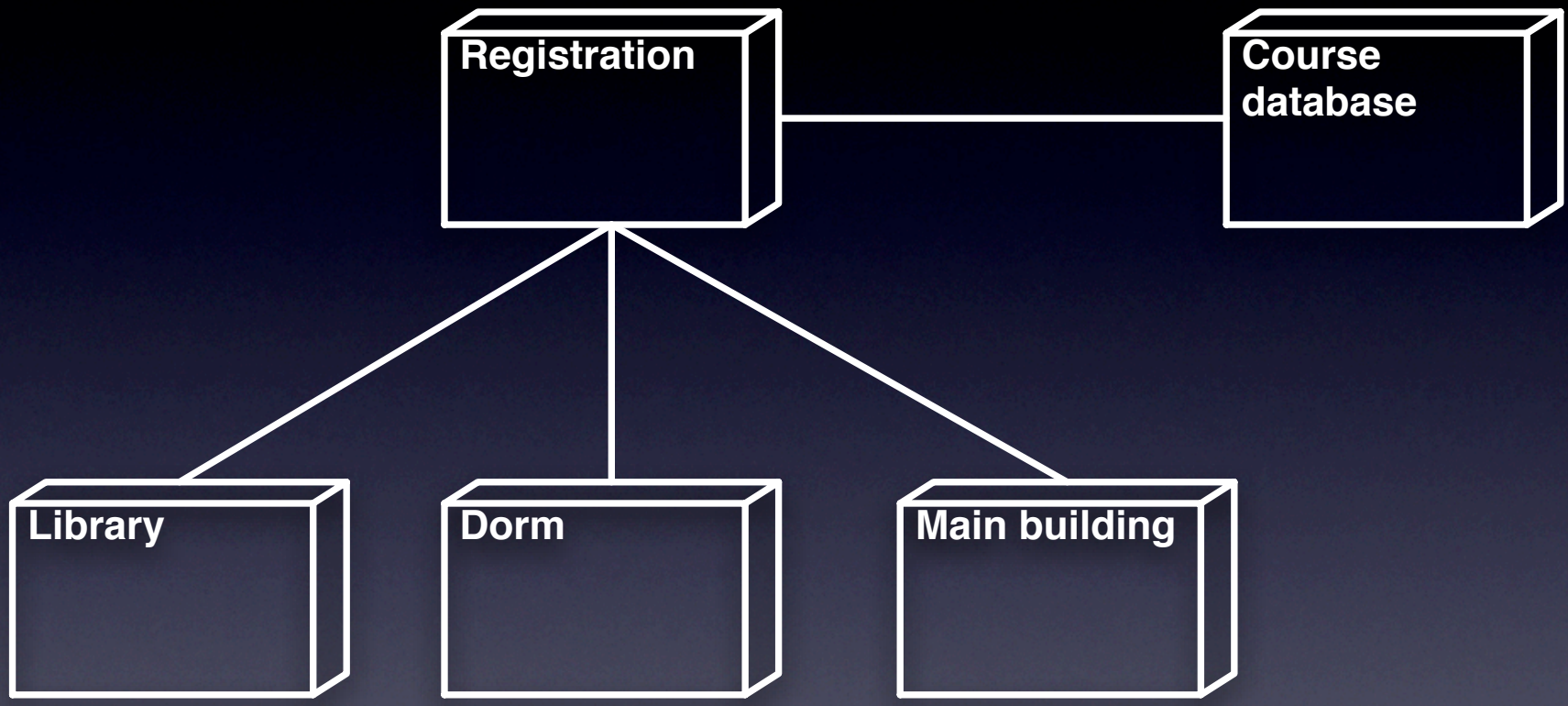
Component diagrams

- A physical part of the system, exists in a system (not an analyst model)
  - source code component
  - run-time/executable component
- Shows structure in finished system
- Illustrates interfaces and relationships among different components


Localize manager
Get zone
Zone database
Login UI
Login
Localize
Login manager
Query
Customer database

Deployment diagrams

- Illustrates “run-time processing elements and the software processes living on them”
  - Components across the enterprise
  - Includes hardware and other physical entities



* Rational Software Corporation

end