Umm Al-Qura University

6001333-3 Human Computer Interaction

Lecture 4 (Week 3)

- User Profile
- Persona
- Use cases
- Scenario

Email: arzahrani@uqu.edu.sa

- **Requirement analysis:** One of the main types of requirements is User requirement.

- User requirements seek to find out who are the users and what they do (tasks).

- To answer these questions, we need to conduct *users and task analysis*.

User analysis

- User analysis: who is the users?

1. Identifying user profile
2. Creating a persona

Identifying the user profile

- User profile is a description of a system's target population attributes.

- A user profile will help you **understand who you are building your product for.**

- How to create a user profile:
  1. Finding information to build your user profile
  2. Understanding the types of users
  3. Creating the user profile

1. Finding information to build your use profile

- If you know who the real users are
  - Give a questionnaire directly to the target users

- If you are unsure about who the real users are; **interview** knowledgeable people in the organization, e.g. managers, product development managers, customer support. Information can also be obtained from market research, competitive analysis, and web metrics analysis.

- What if it is a new system that is never used before?

2. Understanding the types of users

- All the users are ...

OHN CUSACK · CAMERON DIAZ · CATHERINE KEENE
BEING JOHN MALKOVICH
A FILM DIRECTED BY SPIKE JONZE

EQUAL ??

LIKE ME ??

2. Understanding the types of users

- Define what you mean by "user."

- Most people consider the individuals who will interact directly with the product as their users, ***but*** you may need to consider other individuals as well:

stakeholders

- The manager of your direct user
- The system administrator who configures the product for the direct user
- People who receive information from the system

2. Understanding the types of users

- Try to categorize your users into one of two categories:
  - **Primary** users are those individuals who work regularly or directly with the product.
  - **Secondary** users will use the product infrequently or through an intermediary.

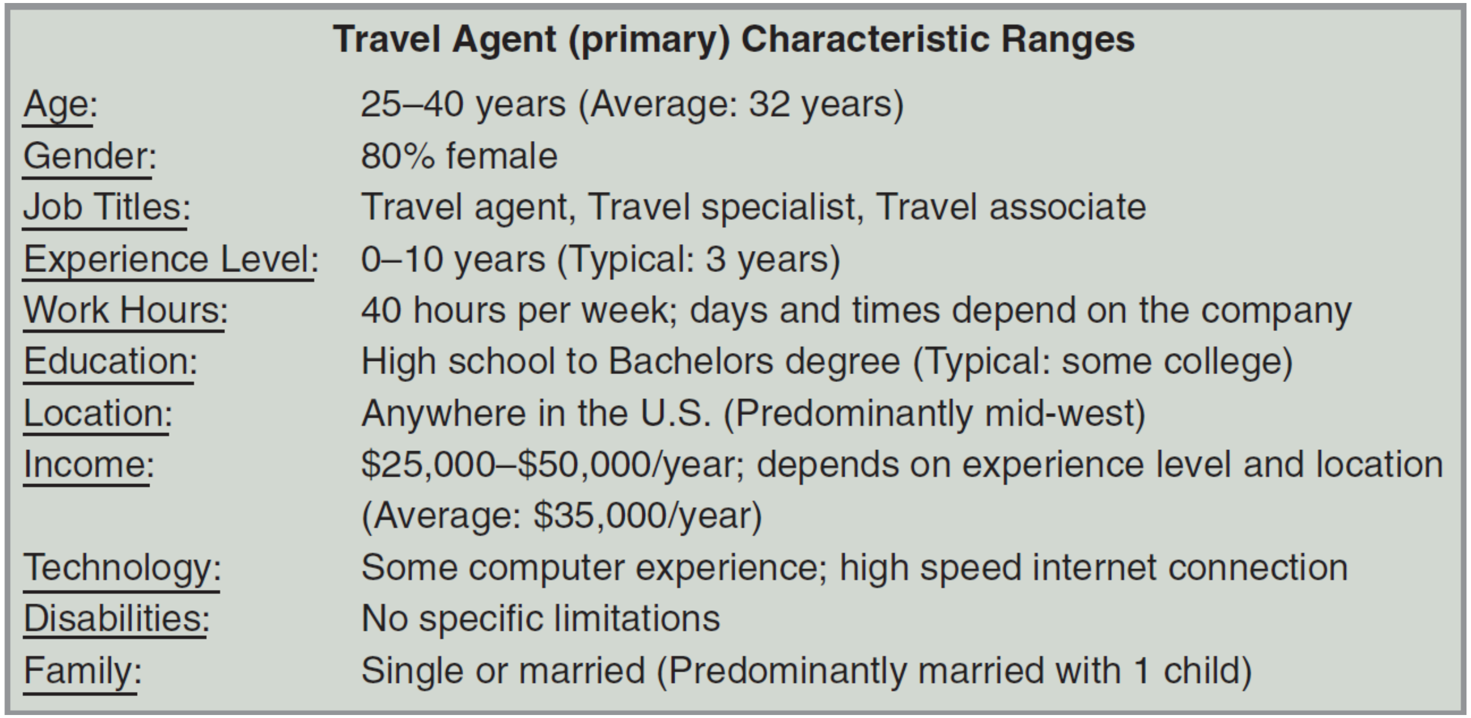
- This does not mean that you have to conduct user analysis with the secondary users, but you should at least know who they are.

3. Creating the user profile

- **Demographic characteristics.** Age, gender, location, socio-economic status
- **Occupation experience.** Current job title, years at the company, years of experience at that position, responsibilities, previous jobs and job titles
- **Company information.** Company size, industry
- **Education**. Degree, major, courses taken
- **Computer experience**. Computer skills, years of experience
- **Specific product experience**. Experience with competitors' products or other domain-specific products, usage trends
- **Tasks**. Primary tasks, secondary tasks
- **Domain knowledge**. The users' understanding of the product area
- **Technology available**. Computer hardware (monitor size, computing speed, etc.), software, other tools typically used
- **Attitudes and values**. Product preferences, fear of technology, etc.
- **Learning style**. Visual learner, audio learner, etc.
- **Criticality of errors**. In general, the possible consequences of a user's error.

3. Creating the user profile

Example:



Travel Agent (primary) Characteristic Ranges

Age:	25–40 years (Average: 32 years)
Gender:	80% female
Job Titles:	Travel agent, Travel specialist, Travel associate
Experience Level:	0–10 years (Typical: 3 years)
Work Hours:	40 hours per week; days and times depend on the company
Education:	High school to Bachelors degree (Typical: some college)
Location:	Anywhere in the U.S. (Predominantly mid-west)
Income:	$25,000–$50,000/year; depends on experience level and location (Average: $35,000/year)
Technology:	Some computer experience; high speed internet connection
Disabilities:	No specific limitations
Family:	Single or married (Predominantly married with 1 child)

Exercise !

- Develop a user profile for University library website (primary and secondary users)

Personas

Personas

- Once you have developed a user profile, you can develop *personas*.

- Ancient Greek: **Persona = Mask**

- In large open air theatres in ancient greek , classical masks (personas) were able to bring a character’s face closer to the audience.

Personas

- *A persona is* (exemplars of your end user) *a rich picture of an imaginary person* who represents your core user group.

- Personas are **not real people**, **but they represent them** throughout the design process.

- A persona is a representation of a subset of your users who show similar behaviours and patterns in the way they use your system.

Benefits of Personas

1. Personas are designed to give the development team a shared understanding of the real users in terms of goals, capabilities and context.

2. All team members think about the same persona
   - instead of each individual working towards his/her own vision of who the end user is

3. Personas give your users life and help team members feel connected emotionally to them

Benefits of Personas

4. A persona can be used in meetings as a discussion tool
    - e.g., "Mary would never use that feature"

5. Personas can also help new team members quickly learn who the end user is

To Be Aware of (when Creating Personas)

1. Developing multiple personas for each user type will help to cover the range of characteristics for each user type. However, should keep the set of personas manageable. It is a balancing act.
    - If you have too many personas to represent one user type, they will simply blur together in everyone’s mind and diminish their benefits.
    - **Three** primary personas is a common recommendation

To Be Aware of (when Creating Personas)

2. Not all users use all parts of a product or system. Therefore, it is unrealistic to assume that the same persona will work for all parts of your product

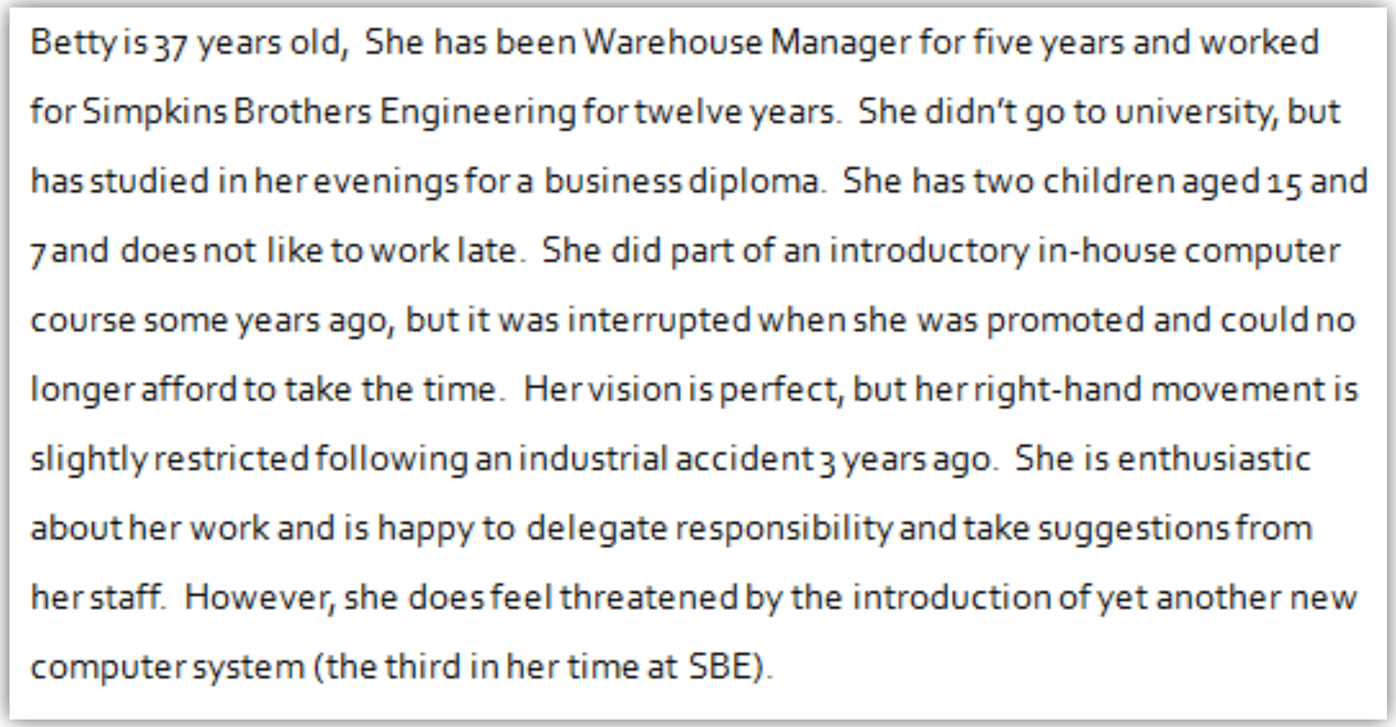
3. Not focus only on the “best” or “most experienced” users...Consider a **range of users** to ensure that the product will work for **80% of potential population**

Creating a Persona

- Personas generally include the following key pieces of information:
    1. Fictional name
    2. Persona Group (i.e. web manager)
    3. Major responsibilities
    4. Demographics such as age, education, ethnicity, and family status
    5. The goals they are trying to complete using the site
    6. Their physical, social, and technological environment

- Details will come from the information in your user profile.

Example



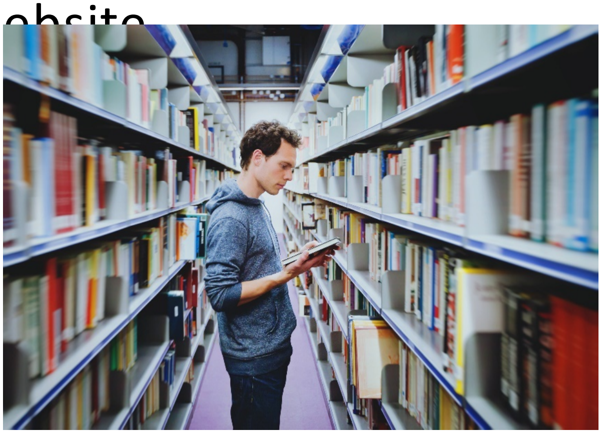
Betty is 37 years old, She has been Warehouse Manager for five years and worked for Simpkins Brothers Engineering for twelve years. She didn’t go to university, but has studied in her evenings for a business diploma. She has two children aged 15 and 7 and does not like to work late. She did part of an introductory in-house computer course some years ago, but it was interrupted when she was promoted and could no longer afford to take the time. Her vision is perfect, but her right-hand movement is slightly restricted following an industrial accident 3 years ago. She is enthusiastic about her work and is happy to delegate responsibility and take suggestions from her staff. However, she does feel threatened by the introduction of yet another new computer system (the third in her time at SBE).

Example

- What we learnt from the previous persona?
  - Job title: She is a manager
  - Education level: diploma
  - Married and has children
  - Problem with right hand
  - Afraid of updates

Exercise !

- Develop a persona for a university student (primary user) who is using a University library website



Task description

Task Description

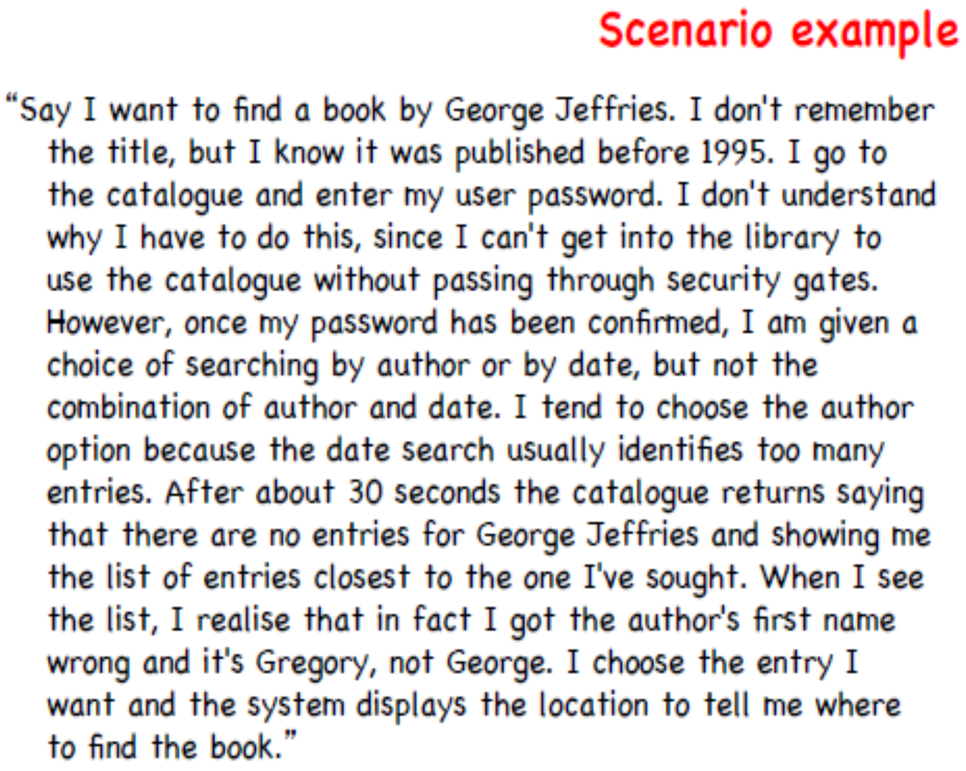
- Once you have developed a user profile and persona, you can start describing the tasks performed on the system under development using one of the following:

1. Scenarios
2. Use cases

Scenarios

- Are stories for design: rich stories of interaction
- Are informal narrative descriptions
- Describe human activities in stories
- Allow exploration of needs, requirements and contexts
- Language is that of the users
- Level of detail can vary

Scenarios



Scenario example

“Say I want to find a book by George Jeffries. I don't remember the title, but I know it was published before 1995. I go to the catalogue and enter my user password. I don't understand why I have to do this, since I can't get into the library to use the catalogue without passing through security gates. However, once my password has been confirmed, I am given a choice of searching by author or by date, but not the combination of author and date. I tend to choose the author option because the date search usually identifies too many entries. After about 30 seconds the catalogue returns saying that there are no entries for George Jeffries and showing me the list of entries closest to the one I've sought. When I see the list, I realise that in fact I got the author's first name wrong and it's Gregory, not George. I choose the entry I want and the system displays the location to tell me where to find the book.”

Scenarios

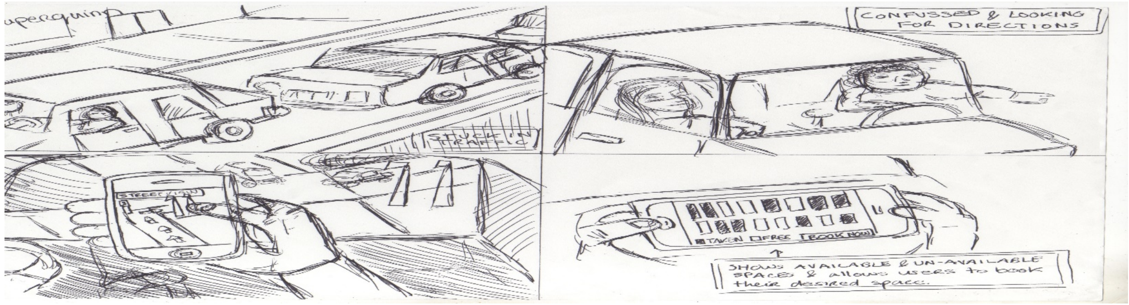
What we learnt from the previous scenario?

- The importance of getting the author’s name right
- Annoyance over the password
- Lack of flexible search
- Usefulness of showing similar entries
- A common use of a library system is to search for book using the author

* You can see already how scenarios force you to think about the design in detail and notice potential problems before they happen.

Scenarios

The scenario example was plain text, but scenarios can be augmented by sketches, simulated screen shots, etc. These sketches and pictures are called ***storyboards***.


uperquinn
STUCK IN TRAFFIC
CONFUSSED & LOOKING FOR DIRECTIONS
STREET VIEW
TAKEN
FREE
BOOK NOW
SHOWS AVAILABLE & UN-AVAILABLE SPACES & allows users to book their desired space.

Exercise !

- Develop a scenario for a university student (primary user) who is using University library website to find out how many books are written by his favouri

Use Cases

- Focus on user-system interaction

- A more formal definition than a scenario or story

- Use cases add value because they help explain how the system should behave and in the process, they also help brainstorm what could go wrong.

Use Cases

- Give an “external view” of the system

- Describe the interaction(s) as
  - text
  - use case diagram

Use Cases text

- A use case text is a written description of how users will perform tasks on your system. Each use case is represented as a sequence of simple steps, beginning with a header containing a user's goal and ending when that goal is fulfilled.

- The users of the system in Use Cases Text are called actors:

  – Actors are external entities (people, other systems or

  other organizations) who interact with the system to

  achieve desired goal

Use Cases text

– In a banking system the most obvious actor is the customer.

– Other actors can be bank employee or cashier depending on the role your trying to show in the use case.

– An example of an external organization can be the central bank. Loan processor is a good example of external system associated as an actor.

Use case text

- **Use Cases text do**

  – Represent the *goal* of an interaction between an actor

  and they system

  – Record a set of paths from start to *goal*

  – Record another set of paths that fail to achieve goal

  – Each step assumed to be successful.

  If there could be a failure, an exception describing

  recovery should be included

Use Case Text Example

- • **Buy a product online**

  1. User browses products

  2. User adds a product to shopping basket

  3. System displays the shopping basket with the new product added

  4. User proceeds to check out

  5. User may register as a new customer, sign in as a returning customer, or have password sent by e-mail in case they have forgotten it

  6. User fills in shipping and payment information

  7. System validates shipping and payment information

Use Case Text Example

- 8. System displays order

  9. User confirms order

  10. System confirms sale

– Exceptions:

6a. User is a returning customer

1. System displays the user's current shipping and payment

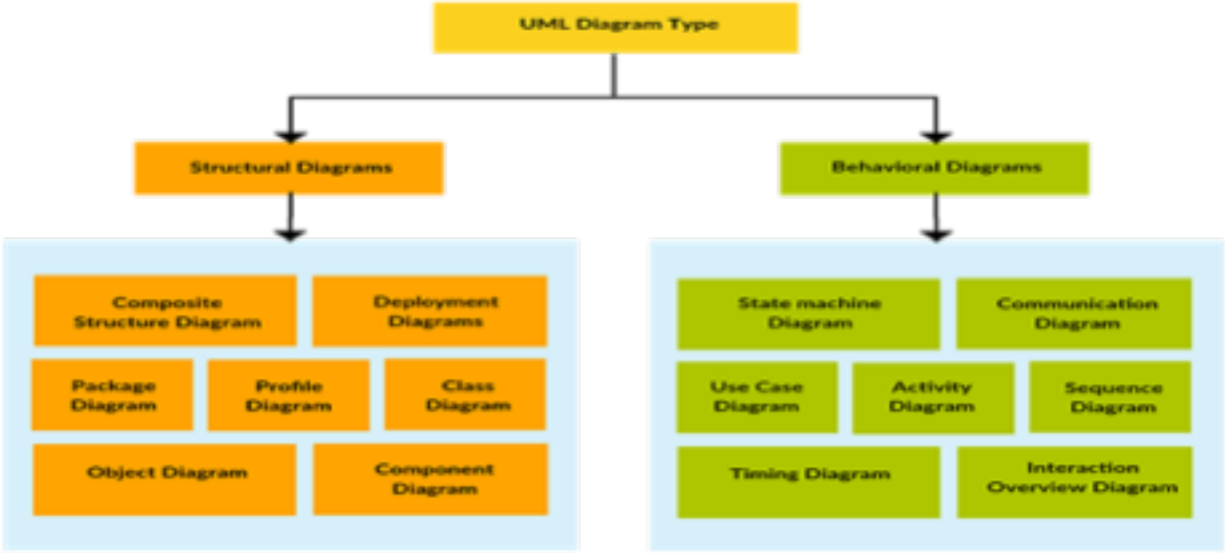
information

2. User may edit current shipping and payment

information

Use Case Diagrams (UCD)

- Use case diagram (UCD) is one of the behavioural Unified Modeling Language (UML) diagrams.



Use Case Diagrams (UCD)

- **Structure diagrams** show the things in a system being modeled. **Behavioral diagrams** shows what should happen in a system.

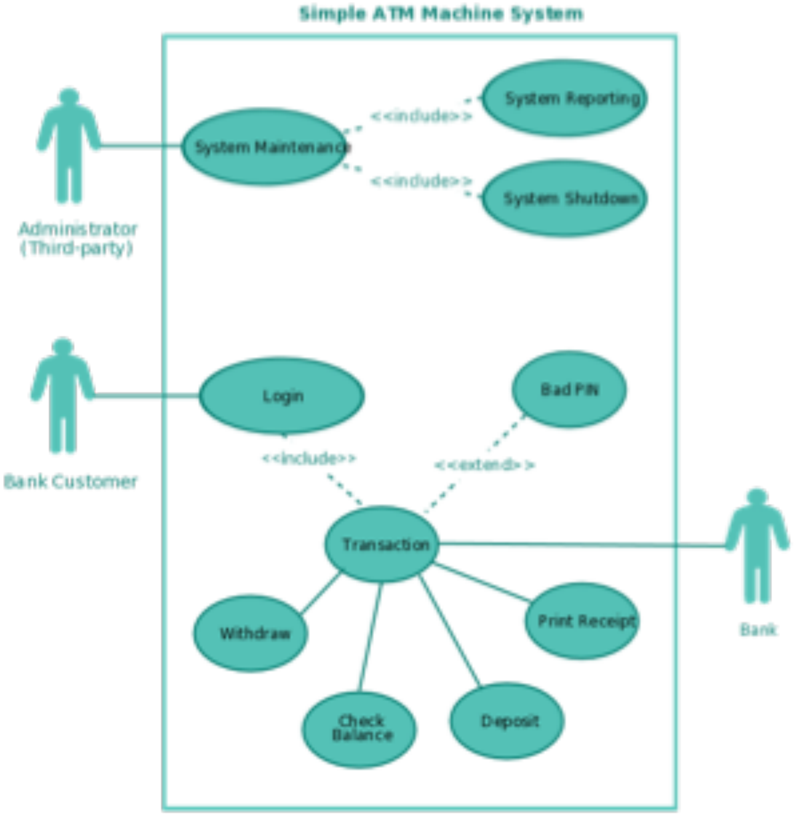
- **UCDs have 4 major elements:**

The **actors** that the system you are describing interacts with, the **system** boundary (the system itself), the **use cases**, or function, that the system knows how to perform, and the lines that represent **relationships** between

relationships

these elements.

Use Case Diagrams (UCD)



Use Case Diagrams (UCD)

- **Actors**
  - **Give meaningful relevant names for actors** – For example if your use case interacts with an outside organization its much better to name it with the function rather than the organization name. (Eg: Airline Company is better than PanAir)
  - **Primary actors should be to the left side of the diagram** – This enables you to quickly highlight the important roles in the system.
  - **Actors don’t interact with other actors**

Use Case Diagrams (UCD)

- **System boundary** (also called system or subject) is presented by a rectangle with system's name, associated keywords and stereotypes in the top left corner. Use cases applicable to the system are located inside the rectangle and actors - outside of the system boundary.

- It is an *optional* element.

Use Case Diagrams (UCD)

- **Use Cases**
  - **Names begin with a verb** – An use case models an action so the name should begin with a verb.
  - **Make the name descriptive** – This is to give more information for others who are looking at the diagram. For example “Print Invoice” is better than “Print”.
  - **Highlight the logical order** – For example if you’re analyzing a bank customer typical use cases include open account, deposit and withdraw. Showing them in the logical order makes more sense.

Use Case Diagrams (UCD)

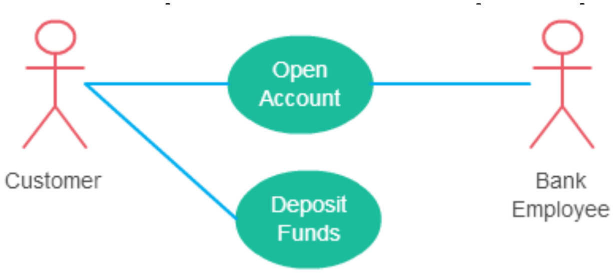
- **Relationships in Use Case Diagrams**
- There are four main types of relationships in a use case diagram. They are

1. Association between an actor and a use case
2. Extend relationship between two use cases
3. Include relationship between two use cases
4. Generalization of a use case

Relationships in Use Case Diagram

1. Association Between Actor and Use Case

- This one is straightforward and present in every use case diagram. Few things to note.
  - An actor must be associated with at least one use case.
  - An actor can be associated with multiple use cases.
  - Multiple actc a single use case.



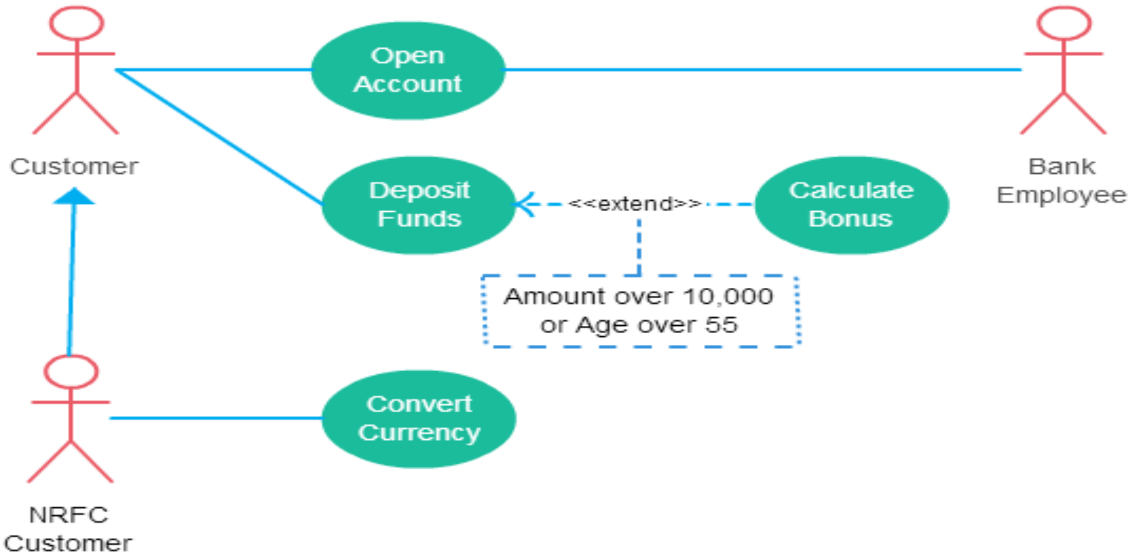
Relationships in Use Case Diagram

2. Extend Relationship Between Two Use Cases

- As the name implies it extends the base use case and adds more functionality to the system

- Here are few things to consider when using the <<**extend**>> relationship.
    - **The extending use case is dependent on the extended (base) use case**. In the below diagram the “Calculate Bonus” use case doesn’t make much sense without the “Deposit Funds” use case.

Relationships in Use Case Diagram



Relationships in Use Case Diagram

- The extending use case is usually optional and can be triggered conditionally. In the diagram you can see that the extending use case is triggered only for deposits over 10,000 or when the age is over 55.

- The extended (base) use case must be meaningful on its own. This means it should be independent and must not rely on the behavior of the extending use case.

Relationships in Use Case Diagram

3. Include Relationship Between Two Use Cases

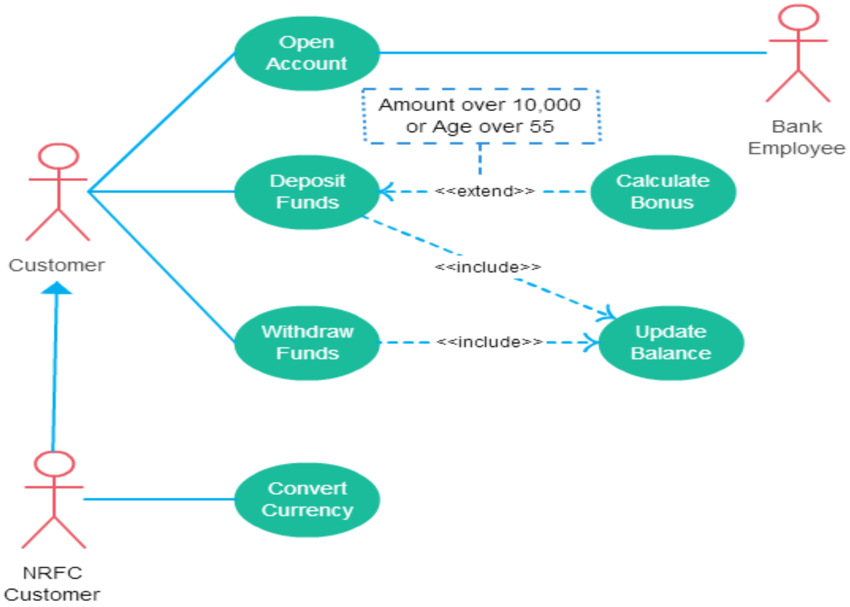
- Include relationship show that the behavior of the included use case is part of the including (base) use case. The main reason for this is to reuse the common actions across multiple use cases. In some situations this is done to simplify complex behaviors. Few things to consider when using the <<include>> relationship.

- The base use case is incomplete without the included use case.

- The included use case is mandatory and not optional.

Relationships in Use Case Diagram

- Lest expand our banking system use case diagram to show include relationships as well.



Relationships in Use Case Diagram

- Arrow points to the base use case when using <<extend>>
- Arrow points to the included use case when using <<include>>
- Both <<extend>> and <<include>> are shown as dashed arrows.
- Actor and use case relationship doesn’t show arrows.

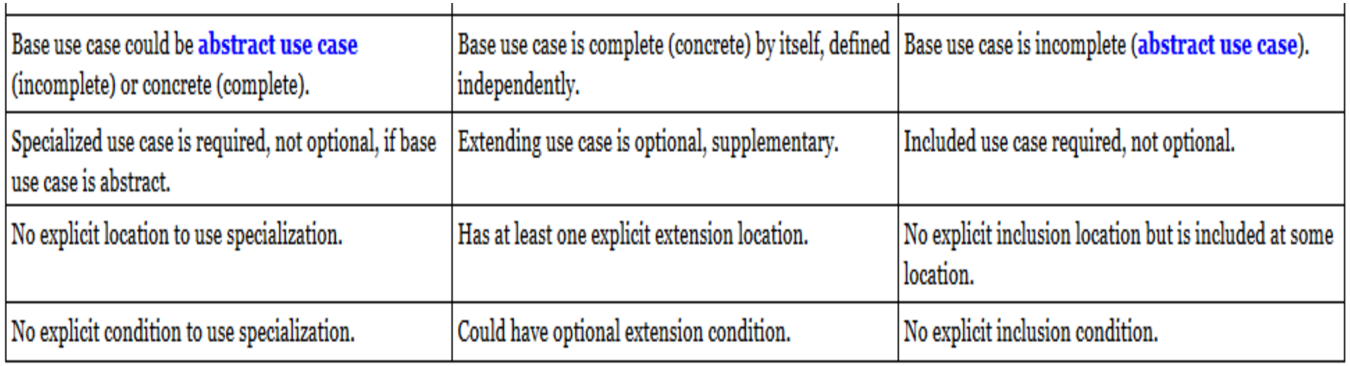
Relationships in Use Case Diagram

4. Generalization of a Use Case

- The behavior of the ancestor is inherited by the descendant. This is used when there are common behavior between two use cases and also specialized behavior specific to each use case.

- For example in the previous banking example there might be an use case called “Pay Bills”. This can be generalized to “Pay by Credit Card”, “Pay by Bank Balance” etc.

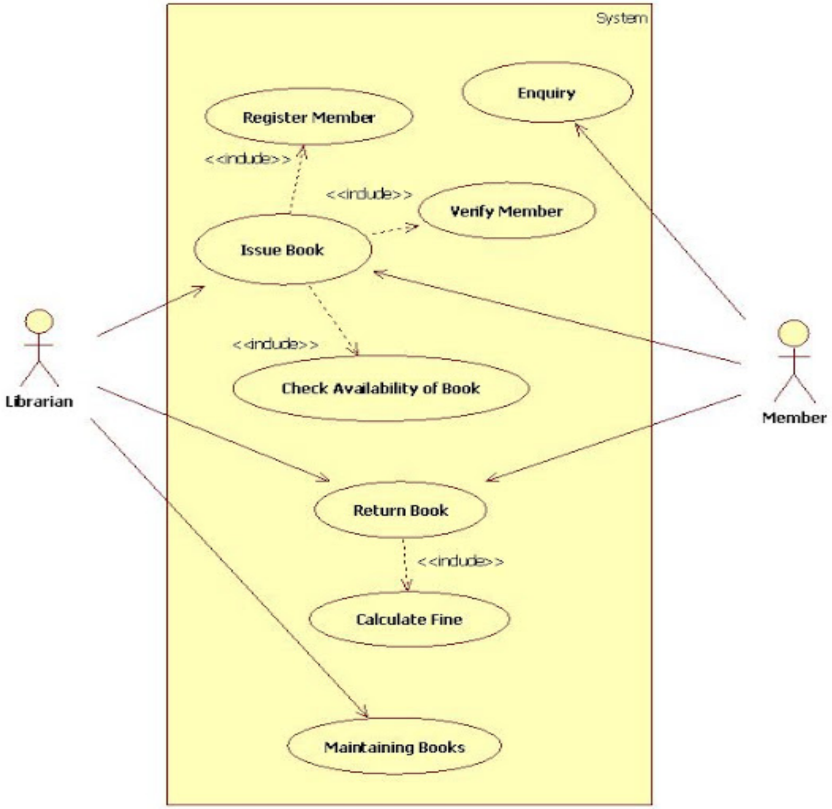
Relationships in Use Case Diagram



Generalization	Extend	Include
Base use case could be **abstract use case** (incomplete) or concrete (complete).	Base use case is complete (concrete) by itself, defined independently.	Base use case is incomplete (**abstract use case**).
Specialized use case is required, not optional, if base use case is abstract.	Extending use case is optional, supplementary.	Included use case required, not optional.
No explicit location to use specialization.	Has at least one explicit extension location.	No explicit inclusion location but is included at some location.
No explicit condition to use specialization.	Could have optional extension condition.	No explicit inclusion condition.

Exercise !

Draw a use case diagram for University library website


System
Enquiry
Register Member
<<include>>
<<include>>
Verify Member
Issue Book
<<include>>
Check Availability of Book
Librarian
Member
Return Book
<<include>>
Calculate Fine
Maintaining Books