

SE202

Introduction to Software Engineering

Lecture 5-3

Database

UML

User Interface design

Pathathai Na Lumpoon

Last class

- Software design
- Architecture design

Today

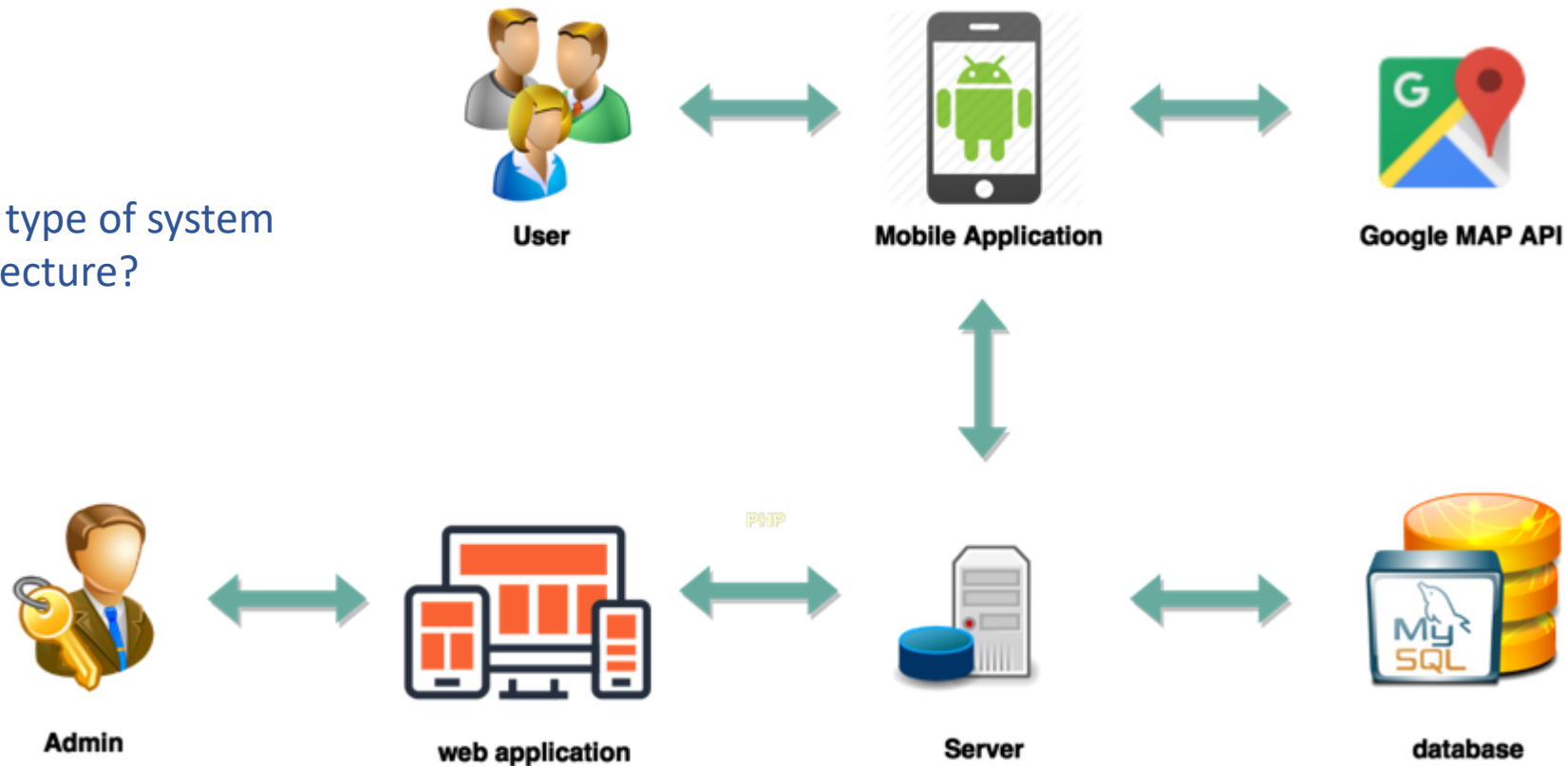
- UML
- Database design
- User Interface design

Different aspects of design

- *Architecture design*:
 - The division into subsystems and components,
 - How these will be connected.
 - How they will interact.
 - Their interfaces.
- *User interface design*
- *Database design*
- *Class design*:
 - The various features of classes.
- *Algorithm design*:
 - The design of computational mechanisms.
- *Protocol design*:
 - The design of communications protocol.

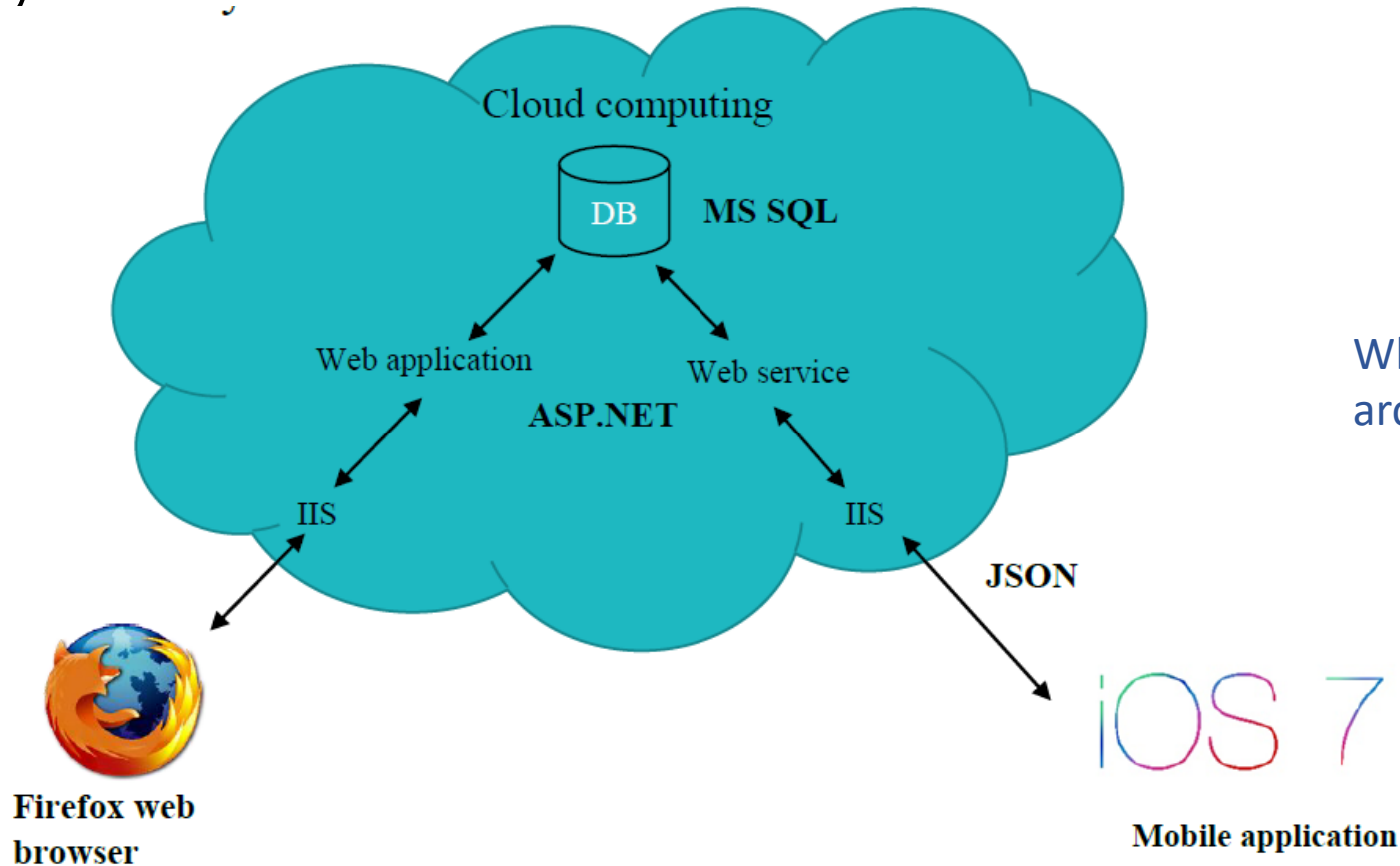
Example: The map interaction management system

What type of system architecture?



The system shall consist of 2 sides, Client, and Server. The Client side separated are mobile and web application, both side are an application that connects and communicate with a server. The server side is where all information which you want to store in the database. The flows of the system are two-ways. First is input flow where user input data in client side, then the input data is translated into computer code and stored in the database. The second way is output flow where the server returns data to client-side by translate data into human language and display on the client side. Other components such as Google Map API are utilities and additional functions that are included in our software.

Example: Smart course Management system in the cloud



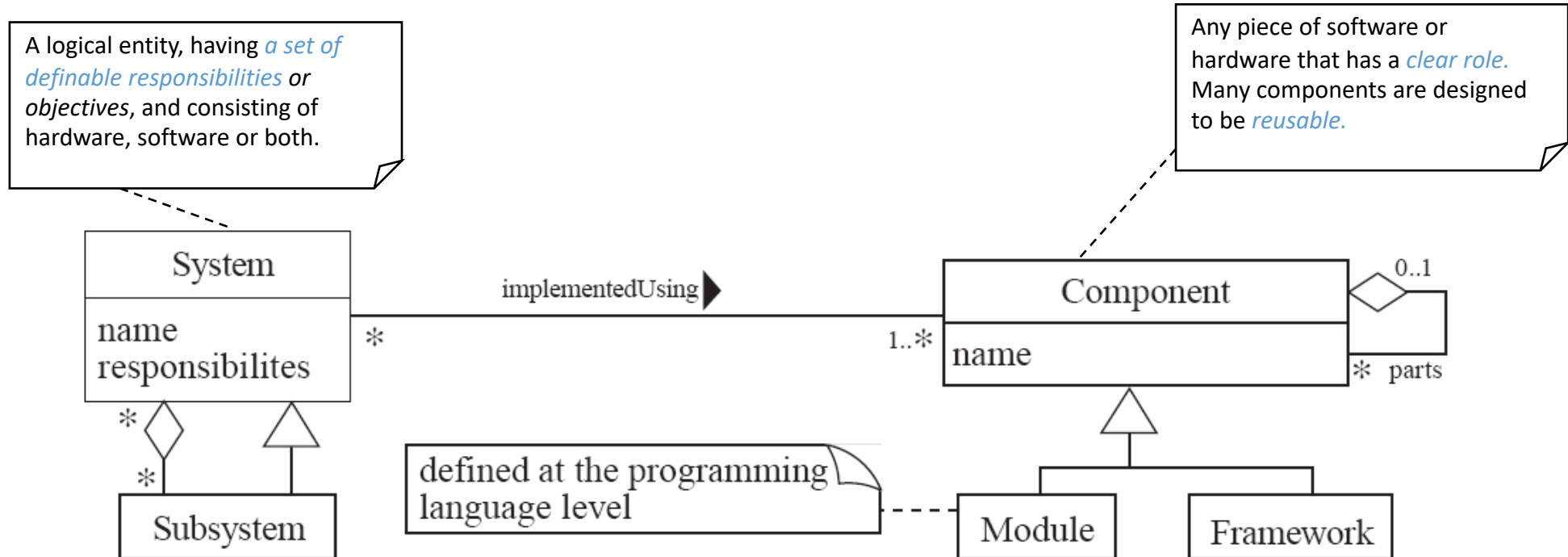
What type of system architecture?

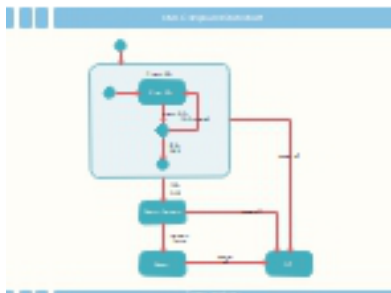
The architecture of Smart course Management system in the cloud is shown. It consists of two parts. The first part is the web application, which the user can access by using a web browser. The second part is the mobile application, which allows the user to access the services provided by the system. The system uses JSON for interchanging data between the web services and the mobiles application.

Object-oriented design

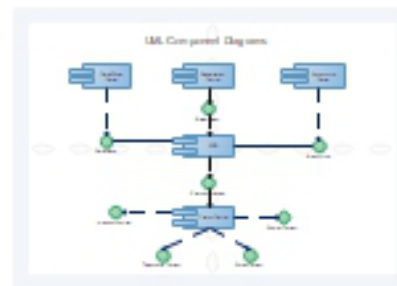
- The process of developing abstract models of a system
- Represents a system using some kind of graphical notation
- Now mostly based on notations in the Unified Modeling Language (UML).
 - The current UML 2.5 standards call for 13 different types of diagrams: class, activity, object, use case, sequence, package, state, component, communication, composite structure, interaction overview, timing, and deployment.
 - <https://www.uml-diagrams.org/>
- Different models present different perspectives

UML diagram of system parts

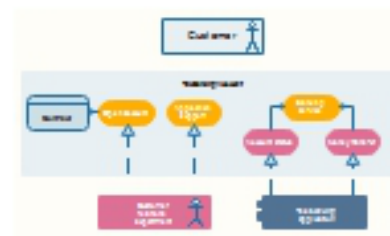




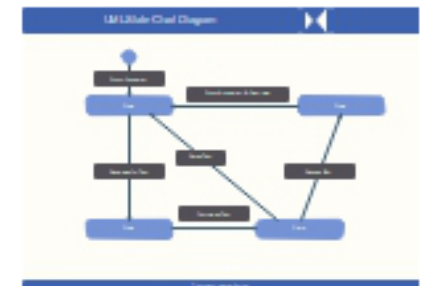
State diagram



Component diagram



Activity diagram



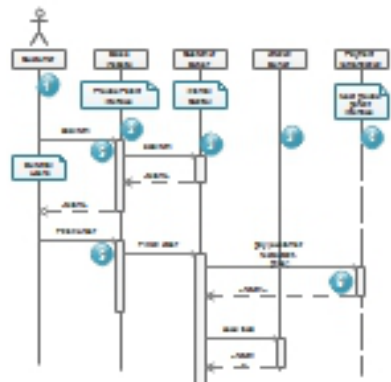
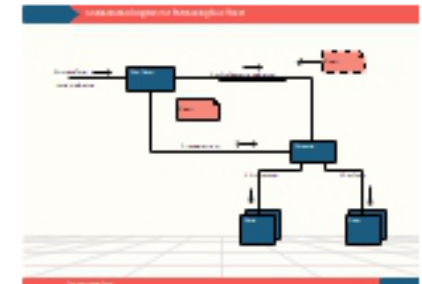
Use case diagram



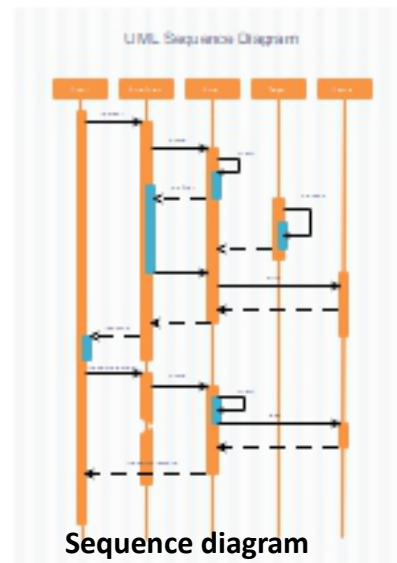
Package diagram



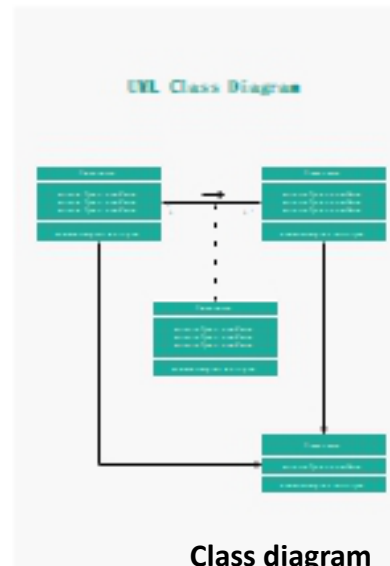
Activity diagram



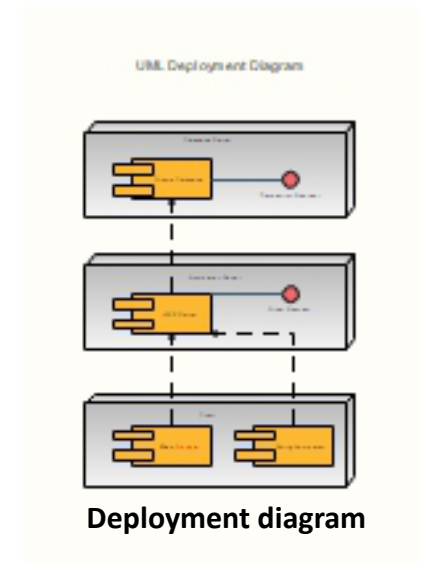
Sequence diagram



Sequence diagram



Class diagram



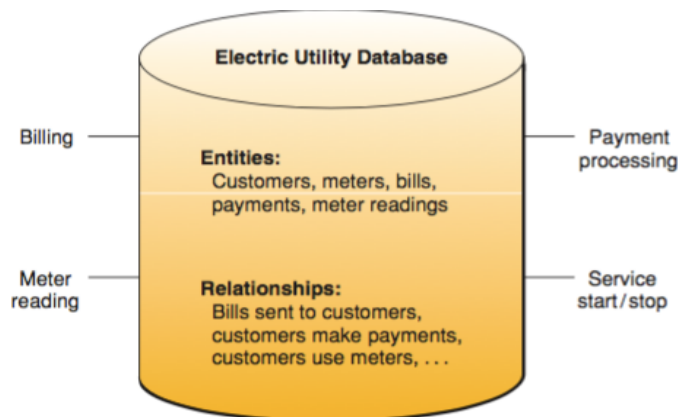
Deployment diagram

Common UML Diagram Types

Activity diagrams	Activities involved in a process or in data processing
Use case diagrams	Interactions between a system and its environment
Sequence diagrams	Interactions between actors and the system and between system components
Class diagrams	Object classes in the system and associations between classes
State diagrams	How the system reacts to internal and external events

Database design

- Database design is an important part of most applications. What is database?
- A database is an organized collection of related information.
- Data in databases
 - Stored as fields, records, files, or objects
 - Independent from applications using it
 - Describe database entities and relationships among entities
 - Database entity is a thing, person, object or any item about which the data should be captured and stored in the form of properties and tables.



SOURCE: Michael V. Mannino, *Database Application Development and Design* (Burr Ridge, IL: McGraw-Hill/Irwin, 2001), p. 6.

Database design

1. Kind of database the program will need.

- Where will the application store data?
 - Text files, XML files, a relational database, or something more exotic such as a temporal database or object store.
 - If you use a relational database, you can sketch out the tables it contains and their relationships during high-level design. Later you can provide more details such as the specific fields in each table and the fields that make up the keys linking the tables.
 - Even a program that doesn't use any database still needs to store data, perhaps inside the program within arrays, lists, or some other data structure.
- Many applications store their data in relational databases such as Access, SQL Server, Oracle, or MySQL.

Database design

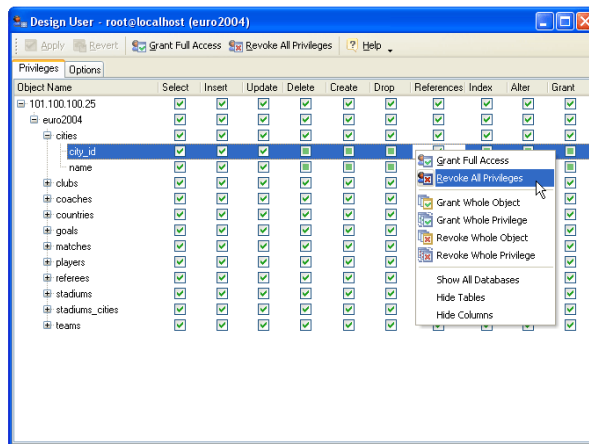
2. Audit trails

- Important for programs that involve money, confidential records or other data that might be tempting to misuse.
- Audit trails keeps track of each user who modifies a specific record.
 - Auditing can be as simple as creating a history table that records a user's name, a link to the record that was modified, and the date when the change occurred.
 - E.g. management can use the audit trails to see which employee gave a customer a 120 percent discount.
- Some database products can even create audit trails for you.
- Many applications don't need auditing.

Database design

3. User Access

- Many applications also need to provide different levels of access to different kinds of data.
- E.g. Only managers need to see the other employees' salary information.
- One way to handle user access is to build a table listing the users and the privileges they should be given.



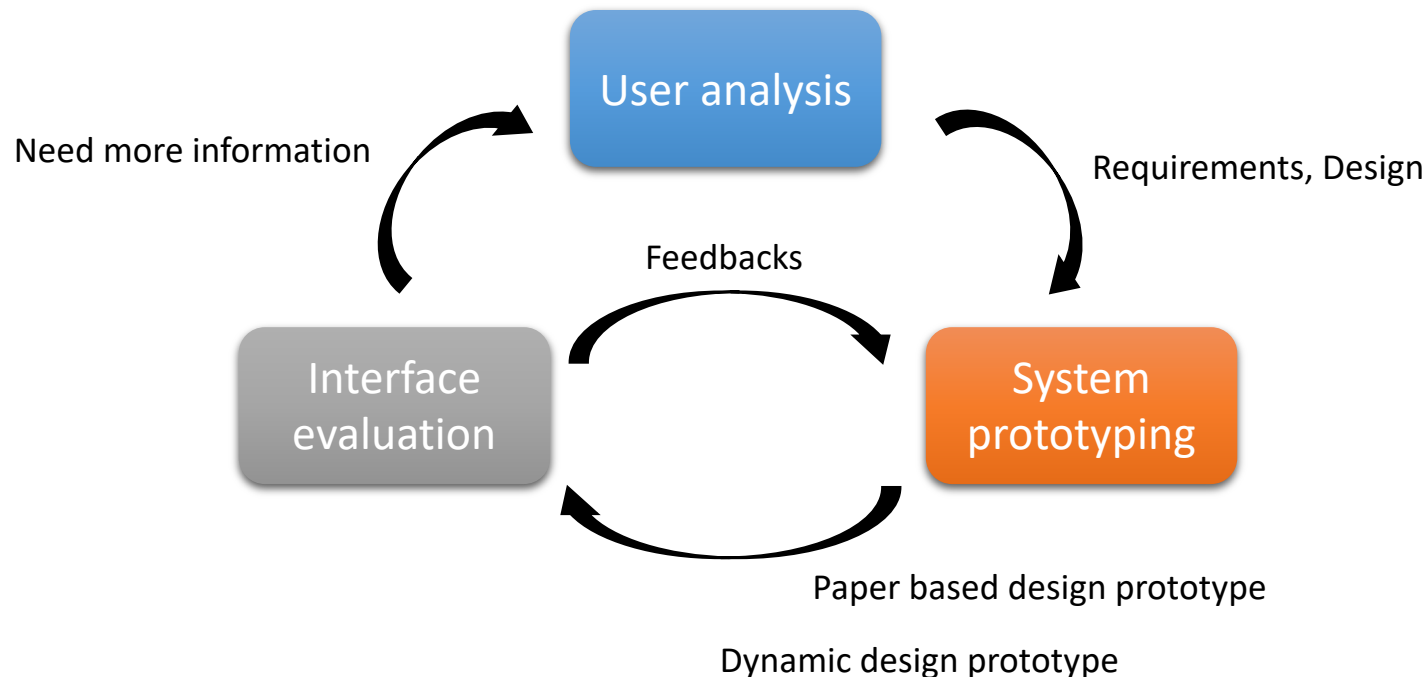
Exercise: Identify database entities of the room reservation system

Requirements of the system

- The system is web application.
- Teacher uses the system to view room available, request for booking.
- Student uses the system to only view room available.
- IT staff uses the system to view room available, manage booking, approve booking and manage room.

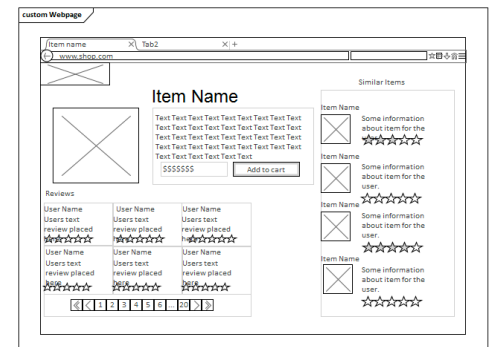
User Interface (UI) design

- UI design is an iterative process involving close channels between users and designers
- Three core activities in UI design



Wireframes

- A wireframe is a layout of a web page that demonstrates what interface elements will exist on key pages.
- It is a critical part of the interaction design process.
- The aim of a wireframe is to provide a visual understanding of a page early in a project to get stakeholder and project team approval before the creative phase gets under way.
- Tools
 - Paper, pen, pencil, colors or some sticky notes
 - Software
 - PPT, Drawing program
 - Balsamiq, Adobe XD



Example: ER Sepsis APP

User Interface Design Principles

Principle	Description
User familiarity	Use terms and concepts familiar to the user
Consistency	Comparable operations should be activated in the same way . E.g. Commands and menus should have the same format.
Minimal surprise	If a command operates in a known way, the user should be able to predict the operation of comparable commands.
Feedback	Provide the user with visual and auditory feedback, maintaining two-way communication .
Memory load	Reduce the amount of information that must be remembered between actions. Minimize the memory load .
Efficiency	Seek efficiency in dialogue, motion and thought. Minimize keystrokes and mouse movements .
Recoverability	Allow users to recover from their errors . Include undo facilities, confirmation of destructive actions.
User guidance	Incorporate some form of context-sensitive user guidance and assistance.

Exercise: Dog wash service system

- Suppose you're building an application to manage the dog washing vehicles run by The Pampered Poodle Emergency Dog Washing Service. When a customer calls in to tell you Fifi (a dog) ran into in the heavy mud, you dispatch an emergency dog-washer to the scene.



- In this case, your drivers might access the system to view their assignment over cell phones.
- A desktop computer back at the office would hold the database and provide a user interface to let you do everything else the business needs such as logging customer calls, dispatching drivers, printing invoices, tracking payments, and ordering doggy shampoo.
- Your tasks
 1. Create the architecture of the system
 - you would specify the kind of phones the drivers will use (such as Windows, iOS, or Android),
 - the model of the computer used to hold the database and business parts of the application, and the type of network connectivity the application will use.
 2. Identify database of the system
 3. Design and draw wireframes of the system