

VIS - Exercise 5



Artifact 3

Non-functional requirements

Repetition Exercise 4

Artifact 2 – Use-Case diagrams

- Use-Case description (Scenarios)
 - At least five steps in the normal flow
 - At least five steps in the alternative flow

Repetition Exercise 4

Artifact 3 – Technical specification (Non-functional requirements)

- ➡ First model of the domain
(Conceptual model)
- ➡ Technological decisions
- ➡ Selected technologies and processes
- ➡ **Justification of choice!**

Artifact 3 content

1. **Chart of Conceptual domain model** (based on Vision and implemented Use-Cases)
2. **Table containing:**
 - Estimation of entity sizes and quantities.
 - Estimation of the number of users working simultaneously with the system.
 - Assign classes to particular storage.
3. **Text with technological decisions**
 - Types of user interactions with the system and estimation of their complexity.
 - First idea of the system layout.
 - Choice of the used platform.
 - Estimation of patterns and architecture.



First model of domain

Generate Domain Vocabulary With a Class Diagram

By working from your **Use-Case descriptions and Vision**, you can design your classes via a **class diagram**. A class diagram will contain a set of classes, attributes and relationships.

1. Identify **Nouns** - a use case description describes the sequence of interactions between the user and the system. It can be at any level of detail that you need. It's pretty simple. The user does something, the system responds. Over and over, until the goal is reached.
2. Represent **Nouns** as candidates for classes or attributes
3. The **Verb** – candidate for procedure/class method

Example of use-case description

Use-Case - Searching for a book

- The **user** requests to **search** for a **book**.
- The **system displays** the **search page**.
- The user enters the **author's name**.
- The system **verifies** that the author's name exists.
- The user enters the **title** of the **book**.
- The user requests, the **search** executes.
- The system searches.
- The system **displays** the **results**:
 - A collection of books that match.
 - An **image** of the cover of each book.
 - A short author **bio**.
 - An **image** of the author.

Step one – Identify nouns

- Once you have that interaction written out, you can play a game of **spot the noun**. Let's define two kinds: sophisticated/complicated or simple. If the noun is sophisticated (it can't be described in a word or two), it's likely to be a **class** in an object-oriented language. If it's a simple noun, then it will probably be an **attribute** of a class.
- For example, a book seems like it would be a class with several items associated with it (author, title). It can't be represented as an integer or string. Conversely, *the title* appears simple and probably just a string. It will be an attribute, but of what? Well, we say it's the "title of a book." So, it will be an attribute of the class book.

What about user and system?

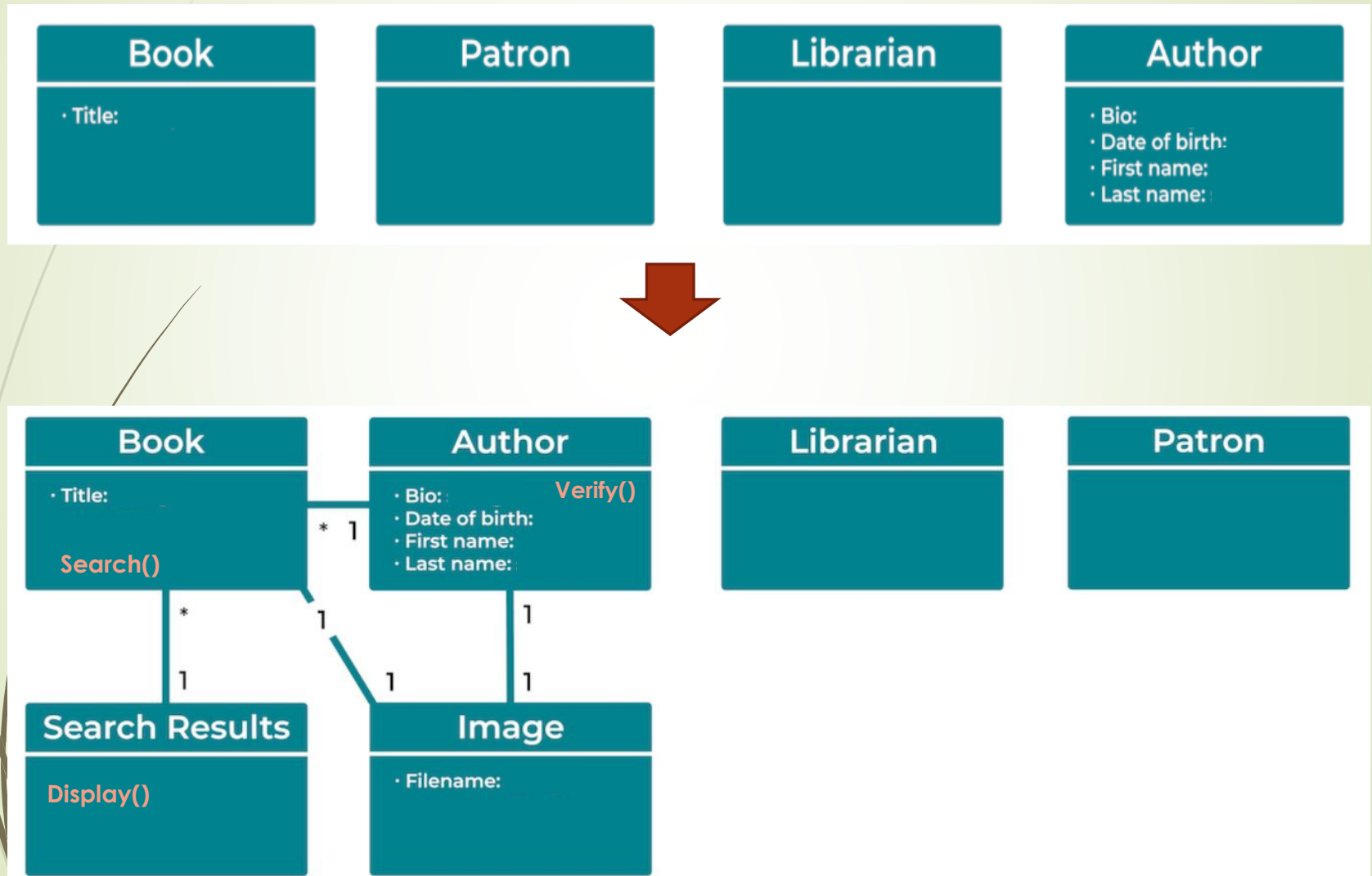
Are those classes, too?

- **System** - since that's what you're building, it's not going to be a single class. It's going to be the collection of classes you come up with, working together.
- **User** - depending on your system, you might have different types of users (librarians, patrons) who have access to different parts of the system. You would want to have classes represent their different capabilities/access rights; however, you traditionally wouldn't have a class named user.

Step two - classes or attributes candidates

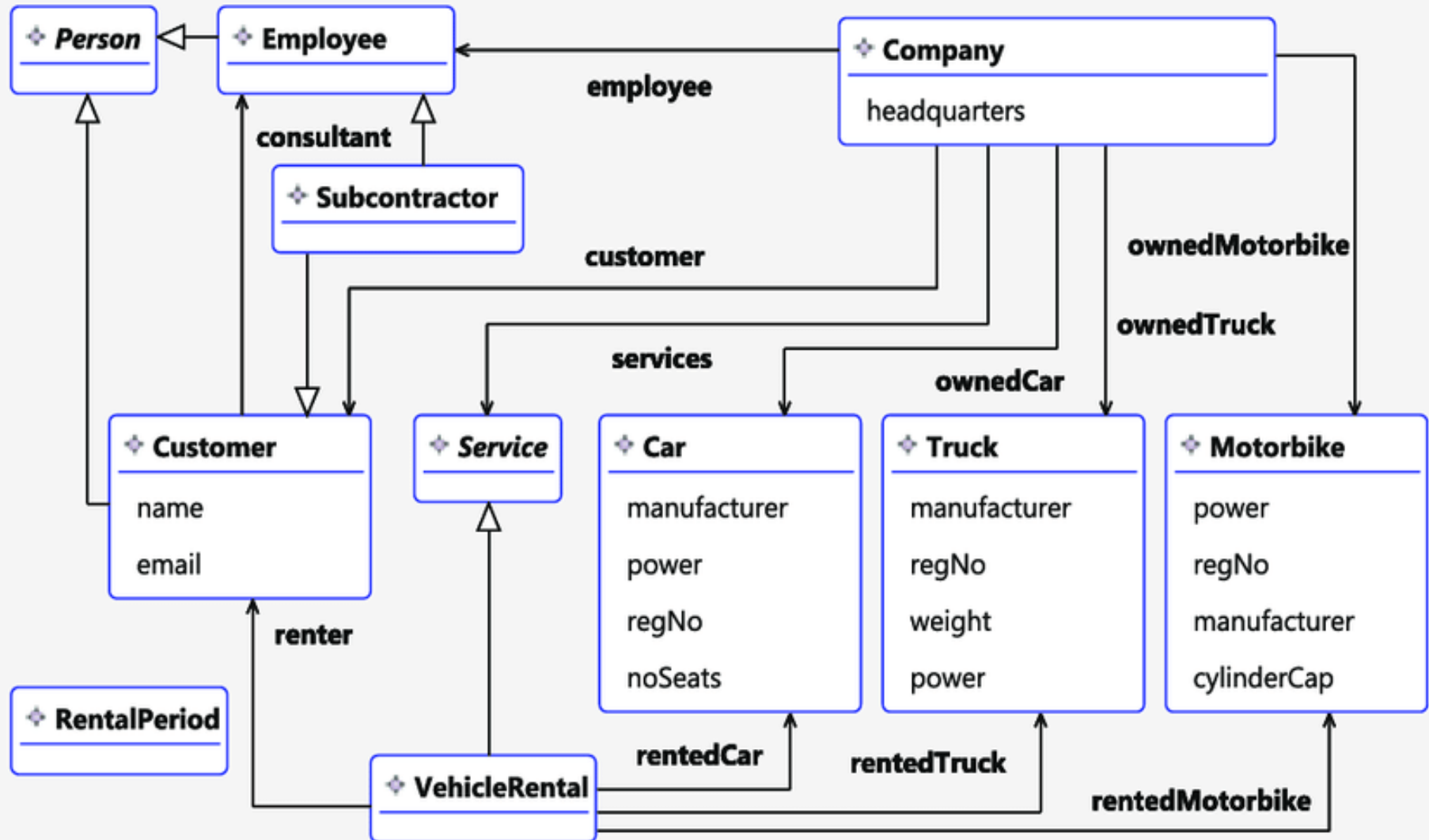
Step three - verbs candidates for methods

Step four - relationships (agregation, inheritance..)




First model of domain - platform independent

VehicleRentalCompany



Technological decisions - estimations



Class	Storage	Estimated Entity size in kB	Estimated number of Entities after one year	Estimated Entities size after one year in kB
Book	MS-SQL	1	1000	1000
Author	MS-SQL	1	300	300
Librarian	XML	0.3	10	3
Patron	XML	0.2	20	4
Search results	MS-SQL	1	100	100
Image	MS-SQL	500	3000	1 500 000

Estimated database size after one year is 1.5 GB

Estimating growing factor is 1.2 per year.

Describe:

- Why is selected MS-SQL?
- Why is selected XML for particular tables?

Types of user interactions with the system and estimation of its complexity (example)

Users access the system primarily through a web interface that offers each type of user different options for working with the system. Administrative staff (librarians) will have full control over the system (except for actions specific to system administrators only).

The web interface will offer the same template for all forms. This should make it intuitive and easy for users to navigate the system once they have been trained. The desktop application will try to be as close as possible to the appearance of the web application in order to make it as easy as possible for the user to navigate and work with the system. All actors will also be given the possibility to create requests corresponding to their rights. Administrative staff work primarily with the desktop application, but can also use the web application.

The library expects a 20% annual increase in the number of books registered and at least a 10% increase in users.

For the design of the information system, a three-layer architecture with two different UIs (web and desktop) and two different data stores are used. For the repository xxxxxx was chosen because of xxxxxx. The IS will be deployed on xxxxxx because of xxxxxx.

Choice of platform



- .NET framework min version
- JRE min version
- System requirements
- OS on stations
- IIS, Apache, Nginx what version
- Data storage (Server, local/remote) + justification
- Describe GoF and architectural design patterns for develop IS

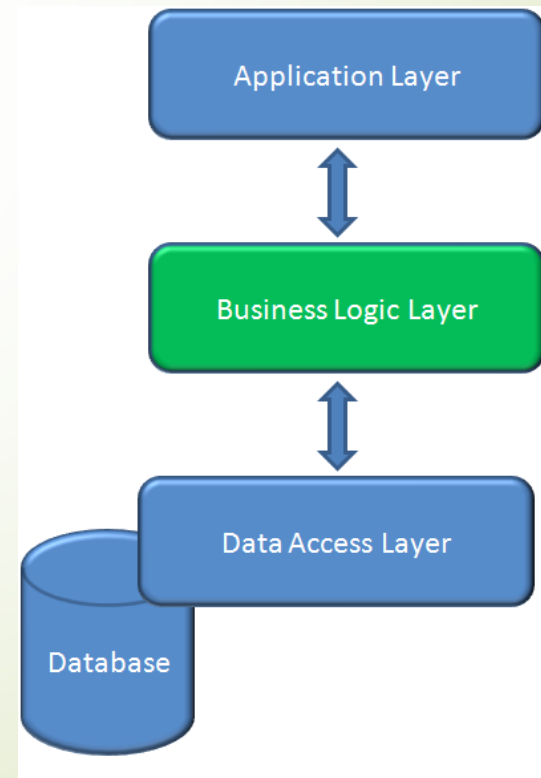
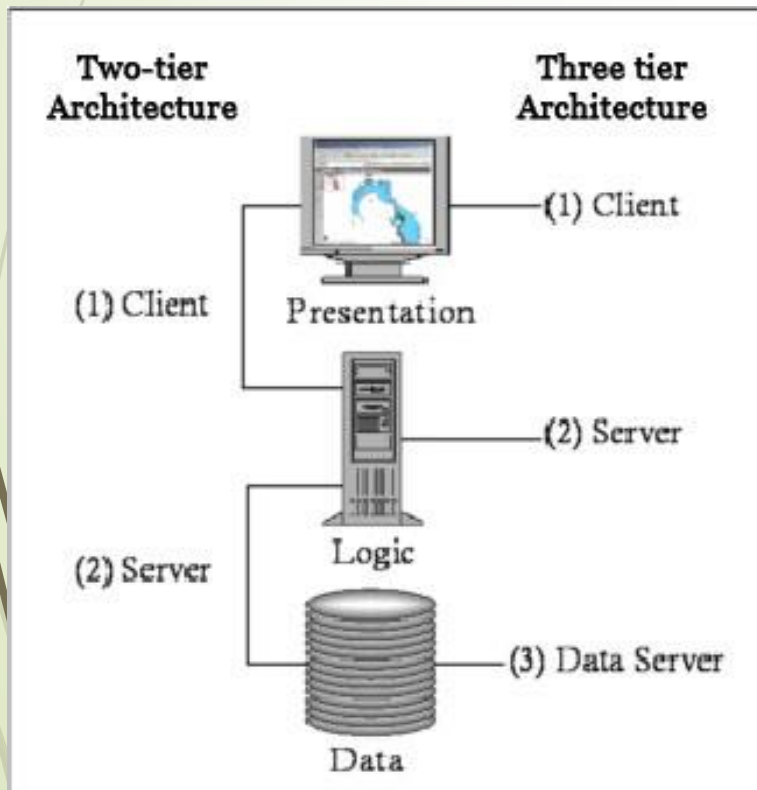


IS requirements by location

- Web application
- Technology, API usage, framework, browser.
- Fat client (desktop)
 - What technology, for what OS, system framework.
- Mobile client
- Client type, target OS, standards (HTML 5).

System layout

- N-layer - IS architecture (logical division)
- N-tier - IS layout (physical division)
- Hw requirements (server, client, connectivity)





Discussion

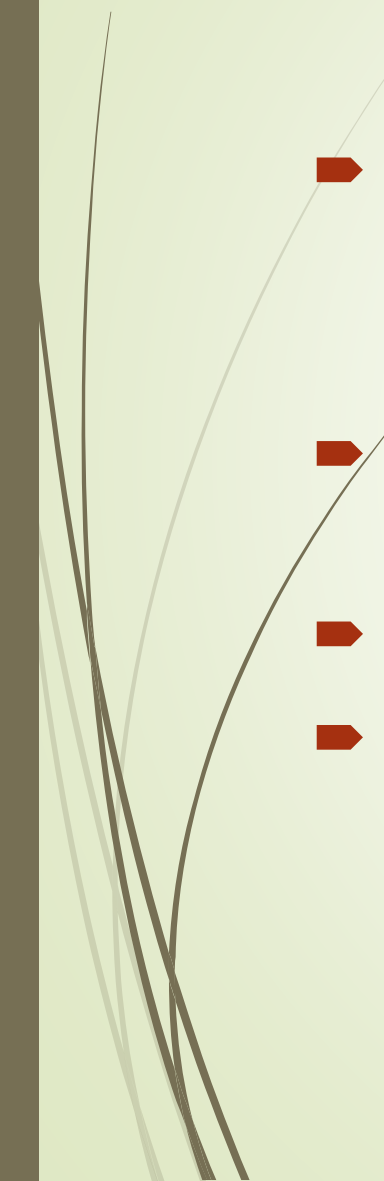
Artifact 2

Artifact 3





Programming tasks

- Continue lectures task and semestral work implementation
 - Create two classes with association links from UI to DB (.NET or JAVA)
 - Use different Business logic patterns
 - Use different Data patterns
- 



https://github.com/MartinRaSt/VIS_course