Spaninks,Luka L.

Modules s3

Software

Inhoud

[Distributed Software System 2](#_Toc55127604)

[Expected outcomes: 2](#_Toc55127605)

[Questions: 2](#_Toc55127606)

[Front-End Development 3](#_Toc55127607)

[Expected outcomes: 3](#_Toc55127608)

[Questions: 3](#_Toc55127609)

[Back-End Development 4](#_Toc55127610)

[Expected outcomes: 4](#_Toc55127611)

[Questions: 4](#_Toc55127612)

[Distributed Communication 5](#_Toc55127613)

[Expected outcomes: 5](#_Toc55127614)

[Questions: 5](#_Toc55127615)

[Data Persistency 6](#_Toc55127616)

[Expected outcomes: 6](#_Toc55127617)

[Questions: 6](#_Toc55127618)

[Quality Assurance 7](#_Toc55127619)

[Expected outcomes: 7](#_Toc55127620)

[Questions: 7](#_Toc55127621)

[Software Release Management 9](#_Toc55127622)

[Expected outcomes: 9](#_Toc55127623)

[Questions: 9](#_Toc55127624)

# Distributed Software System

## Expected outcomes:

* You can present an architecture that describes your system as a distributed software system and clearly makes use of important distributed architectural principles.
* You divided your application into several components that each could run (in theory) on a separate server and have a clear single task in the system.
* Your architecture describes the system on multiple levels: system overview, component overview, infrastructural overview, a dynamic and static overview of components
* You designed several APIs that offer sufficient possibilities for your components to communicate with each other (exchange data). You are using messaging & event techniques to achieve communication.

## Questions:

* What is a distributed software system?

*A distributed system is a software system where the functionality of your system (your requirements) is provided by several small applications running on multiple separate servers. Each individual application is responsible for a specific piece of functionality that is offered by your system. It is like you have*distributed*your work over multiple servers. These small pieces of software on each server are often called*components*or*web-services*.*

* What is the difference between a monolith and a distributed software system?

*In a monolith application the application runs on one server/system. In a distributed system several services are spread over different servers.*

* Which advantages does a distributed software system offer?

*It improves maintenance, it improves re-use of functionality and makes your system more resistant to failure.*

* Explain why your architecture is distributed

*I use both a WebSocket server and a REST api. These should both be deployed on a different server and could even be split up into smaller components.*

* What are the important architectural principles and techniques when developing a distributed software system?

<http://www0.cs.ucl.ac.uk/staff/ucacwxe/lectures/ds98-99/dsee3.pdf>

<https://dev.to/smartym/top-5-principles-of-software-distributed-systems-that-you-need-to-know-5d9>

<https://www.martinfowler.com/articles/microservices.html>

<https://www.tutorialspoint.com/software_architecture_design/distributed_architecture.htm>

# Front-End Development

## Expected outcomes:

* You selected a JavaScript framework based on research.  
  You can explain why the framework you choose is a good, if not the best, choice for your project. A report might be needed because of your research, and as a means of communicating your selection.
* You build a front-end for your project that has as a primary language the selected framework.
* You demonstrate applied UX practices in your project in the analysis, design, implementation, and test phases of your project.

## Questions:

* What is the difference between JavaScript and a JavaScript framework?

*Javascript is a programming language which works in most modern webbrowsers. A javascript framework is a boilerplate on which a user can build an application using pre-written code. Angular is an example of a javascript framework.*

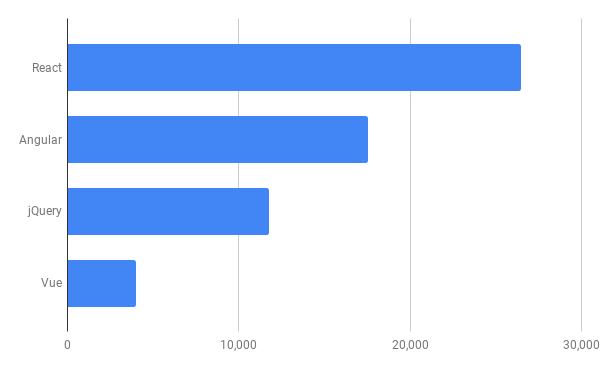
* What are the advantages and disadvantages of using a JavaScript framework?

***Pros****:*

* ***Time****- Makes development faster by giving the product engineers ready-made set of utilities those are common across typical product development*
* ***Development maturity****- Since the patterns used are usually from a community that's led by proven talents and experts in the industry, they effectively contribute towards the overall quality, especially performance and cross-platform compatibility, of the product*
* ***Standards****- All popular frameworks follow the best of industry standards and promote healthy engineering practices. The product team adopting it is likely to follow the same*
* ***Reduced learning curve for the team****- Any product would eventually need a framework - either the team will have to develop one in-house or they adopt a third-party solution, but it must exist for a product to be maintainable. If the team utilizes an already established framework, it would be greatly possible that the team members have some exposure to the framework way of programming and the same can be expected from the future team members, too*
* ***Less in-house maintenance****- On an average, roughly 70-80% of the cost spent during the product life goes in maintenance and support. If majority part of your engineering efforts are from the framework, which it will be, the community behind it would be taking care of its enhancements and bug fixes*

***Cons****:*

* ***Greatly reduces innovation challenges in engineering****- For people who love these challenges will soon run out of juice in their work*
* ***Corner cases\*****- Frameworks need to work for masses. In order to achieve that goal, it has to avoid addressing situations that occur rarely. It becomes a big pain if solving such a problem goes against the framework fundamentals. Almost every product has something unique about it. And eventually there will at least be one such case that doesn't fit into the approach suggested by the framework*
* ***Performance\*****- Although frameworks help in gaining overall performance, there will be areas where the core technology would perform better. All the processing that framework utilities do to make it work across all possible ports aren't always required by a product*
* ***Upgrades/Compatibility****- When there are components of the product that are developed with a particular version of the framework, upgrading the base framework to a future version becomes a major issue. This is one problem which would exist anyway but frameworks need to make intentional efforts to make upgrades easier*
* What are some recent popular JavaScript frameworks?



* What were the specific reasons for choosing your JavaScript framework?

*React: it is the most popular, so there is a lot of documentation on it.*

* Can you name and demonstrate some best practices within your JavaScript framework of choice?
  + *Keep components small and function specific.*
  + *Reusability is important, so keep creation of new components to the minimum required.*
  + *Consolidate duplicate code – DRY your code.*
  + *Put CSS in JavaScript.*
  + *Comment only where necessary.*
  + *Name the component after the function.*
  + *Use capitals for component names.*
* What does UX mean?

*User experience is a person's emotions and attitudes about using a particular product, system or service. It includes the practical, experiential, affective, meaningful, and valuable aspects of human–computer interaction and product ownership.*

* What are common practices within UX to match the expectation of your user target group?
  + User research
  + Walk the user flow
  + Consider user expectation
* What are some examples of techniques to analyze your UX expectations?
  + Rely on a list of conventions and standards (heuristics)
  + Walk the user flow
* What are some examples of techniques to test your UX expectations?
  + UX-testing
  + [Online Surveys](https://careerfoundry.com/en/blog/ux-design/how-to-conduct-user-experience-research-like-a-professional/#4-online-surveys)
  + [Usability Testing](https://careerfoundry.com/en/blog/ux-design/how-to-conduct-user-experience-research-like-a-professional/#2-usability-testing)
  + Google analytics

# Back-End Development

## Expected outcomes:

*For your project, you need to select a back-end framework applicable for your project. This will be your primary 'language' in which you will create the back end of your application.*

*Further requirements for the back-end are to use persistence (using an*[*ORM*](https://fhict.instructure.com/courses/10457/modules/items/542273)*), and distributed client-server communication. Also, we need you to take*[*concurrency*](https://fhict.instructure.com/courses/10457/modules/items/542271)*into consideration.*

#### **By the end of the semester**

* You have implemented a distributed software system by using an OO application framework, which has been agreed upon with your teacher. For this, you make a choice which object-oriented language you want to use for your project. In principle we want you to choose between Java and .NET.
* Your back end also integrates the expected outcomes as mention in modules:
  + [Distributed Communication](https://fhict.instructure.com/courses/10457/modules/items/542269)
  + [Data Persistency](https://fhict.instructure.com/courses/10457/modules/items/542272)

## Questions:

* What is the definition of software back-end?

*The "****backend****" refers to any part of a website or software program that users do not see. This means data handling, processing requests and more.*

* What is an application framework?

*A software framework is an abstraction in which software providing generic functionality can be selectively changed by additional user-written code, thus providing application-specific software*

* What are some commonly used application frameworks?

*Spring boot, Django, ASP.NET Core and express.js*

* Why is the framework I chose suitable for my application?
  + *Reduces development time and increases the overall productivity of the development team.*
  + *Helps you autoconfigure all components for a production-grade Spring application.*
  + *Makes it easier for developers to create and test Java-based applications by providing a default setup for unit and integration tests.*
  + *Avoids writing lots of boilerplate code, annotations, and XML configuration.*
  + *Comes with embedded HTTP servers like*[*Tomcat or Jetty*](https://stackify.com/tomcat-vs-jetty-vs-glassfish-vs-wildfly/)*to test web applications.*
  + *Adds many plugins that developers can use to work with embedded and in-memory databases easily. Spring allows you to easily connect with database and queue services like Oracle, PostgreSQL, MySQL, MongoDB, Redis, Solr, ElasticSearch, Rabbit MQ, ActiveMQ, and*[*many more*](https://spring.io/guides)*.*
  + *Allows admin support—meaning you can manage via remote access to the application.*

# Distributed Communication

## Expected outcomes:

* How your individual components in your distributed system communicate with each other.
* Demonstrate both direct messaging to other components and handling events triggered by components in your distributed system.
* Have an overview of used API's in your system

## Questions:

* What is a technique that can be used to send messages to other components in your distributed system?

Using an api endpoint for example.

* What is a technique that can be used to implement an Observer or publish-subscribe pattern in your distributed system?

A form of websocket implementation

* What is REST?

*A RESTful web application exposes information about itself in the form of information about its resources. It also enables the client to take actions on those resources, such as create new resources (i.e. create a new user) or change existing resources (i.e. edit a post).*

* What is a Websocket?

WebSocket is a computer communications protocol, providing full-duplex communication channels over a single TCP connection.

* What does API mean?

*API stands for****application programming interface****. This is a concept in software technology that essentially refers to how multiple applications can interact with and obtain data from one another.*

* Can you name some architectural styles that can be considered a distributed software architecture?
  + *Microservices Architecture*
  + *Client-Server Architecture*
  + *Broker Architectural Style*
* What is Swagger?

*Swagger is an Interface Description Language for describing RESTful APIs expressed using JSON. Swagger is used together with a set of open-source software tools to design, build, document, and use RESTful web services. Swagger includes automated documentation, code generation, and test-case generation.*

[*https://www.youtube.com/watch?v=gduKpLW\_vdY&ab\_channel=JavaBrains*](https://www.youtube.com/watch?v=gduKpLW_vdY&ab_channel=JavaBrains)

# Data Persistency

## Expected outcomes:

* You can select the best fitting database for your application.
* You design and build the database layer in your application using an ORM framework.
* You have created several ORM queries that combine information from multiple tables.
* Within your architecture, you have a description of your data model.
* You used the data model in your architecture to describe your domain objects in your ORM code.  
  *This means you did****not****generate ORM code from an existing database.*

## Questions:

* What is data persistence?

*Persistence is "the continuance of an effect after its cause is removed". In the context of storing data in a computer system, this means that the data survives after the process with which it was created has ended*

* What are some commonly used types of databases?

*MySQL, MsSQL, MongoDB, MariaDB and Redis*

* What is an ORM?

*Object-relational mapping in computer science is a programming technique for converting data between incompatible type systems using object-oriented programming languages.*

* Which ORMs are available?

***.NET***

* + - *Base One Foundation Component Library, free or commercial*
    - *Dapper, open source*
    - *Entity Framework, included in .NET Framework 3.5 SP1 and above*
    - *iBATIS, free open source, maintained by ASF but now inactive.*
    - *LINQ to SQL, included in .NET Framework 3.5*
    - *NHibernate, open source*
    - *nHydrate, open source*
    - *Quick Objects, free or commercial*

***JAVA***

* + - *ActiveJDBC, Java implementation of Active record pattern, inspired by Ruby on Rails*
    - *Apache Cayenne, open-source for Java*
    - *Apache OpenJPA, open-source for Java*
    - *DataNucleus, open-source JDO and JPA implementation (formerly known as JPOX)*
    - *Ebean, open-source ORM framework*
    - *EclipseLink, Eclipse persistence platform*
    - *Enterprise JavaBeans (EJB)*
    - *Enterprise Objects Framework, Mac OS X/Java, part of Apple WebObjects*
    - *Hibernate, open-source ORM framework, widely used*
    - *Java Data Objects (JDO)*
    - *JOOQ Object Oriented Querying (jOOQ)*
    - *Kodo, commercial implementation of both Java Data Objects and Java Persistence API*
    - *TopLink by Oracle*
* Which ORM did you use in your software application?

Hibernate

* What is an entity Manager?

*The EntityManager API is used to create and remove persistent entity instances, to find entities by their primary key, and to query over entities. The set of entities that can be managed by a given EntityManager instance is defined by a persistence unit*

* What is the difference between eager and lazy loading and when do you use which one?

***When to use eager loading***

1. *In "one side" of one-to-many relations that you sure are used everywhere with main entity. like User property of an Article. Category property of a Product.*
2. *Generally, when relations are not too much, eager loading will be good practice to reduce further queries on server.*

***When to use lazy loading***

1. *Almost on every "collection side" of one-to-many relations. like Articles of User or Products of a Category*
2. *You exactly know that you will not need a property instantly.*

* How did you define 1-1; 1-N; N-M relations in your ORM code?

# Quality Assurance

## Expected outcomes:

* You have a written test plan for your software application that describes which type of tests are included, if there is automation present, *why*these tests are necessary, and how much of your project is covered. This test plan is a prove that you have planned the necessary steps to guarantee a certain level of quality of your software project.
* The plan clearly distinguishes between unit, integration, and acceptance tests. These tests should complement each other.
* You have implemented tests mentioned in your test plan with the proper tools and frameworks.
* You use static code analysis tools to score your software quality.
* All of the used tools and frameworks for testing and analysis are integrated into your continuous integration environment.
* You addressed security and performance testing in your test plan.
* *Bachelor only*: By following the approach of the DOT research framework, you perform research specifically on the topic of security.

## Questions:

* What do you consider to be software of high quality?
  + Close to design
  + Secure
  + Maintainability
  + Good Performance
  + Compatibility
  + Usability
  + Functional
* Which software properties can you measure or test, so you can say something about the quality of your project? (possible answers: functionality, performance, resource usage, usability, security, etcetera)
* Which activities can be performed with the goal to increase software quality? (possible answers: code reviews, static code analysis, all types of testing)
* Can you name some methods of testing to measure software quality properties? (possible answers: unit testing, e2e test, performance test, pen testing, usability testing)
* What is an indicator to determine how much of my code has been tested? (possible answer: coverage)
* Should you test all your code and in exactly the same way? (possible answers: no, risk-based)
* What are the advantages and disadvantages of automated testing?

**Advantages**

1. **Reliable:** Tests perform precisely the same operations each time they are run, thereby eliminating human error
2. **Repeatable:** You can test how the software reacts under repeated execution of the same operations.
3. **Programmable:** You can program sophisticated tests that bring out hidden information from the application.
4. **Comprehensive:** You can build a suite of tests that covers every feature in your application.
5. **Reusable:** You can reuse tests on different versions of an application, even if the user interface changes.
6. **Better Quality Software:** Because you can run more tests in less time with fewer resources
7. **Fast:** Automated Tools run tests significantly faster than human users.
8. **Economical:** As the number of resources for regression test are reduced.

**Disadvantages of Automated Testing:**

1. Proficiency is required to write the automation test scripts.
2. Debugging the test script is major issue. If any error is present in the test script, sometimes it may lead to deadly consequences.
3. Test maintenance is costly in case of playback methods. Even though a minor change occurs in the GUI, the test script must be re-recorded or replaced by a new test script.
4. Maintenance of test data files is difficult, if the test script tests more screens.

* Which security issues are to be considered for a (distributed) web application?
* Injection.
* Broken Authentication.
* Sensitive Data Exposure.
* XML External Entities (XXE).
* Broken Access Control.
* **Security** Misconfiguration.
* Cross-**Site** Scripting XSS.
* How can I test the basic performance of my application?
* What are the tools and frameworks that can help me test and analyze the quality of my code?

# Software Release Management

## Expected outcomes:

* You have version control set-up for your personal project and used it appropriately.
* You have built a CI/CD pipeline that automates all software development activities relevant to your project.
* You have integrated the tooling to measure SQA (e.g. performance, e2e, unit tests, etc) in your CI/CD pipeline. These tools you will discover in [What is Quality Assurance and why is it important?](https://fhict.instructure.com/courses/10457/modules/items/542275)
* You have set-up your CI/CD pipeline following the principle of  'Configuration as Code'.
* You have containerized your software using technology like e.g. Docker and perform deployment from within your CI/CD pipeline
* You make use of multiple branches in your software version management of which you can explain why these branches are relevant in your project.

## Questions:

* What is a CI/CD pipeline?
* Which software development activities can be integrated and automated in a CI/CD pipeline?
* What is the difference between CI and CD?
* What does the deployment of software mean?
* Why would I need multiple branches in my software version control system (GitLab)?
* What is 'Containerized Software' and what are the advantages of using this technology?
* What is ‘Configuration by Code’?