

CURRICULUM

for

Bachelor's Degree Programme in Software Development

Revised June 9 and August 24 2017

Content

1. The programme's goals for learning outcomes.....	3
2. The programme includes four national subject elements.....	3
2.1. Developing Large Systems.....	3
2.2. Databases for Developers.....	4
2.3. System Integration.....	5
2.4. Testing.....	6
2.5. The number of exams in the national subject elements.....	7
3. Internship	7
4. Requirements for the bachelor project.....	8
5. Rules on credit	8
6. Academic criteria for selecting candidates for top-up programmes	9
7. Commencement and transitional schemes	9

This national part of the curriculum for the Bachelor's Degree Programme in Software Development has been released in accordance with section 18(1) in the Ministerial Order for technical and commercial Academy Profession Programmes and Professional Bachelor Programmes. This curriculum is supplemented with an institutional component, provided by the institution offering the programme.

After it has been approved by either the Board of Directors (or the Rectors) and after consultation with the institutions' Educational Committee and the External Examiners' chairmanship for the specific programme, the educational network for the Bachelor's Degree Programme in Software Development prepares the institutional part.

1. The programme's goals for learning outcomes

Knowledge

The student must have knowledge of:

- The strategic role of testing in system development
- Globalisation of software production
- System architecture and its strategic importance for the company's business
- Applied theory and methodology and common technologies within the domain
- Various database types and their applications.

Skills

The student can:

- Integrate IT systems and develop systems that support future integration
- Use contracts as a control and coordination mechanism in the development process
- Assess and select database systems, and design, redesign and optimise databases
- Plan and manage development processes involving many geographically separated project participants
- Identify links between applied theory, methods and technology and reflect on their suitability in various situations

Competencies

The student can:

- Plan and implement testing for large IT systems
- Engage in professional collaboration to develop large systems by applying common methods and technologies
- Familiarise themselves with new technologies and standards for handling integration between systems,
- Through practice, develop their own competency profile from a primarily back-end developer profile to performing tasks as a system architect
- Handle the establishment and realisation of a business and technologically appropriate architecture for large systems.

2. The programme includes four national subject elements

2.1. Developing Large Systems

Developing Large Systems

Scope: 10 ECTS
Content: The aim of the subject element is to TRAIN the student to develop large-scale IT systems, where scalability is a key characteristic. The student must have knowledge of how key system development methods handle issues related to scalability and the development of large distributed systems. The student must have knowledge of concepts, techniques and technologies for the continuous integration and delivery of software-based systems. The student must be able to design, implement, and maintain large distributed systems in distributed development teams.
Learning objectives: <i>Knowledge</i> The student must have knowledge of: <ul style="list-style-type: none"> • Issues related to the development of distributed and large-scale IT systems, and how disciplined and agile development methods prescribe how these issues should be handled • The advantages, disadvantages and costs of using a system for the continuous integration and delivery of IT systems • Quality criteria for the design of interfaces to subsystems • Configuration and error reporting systems dedicated to the development of large distributed systems <i>Skills</i> The student can: <ul style="list-style-type: none"> • Apply techniques for dividing a system into subsystems • Design and specify requirements for subsystems • Use version control systems dedicated to the development of large distributed systems in a distributed development team • Use a system for continuous integration and delivery • Use architecture patterns dedicated to the development of large distributed systems. <i>Competencies</i> The student can: <ul style="list-style-type: none"> • Cooperate in large systems development organizations • Participate in globally distributed development • Adapt development methods and processes to the development of large distributed systems

2.2. Databases for Developers

Databases for Developers
Scope: 10 ECTS
Content: The aim of the subject element is to train the student to be able to select and apply various database types appropriately in relation to various fields of application. The student must also be able to analyse and develop in relation to large databases, including redesign and optimisation.
Learning objectives: <i>Knowledge</i>

The student must have knowledge of:

- Various database types and the underlying models
- A specific database system's storage organisation and query execution
- A specific database system's optimisation possibilities – including advantages and disadvantages
- Database-specific security problems and their solutions
- Concepts and issues in relation to data warehousing, including big data
- The particular issues raised by having many simultaneous transactions, including in connection with distributed databases
- Relational algebra (including its relationship to execution plans)

Skills

The student can:

- Transform logical data models into physical models in various database types
- Implement database optimisation
- Use parts of the administration tool to assist in the optimisation and tuning of existing databases, including the incorporation of a specific DBMS' execution plans
- Use a specific database system's tools for handling simultaneous transactions
- Use the programming and other facilities provided by a modern DBMS
- Use an object-relational mapping tool

Competencies

The student can:

- Analyse the application domain in order to select a database type
- Divide responsibility for tasks between the application and DBMS during system development, to ensure the best possible implementation.

2.3. System Integration

System Integration
Scope: 10 ECTS
Content: This subject element must help ensure that the student develops the competencies to be able to work with technical system integration. After completing this module, the student must be able to integrate existing systems in connection with the development of new systems, and develop new systems supporting future integration.
Learning objectives: <i>Knowledge</i> The student must have knowledge of: <ul style="list-style-type: none">• Business considerations in relation to system integration• Standards and standards organisations• Storage, transformation and integration of data sources

- The concept of services and its tie to service-oriented architectures
- Technologies which can be used to implement a service-oriented architecture
- Tools for integration.

Skills

The student can:

- Use an object-oriented system in a service-oriented architecture
- Design a system that is easy to integrate with other systems, and uses existing services
- Transform or expand a system so that it can function in a service-oriented architecture
- Use patterns that support system integration
- Integrate generic and other systems
- Choose from various integration methods
- Translate elements in a business strategy into specific requirements for system integration.

Competencies

The student can:

- Choose from various integration techniques
- Acquire knowledge of developments in standards for integration
- Adapt IT architecture to take into account future system integration.

2.4. Testing

Testing

Scope: 10 ECTS

Content: The aim of the subject element is to train students in planning and conducting testing. The student must understand the place and significance of testing in methods for system development. The student must be able to design and carry out systematic testing for large systems, including the establishment of automated testing. The student must also master concepts and techniques for the design and construction of testable systems.

Learning objectives:

Knowledge

The student must have knowledge of:

- Significant test strategies and models and their role in system development
- Testing as an integral part of a development project
- Various types of testing and their applications.

Skills

The student can:

- Ensure traceability between system requirements and testing at all levels
- Apply both black-box and white-box testing techniques
- Apply various criteria for the degree of test coverage
- Use techniques for verification and validation
- Use techniques and tools for automated testing

- Build systems to manage testing and the fault rectification process in development projects.

Competencies

The student can:

- Define, plan and carry out testing in a development project that matches the project's quality requirements
- Plan and manage the implementation of internal and external testing of software systems.
- Design testable systems

2.5. The number of exams in the national subject elements

There are 4 exams in the national subject elements, as well as one further exam in the bachelor project. For the number of exams in the internship, please refer to section 3.

For a comprehensive overview of all the programme's exams, please refer to the institutional part of the curriculum, as the national subject elements described in this curriculum can be examined together with the subject elements specified in the institutional part of the curriculum.

3. Internship

Learning objectives for the programme's internship

Internship
Scope: 15 ECTS
Content: The internship is organised so that it contributes – in combination with the rest of the study programme – to the student developing practical competencies. The aim of the internship is to enable the student to apply the programme's methods, theories and tools by performing specific practical software development tasks.
<p>Learning objectives:</p> <p><i>Knowledge</i></p> <p>The student must have knowledge of:</p> <ul style="list-style-type: none"> • Daily operations throughout the internship company. <p><i>Skills</i></p> <p>The student can:</p> <ul style="list-style-type: none"> • Apply versatile technical and analytical working methods linked to employment within the profession • Evaluate practice-oriented issues and identify possible solutions • Manage the structuring and planning of day-to-tasks within the profession • Communicate practice-oriented issues and reasoned solution proposals. <p><i>Competencies</i></p> <p>The student can:</p> <ul style="list-style-type: none"> • Handle development-oriented, practical and professional situations in relation to the

profession. <ul style="list-style-type: none"> • Acquire new knowledge, skills and competencies related to the profession • Participate in academic and interdisciplinary collaboration with a professional approach.
Number of examinations: 1

4. Requirements for the bachelor project

The learning objectives for the bachelor project are identical to the programme's learning objectives listed above under section 1.

The bachelor's project must document the student's understanding of and ability to reflect on the practices of the profession and the use of theory and methods in relation to a real-life problem. The problem statement, which must be central to the programme and profession, is formulated by the student, possibly in collaboration with a private or public company. The Academy approves the problem statement.

Bachelor project
Scope: 15 ECTS
Content: In their bachelor's project, the student must document the ability to work with a complex and practice-oriented issue in relation to a specific IT project, using an analytical and methodological basis.
Learning objectives: The final bachelor project must demonstrate that the programme's graduation level has been reached, see chapter 1 of this document.
Assessment: <ul style="list-style-type: none"> • The exam is an oral and written examination with an external co-examiner. A combined mark is given based on the 7-point scale for the written project and the oral presentation.

5. Rules on credit

Passed programme elements are equivalent to similar programme elements taken at other educational institutions offering this programme.

Students are obliged to inform us of any completed educational elements from another Danish or foreign higher education programme or any jobs which are likely to provide credit.

The Academy approves credit, in each instance, on the basis of completed programme elements and any jobs which meet the objectives of the subjects, the educational part and the internship parts.

The decision is based on an academic assessment.

For prior credit approval of studies in Denmark or abroad, students are required to document each approved and completed programme element on the completion of these studies.

In connection with applying for prior credit approval, the students give the Academy permission to obtain the necessary information after the student's completion.

Following approval according to the above, the programme element is deemed to be passed if it was passed according to the rules of the programme in question.

6. Academic criteria for selecting candidates for top-up programmes

Having completed a computer science degree fulfils the formal admission requirements for the professional bachelor programme in software development.

If there are more applicants than student places, applicants will be according to the following criteria:

- Average grade from the qualifying study programme
- Grades and ECTS credits in programming and system development
- Relevant work experience

7. Commencement and transitional schemes

Commencement

All enrolled students will be transferred to this curriculum on 1 September 2017.

Simultaneously, previous joint national curricula are NOT valid from this date.