SYLLABUS

1. Information regarding the programme

1.1 Higher education institution	**Babeş Bolyai University**
1.2 Faculty	**Faculty of Mathematics and Computer Science**
1.3 Department	**Department of Computer Science**
1.4 Field of study	**Computer Science**
1.5 Study cycle	**Bachelor**
1.6 Study programme / Qualification	**Computer Science**

2. Information regarding the discipline

2.1 Name of the discipline (en) (ro)	**Systems for Design and Implementation (Medii de proiectare și programare)**					
2.2 Course coordinator		**Lect. Prof. PhD. Radu D. Găceanu**				
2.3 Seminar coordinator		**Lect. Prof. PhD. Radu D. Găceanu**				
2.4. Year of study	**2**	2.5 Semester **4**	2.6. Type of evaluation **E**	2.7 Type of discipline	**Compulsory**
2.8 Code of the discipline					

3. Total estimated time (hours/semester of didactic activities)

3.1 Hours per week	4	Of which: 3.2 course	2	3.3 seminar/laboratory	2	
3.4 Total hours in the curriculum		56	Of which: 3.5 course	28	3.6 seminar/laboratory	28
Time allotment:						hours
Learning using manual, course support, bibliography, course notes						20
Additional documentation (in libraries, on electronic platforms, field documentation)						30
Preparation for seminars/labs, homework, papers, portfolios and essays						25
Tutorship						5
Evaluations						14
Other activities:						-
3.7 Total individual study hours		94				
3.8 Total hours per semester		150				
3.9 Number of ECTS credits		6				

4. Prerequisites (if necessary)

4.1. curriculum	ᵒ Advanced Programming Methods ᵒ Databases ᵒ Distributed Operating Systems	ᵒ
4.2. competencies	ᵒ Average programming skills in a high level programming language ᵒ Basic concepts of databases ᵒ Basic concepts of networking	

5. Conditions (if necessary)

5.1. for the course	ᵒ Projector	ᵒ
5.2. for the seminar /lab activities	ᵒ Laboratory with internet access and ability to use personal laptops.	

6. Specific competencies acquired

Professional competencies	C2.1 Identification of suitable methodologies for developing software systems. C2.2 Identification and explanation of suitable mechanism for software systems specification C2.3 Usage of methodologies, specification mechanisms and	
ransversal competencies	CT1 Application of rules for organized and efficient work, of responsible attitudes towards education-scientific domain for creative evaluation of self-potential, respecting the professional ethics principles and norms. CT2 Efficient development of activities organized in a inter-disciplinary group and the development of emphatic abilities of inter-human communication, relationships and collaboration with different groups CT3 Usage of efficient learning, information, research and development methods and techniques for knowledge revaluation abilities, for adaptation to the requirements of a dynamic society, and for communication in Romanian language and another foreign language.	

7. Objectives of the discipline (outcome of the acquired competencies)

7.1 General objective of the discipline	⚬ To understand distributed software concepts and problems ⚬ Improved design and programming skills	⚬
7.2 Specific objective of the discipline	⚬ To have a systematic knowledge concerning application development methodologies ⚬ To be familiarized with modern concepts and preoccupations in the field of developing application software ⚬ To know the use of computer-aided software development tools	

8. Content

8.1 Course	Teaching methods	Remarks	
1. Inversion of Control e.g: Spring	Presentation, conversation, case studies		
2. Object Relational Mapping e.g: Hibernate, Entity Framework	Presentation, conversation, case studies		
3. Client-server applications; Proxy pattern	Presentation, conversation, case studies		
4. Remote Procedure Call e.g: RMI, Remoting, Spring Remoting	Presentation, conversation, case studies		
5. Remote Procedure Call (cont.)	Presentation, conversation, case studies		
6. Enterprise Application Integration e.g: Protocol buffers, protobuf, thrift, amqp, rabbitmq, jms	Presentation, conversation, case studies		
7. Enterprise Application Integration (cont.)	Presentation, conversation, case studies		
8. Enterprise Application Integration (cont.)	Presentation, conversation, case studies		
9. Web services e.g: REST	Presentation, conversation, case studies		
10. Web services (cont.)	Presentation, conversation, case studies		
11. Web Sockets	Presentation, conversation, case studies		

12. Rich Internet Applications	Presentation, conversation, case studies	
13. Securing web applications and services	Presentation, conversation, case studies	
14. Patterns for web applications	Presentation, conversation, case studies	

Bibliography
1. Joseph Albahari and Ben Albahari, C# 6.0 in a Nutshell, Sixth Edition, O'Reilley, 2015.
2. Larman, C.: Applying UML and Design Patterns: An Introduction to OO Analysis and Design and Unified Process, Berlin, Prentice Hall, 2002.
3. Fowler, M., Patterns of Enterprise Application Architecture, Addison-Wesley, 2002.
4. Hohpe, G., Woolf, B., Enterprise integration patterns, Addison-Wesley, 2003.
5. ***, Microsoft Developer Network, Microsoft Inc., http://msdn.microsoft.com/
6. ***, The Java Tutorial, SUN Microsystems, Inc. http://download.oracle.com/javase/tutorial/
7. Eckel, B., Thinking in Java, 4th edition, Prentice Hall, 2006
8. Walls, Craig, Spring in Action, Fourth Edition, Ed. O'Reilley, 2015.
9. Spring http://projects.spring.io/spring-framework/

8.2 Seminar / laboratory	Teaching methods	Remarks
1. Application assignment.	Conversation, case studies, evaluation	
2. Inversion of Control	Conversation, case studies, evaluation	
3. Persistence. XML, DB, ORM	Conversation, case studies, evaluation	
4. Designing and implementing services; proxy pattern.	Conversation, case studies, evaluation	
5. RMI/Remoting	Conversation, case studies, evaluation	
6. RMI/Remoting	Conversation, case studies, evaluation	
7. Enterprise Application Integration	Conversation, case studies, evaluation	
8. Enterprise Application Integration	Conversation, case studies, evaluation	
9. Web Services	Conversation, case studies, evaluation	
10. Web Sockets	Conversation, case studies, evaluation	
11. Web Sockets	Conversation, case studies, evaluation	
12. Web applications. Frameworks	Conversation, case studies, evaluation	
13. Web Security	Conversation, case studies, evaluation	
14. Recap. Evaluation	Conversation, case	

	studies, evaluation	

Bibliography
1. Joseph Albahari and Ben Albahari, C# 6.0 in a Nutshell, Sixth Edition, O'Reilley, 2015.
2. ***, Microsoft Developer Network, Microsoft Inc., http://msdn.microsoft.com/
3. ***, The Java Tutorial, SUN Microsystems, Inc. http://download.oracle.com/javase/tutorial/
4. Walls, Craig, Spring in Action, Fourth Edition, Ed. O'Reilley, 2015.
5. Spring http://projects.spring.io/spring-framework/

9. Corroborating the content of the discipline with the expectations of the epistemic community, professional associations and representative employers within the field of the program

- ෆ The course fulfils the IEEE and ACM Curricula Recommendations for Computer Science studies;
- ෆ The content of the course is considered by software companies as being important for average design and advanced programming skills .

10. Evaluation

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Share in the grade (%)	
10.4 Course	- To know the basic concepts of developing distributed applications	Written exam	40%	
	- To apply these concepts to design and implement a small distributed application	Practical exam	40%	
10.5 Seminar/lab activities	Being able to design and implement distributed applications using various technologies	Practical examination, observation documentation	20%	
10.6 Minimum performance standards				
At least grade 5 (1 to 10 scale) at all activities seminar/lab, written exam, practical exam (and the final grade at least 5).				

Date	Signature of course coordinator	Signature of seminar coordinator
29.04.2016	Lect. Prof. PhD. Radu D. Găceanu	Lect. Prof. PhD. Radu D. Găceanu

Date of approval   Signature of the head of department

....…..........................