SYLLABUS

1. Information regarding the programme

1.1 Higher education institution	**Babes-Bolyai University**
1.2 Faculty	**Mathematics and Computer Science**
1.3 Department	**Computer Science**
1.4 Field of study	**Computer Science**
1.5 Study cycle	**Master**
1.6 Study programme / Qualification	**Software engineering**

2. Information regarding the discipline

2.1 Name of the discipline	Design of interactive software systems						
2.2 Course coordinator	Ph. D. Lecturer Adriana-Mihaela Guran						
2.3 Seminar coordinator	Ph. D. Lecturer Adriana-Mihaela Guran						
2.4. Year of study	**1**	2.5 Semester	**2**	2.6. Type of evaluation	**E**	2.7 Type of discipline	**Compulsory**

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	1
3.4 Total hours in the curriculum	4	Of which: 3.5 course	2	3.6 seminar/laboratory	2
Time allotment:					hours
Learning using manual, course support, bibliography, course notes					30
Additional documentation (in libraries, on electronic platforms, field documentation)					42
Preparation for seminars/labs, homework, papers, portfolios and essays					40
Tutorship					5
Evaluations					2
Other activities:					

3.7 Total individual study hours	119
3.8 Total hours per semester	175
3.9 Number of ECTS credits	7

4. Prerequisites (if necessary)

4.1. curriculum	•
4.2. competencies	•

5. Conditions (if necessary)

5.1. for the course	• A room with Internet access and presentation devices
5.2. for the seminar /lab activities	• A room with computers and Internet access

6. Specific competencies acquired

Professional competencies	• Knowledge, understanding and use of basic concepts of theoretical Computer Science • Understanding of basic concepts of mathematics and use them to problem-solving activities. • Knowledge, understanding and use of the fundamental methods, processes and tools of software engineering • Ability to understand and approach problems of modeling nature from other sciences • Ability to work independently and/or in a team in order to solve problems in defined professional contexts.
Transversal competencies	•

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

7.1 General objective of the discipline	• To design/develop usable systems • To be able to evaluate the usability of a system and to be able to find improvement methods regarding the usability of a system • To be aware of accessibility issues in systems design • To be able to develop multiplatform applications
7.2 Specific objective of the discipline	At the end of the semester students must be able to: • understand the human capacities in interaction with software systems • achieve knowledge of psychological aspects of human-computer interaction • understand the importance of real users in the development of interactive systems • indentify the best communication methods with the clients • use knowledge and models form sociological sciences in order to improve the communications with clients • apply interaction modeling methods

	• apply task analysis methods • be able to apply user centered design • achieve usability related notions • be able to apply automatic usability evaluation methods • design and develop groupware systems • apply interaction design patterns • use interface abstract description languages • develop plastic user interfaces • develop adaptive and adaptative user interfaces

8. Content

8.1 Course	Teaching methods	Remarks
1. Human-Computer Interaction • Human factor in HCI • Input/output channels • Capacities, limitations • Computer • Output devices • Virtual reality and 3D devices • Non-conventional interaction devices	Presentation, discussions, case studies, problem solving	

2. Interaction models
 	
• Interaction styles
• WIMP Interfaces
• Interaction paradigms
• Interaction design
  ○ Interaction models
  ○ Cognitive models
  ○ Linguistic models
  ○ Physical models
	idem	

3. Interaction design process • Scenarios, personas • Navigation design • Prototyping • Interface prototyping tools • Interaction design guidelines • Usability principles • Standards • Rules • Heuristics	idem	

• Interaction and interfaces design patterns • Interaction design patterns • Web applications design patterns • Mobile application interfaces design patterns		
4.Task analysis • Fundamentals of task analysis • Task analysis methods: HTA, GTA • Task analysis tools:EUTERPE, CTTE • Task analysis in requirements engineering	idem	
5. Dialog description languages and notations • Seeheim dialog model • Diagramatic languages :STN, RTN, CCT, statecharts • Textual notations:BNF, TAG, ETAG, Petri nets	idem	
6. Graphic elements in user interfaces • Basic graphic elements in UIs • Criteria and recommendations for graphic elements	idem	
7-8. Usability engineering • What is usability? • Designing for usability • Multidisciplinary approaches for user needs identification: o Interviews o Focus-groups o Observation o questionnaires • Usability testing o Usability metrics o Automatic usability evaluation o Usability questionnaires • Usability in system development models evolution: • Waterfall model • Iterative model • RAD • Agile models	idem	
9. User interface testing • user interface testing tools • automatic user interface testing tools	idem	
10. GROUPWARE design • CSCW and Groupware • DUTCH method	idem	
11. Emotions in interaction design • Expressive interfaces • Antropomorphic interfaces	idem	

• User experience		
12. Intelligent interfaces • Interface agents • Adaptive systems • Adaptative systems	idem	
13. Plastic interfaces/ubiqitous computing (ubicomp) • Abstract desvcrption languages for UIs:UsiXML, SUNML, UIML,… • Development tools for platic interfacese:TERESA	idem	
14. Designing for accesibility • Disabilities • Interaction devices for people with disabilities • Accessibility guidelines • Accessibility APIs • Accessibility testing tools	idem	

Bibliography:

1. Alan Dix, Janet Finlay, Gregory D Abowd, Russell Beale - Human-Computer Interaction,

Prentice Hall, third edition, 2004

2. Donald A. Norman - Emotional Design - Why we love (or hate) everiday things, 2004

3. Martijn van Welie - Task-based User Interface Design, 2001

4. Donald A Norman - The design of everyday things, basic Books, 1988

5. Fabio Paterno - Model-based design and evaluation of interactive applications, Springer, 1999

6. Jennifer Tidwell - Designing Interfaces: Patterns for Effective Interaction Design, O@Reilly, 2005

7. Jacob Nielsen - Usability Engineering, Academic Press, 1993

8. Marc Hassenzahl- Experience Design: Technology For All The Right Reason, Morgan & Claypool,2010

8.2 Seminar / laboratory	Teaching methods	Remarks
1. Students will have to identify two examples of bad designed user interfaces for desktop applications, web applications and mobile applications and will have to justify their choice and to propose enhancements		
2. Students will have to chose a project refering a medium		

size application that will have to run on multiple platforms (week 2-4)		
3. Project design and development : 1. Users identification 2. Justify the need for the proposed product 3. Functionalities identification (Week 4-Week 6) 4. Finding the functional specification 5. Crearea unui prototip al sistemului 6. Prototype evaluation with real users (Week 8) 7. Applying the required changes to the prototype 8. User interface testing (Week 10) 9. Usability testing (Week 11) 10. User interface generation for multiple devices (Week 12) 11. Applying changes to the developed product in order to be used by people with disabilities/ designing an non-conventional interaction method to the product (Week 14)		

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

- Students will be able to design interaction for multiple devices in a user centred approach and to evaluate the quality of their prototypes, enhancing the quality of the developed products

10. Evaluation

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Share in the grade (%)
10.4 Course			
10.5 Seminar/lab activities	Project – students will be graded based on the quality of their projects	Grading will be done for every stage in project development, the final grade will be computed based on the following criteria: • Use of design principles (30%) • Usability of the application (measured using a method from the literature) (30%) • Ability of the	100%

		application to run on multiple platforms (25%) • Accessibility (15%)	
10.6 Minimum performance standards			
➢ Students have to deliver a working software product that satisfies the client requirements.			

Date	Signature of course coordinator	Signature of seminar coordinator
14.05.2013	Ph. D. Lecturer Adriana Guran	Ph. D. Lecturer Adriana Guran

Date of approval

.....

Signature of the head of department

Ph. D. Prof. Bazil Pârv