

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	3	Of which: 3.2 course	2	3.3 seminar/laboratory	1	
3.4 Total hours in the curriculum	42	Of which: 3.5 course	28	3.6 seminar/laboratory	14	
Time allotment:						hours
Learning using manual, course support, bibliography, course notes						40
Additional documentation (in libraries, on electronic platforms, field documentation)						45
Preparation for seminars/labs, homework, papers, portfolios and essays						41
Tutorship						5
Evaluations						2
Other activities:						
3.7 Total individual study hours			133			
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	<ul style="list-style-type: none"> • Understanding and working with basic concepts in software engineering; • Capability of analysis and synthesis; • Modeling and solving real-life problems;
Transversal competencies	<ul style="list-style-type: none"> • Assimilation of mathematical concepts and formal models to understand, verify and validate software systems ; • Analysis, design, and implementation of software systems • Proficient use of methodologies and tools specific to programming languages and software systems • Organization of software production processes

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

7.1 General objective of the discipline	<ul style="list-style-type: none"> • To understand and approach problems of modeling nature from other sciences • 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	<p>At the end of the semester students must be able to:</p> <ul style="list-style-type: none"> • 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 • 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

	<ul style="list-style-type: none"> • use interface abstract description languages • develop plastic user interfaces • develop adaptive and adaptative user interfaces
--	--

8. Content

8.1 Course	Teaching methods	Remarks
1. Interaction Design <ul style="list-style-type: none"> • What is ID? • HCI and ID • UCD and ID • Professions in ID 		
2. Basics of Human-Computer Interaction <ul style="list-style-type: none"> • Human factor in HCI • Input/output channels • Capacities, limitations • Computer • Output devices • Virtual reality and 3D devices • Non-conventional interaction devices • Usability • Errors 	Presentation, discussions, case studies, problem solving	
3. Interaction Design Process <ul style="list-style-type: none"> • Interaction design lifecycle models • Needs and requirements • Interaction models • Interaction style 	idem	
4. Task Analysis <ul style="list-style-type: none"> • Fundamentals of task analysis • Task analysis methods: HTA, GTA • Task analysis tools: EUTERPE, CTTE • The Bridge Method 	idem	
5. Interdisciplinary approaches in Usability Engineering – a focus on user needs analysis <ul style="list-style-type: none"> • Qualitative vs quantitative approaches • A framework for early usability integration in the development of interactive software systems 	idem	
6. Conceptual Models in Interaction Design <ul style="list-style-type: none"> • Conceptual models • Metaphors • Interaction modes 	idem	
7. Prototyping vs. Pretotyping <ul style="list-style-type: none"> • prototyping 	idem	

<ul style="list-style-type: none"> • prototyping • case studies 		
8. Graphic elements in user interfaces <ul style="list-style-type: none"> • Basic graphic elements in UIs • Criteria and recommendations for graphic elements • Focus, flow and layout in UI design 	idem	
9. Usability engineering & User experience <ul style="list-style-type: none"> • What is usability? • Designing for usability • Usability engineering lifecycle • Usability metrics • Measuring user experience 	idem	
10. Designing for accessibility <ul style="list-style-type: none"> • Disabilities • Accessibility • Accessibility APIs • Accessibility evaluation 	idem	
11. GROUPWARE design <ul style="list-style-type: none"> • CSCW and Groupware • DUTCH method 	idem	
12. Designing Gestural Interfaces <ul style="list-style-type: none"> • Gesture definition • Gesture in everyday life • Designing interactive gestures • Interface conventions 	idem	
13. Designing Social Interfaces <ul style="list-style-type: none"> • History of computer mediated interaction • Social network vs. social media 	idem	
14. User Interface Testing <ul style="list-style-type: none"> • GUI Testing • Model-based testing • Web applications testing 	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) everyday 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
9. Alberto Savoya – Prototyping IT, 2011
10. Tom Tullis, William Albert – Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2008

8.2 Seminar / laboratory	Teaching methods	Remarks
<p>Students will have to choose a project subject referring a medium size application that will be developed using a user centered approach (week 4). The project will be developed in teams of 3-5 members</p>		
<p>Project design and development :</p> <ol style="list-style-type: none"> 1. Users identification 2. Justify the need for the proposed product 3. Functionalities identification (Week 4-Week 6) 4. Building the conceptual model 5. Creating prototypes 6. Evaluating prototypes (Week 8) 7. Applying the required changes to the prototype 8. Usability testing (Week 11) 9. 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 based on 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: <ul style="list-style-type: none"> • Use of appropriate methods to identify user needs (25%) • Use of design principles (30%) • Usability of the application (measured using a method from the literature) (30%) • Accessibility (15%) 	100%
10.6 Minimum performance standards			
➤ Students have to deliver a working software product that satisfies the client requirements.			

Date

25.04.2016

Signature of course coordinator

Ph. D. Lecturer Adriana Guran

Signature of seminar coordinator

Ph. D. Lecturer Adriana Guran

Date of approval

.....

Signature of the head of department

Ph. D. Prof. Anca Andreica