

OPIS PRZEDMIOTU/MODUŁU KSZTAŁCENIA (SYLABUS)

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Projekt aplikacji mobilnej 1
2.	Dyscyplina Informatyka techniczna i telekomunikacja - 5
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Wydział Fizyki i Astronomii
5.	Kod przedmiotu/modułu 24-ISSP-S1-E6-PAM1
6.	Rodzaj przedmiotu/modułu (<i>obowiązkowy lub do wyboru</i>) obowiązkowy
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka stosowana i systemy pomiarowe
8.	Poziom studiów (<i>I lub II stopień, jednolite studia magisterskie, studia doktoranckie</i>) I stopień
9.	Rok studiów (<i>jeśli obowiązuje</i>) 3
10.	Semestr (<i>zimowy lub letni</i>) letni
11.	Forma zajęć i liczba godzin Laboratorium 30 godz. Metody nauczania Metoda projektowa
12.	Imię, nazwisko, tytuł/stopień naukowy osoby prowadzącej zajęcia Jakub Poła, mgr
13.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu <ul style="list-style-type: none">• podstawy programowania, co najmniej jeden z poniższych:<ul style="list-style-type: none">◦ Java dla platformy Android◦ Kotlin◦ C#◦ C++◦ JavaScript• znajomość podstaw programowania urządzeń mobilnych

	<ul style="list-style-type: none"> • umiejętność pracy w grupie
14.	<p>Cele przedmiotu</p> <p>Realizacja projektu programistycznego przeznaczonego na platformę Android. Przygotowanie dokumentacji projektowej, planu realizacji, analizy technologii.</p> <p>Student realizuje, samodzielnie bądź w grupie, projekt aplikacji przeznaczonej na urządzenia mobilne. Realizacja projektu powinna składać się z co najmniej trzech etapów:</p> <ol style="list-style-type: none"> 1. Planowanie projektu, analiza technologii, analiza wykonalności poszczególnych komponentów. 2. Realizacja projektu, kodowanie, „review meetings” 3. Prezentacja projektu. <p>Ad. 1. W pierwszym etapie (~ 6 tygodni) student powinien poświęcić na poszukiwaniu technologii, która wg niego będzie najlepsza do realizacji wybranego projektu. Powinna być to analiza języków programowania, bibliotek oraz frameworków. Na tym etapie powinny powstać krótkie programy pozwalające mu zdecydować, która z technologii będzie najlepsza. Po przeprowadzonej analizie student spisuje cel i założenia projektu, wykorzystane technologie oraz funkcjonalności swojego projektu. Prezentuje szkice ekranów oraz zasadę działania („workflow”) aplikacji. Dokument dostarcza do akceptacji prowadzącemu (stakeholderowi). Prowadzący może sugerować pewne zmiany (zwykle upraszczające). Po uzgodnieniu zmian dokument podpisany jest przez obie strony jako umowa realizacji projektu. Podpisana dokumentacja stanowi podstawę do rozliczenia zadania. Wszelkie zmiany powinny być zgłaszane prowadzącemu zajęcia (tutaj w roli klienta), te mogą, ale nie muszą być wprowadzone do projektu. Istotnym elementem tej części projektu jest przygotowanie dobrej jakości dokumentu.</p> <p>(W 90% dostarczane przez studentów dokumenty są fatalnej jakości pod względem merytorycznym, stylistycznym oraz technicznym samego dokumentu. Warto się skupić również nad tym aspektem. Umowy, dokumentacje, tzw. design documents różnego rodzaju są istotnymi elementami tworzenia oprogramowania.)</p> <p>Ad2. Przez kolejnych ~8 tygodni student realizuje projekt używając zadeklarowanej technologii, bibliotek, frameworków i innych narzędzi zadeklarowanych w dokumentacji. Co 4 tygodnie odbywają się spotkania z prowadzącym, na których student prezentuje progres prac (tzw „review meeting”). Idealnie byłoby, gdyby na każde spotkanie student przygotował tzw. MVP (Minimum Viable Product) tj. program/komponent w minimalnym stopniu realizujący zestaw wybranych funkcjonalności produktu końcowego.</p> <p>Ad 3. Prezentacja projektu. Na forum grupy student prezentuje efekty swoich prac. Prezentacja projektu powinna zawierać 2 elementy:</p> <ol style="list-style-type: none"> 1. Założenia projektu bazujące na podpisanym na początku semestru planie. 2. Działający produkt w formie live, demo lub nagranych wcześniej filmu prezentującego sposób realizacji poszczególnych funkcjonalności projektu. <p>(Mówienie o swoim projekcie to istotna część tworzenia oprogramowania. Takie prezentacje pozwalają skonfrontować plany z rzeczywistością.)</p>
15.	<p>Treści programowe</p> <ul style="list-style-type: none"> • Planowanie projektu • Wymagania funkcjonalne/niefunkcjonalne aplikacji • Design documents

	<ul style="list-style-type: none"> Prezentacje projektu 	
16.	Zakładane efekty uczenia się <ul style="list-style-type: none"> Potrafi zaplanować i zrealizować zaproponowany projekt programistyczny w wybranej technologii. Stworzy niezbędną dokumentację projektową. Zaprezentuje efekty swojej pracy w formie multimedialnej prezentacji. Zorganizuje pracę zespołu projektowego 	Symbole odpowiednich kierunkowych efektów uczenia się: I1_W04, I1_W08, I1_W10 I1_U06, I1_U07, I1_U08, I1_U11, I1_U16 I1_K01, I1_K02, I1_K04
17.	Literatura obowiązkowa i zalecana (<i>źródła, opracowania, podręczniki itp.</i>) <ul style="list-style-type: none"> “Android Programming, The Big Nerd Ranch Guide” Bill Phillips, Chris Stewart, Kristin Marsicano Head First Android Development: A Brain-Friendly Guide, Dawn Griffiths, David Griffiths 	
18.	Metody weryfikacji zakładanych efektów uczenia się: przygotowanie i zrealizowanie projektu (indywidualnego lub grupowego)	
19.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu: - wystąpienie ustne (indywidualne lub grupowe), - przygotowanie i zrealizowanie projektu (indywidualnego lub grupowego)	
20.	Nakład pracy studenta/doktoranta	
	forma działań studenta/doktoranta	liczba godzin na realizację działań
	Zajęcia (wg planu studiów) z prowadzącym: - wykład: - konwersatorium: - laboratorium: - inne:	0 0 30 0
	Praca własna studenta, doktoranta (w tym udział w pracach grupowych): - przygotowanie do zajęć: - opracowanie wyników: - czytanie wskazanej literatury: - przygotowanie prac/wystąpień/projektów: - napisanie raportu z zajęć: - przygotowanie do sprawdzianów i egzaminu:	0 0 0 90 5
	łączna liczba godzin	125
	Liczba punktów ECTS	5