

# Implementacija tehnologije udaljenog nadzora ispitanika pri e-testiranju znanja

---

Luketić, Matej

Master's thesis / Diplomski rad

2021

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Rijeka, Faculty of Humanities and Social Sciences / Sveučilište u Rijeci, Filozofski fakultet**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:231:233706>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-01-09**

*Repository / Repozitorij:*

[Repository of the University of Rijeka University  
Studies, Centers and Services - RICENT Repository](#)



**SVEUČILIŠTE U RIJECI**

**Studij politehnike**

**Matej Luketić**

**IMPLEMENTACIJA TEHNOLOGIJE  
UDALJENOG NADZORA ISPITANIKA PRI  
E-TESTIRANJU ZNANJA**

**Diplomski rad**

**Rijeka, 2021.**

SVEUČILIŠTE U RIJECI

**Studij politehnike**

Diplomski sveučilišni studij Politehnika i informatika

**IMPLEMENTACIJA TEHNOLOGIJE  
UDALJENOG NADZORA ISPITANIKA PRI  
E-TESTIRANJU ZNANJA**

Diplomski rad

Mentor: Doc. dr. sc. Damir Purković

Student: Matej Luketić

JMBAG: 9998000523

Rijeka, 2021.

## Zadatak za diplomski rad

Pristupnik: **Matej Luketić**

Naziv diplomskog rada: *Implementacija tehnologije udaljenog nadzora ispitanika pri e-testiranju znanja*

Naziv diplomskog rada na eng. jeziku: *Application of Remote Monitoring Technology of Respondents during E-testing of Knowledge*

Sadržaj zadatka:

Rješenjem zadatka je potrebno obuhvatiti sljedeće:

1. Uvodni dio – važnost primjene suvremene tehnologije za testiranje znanja, teorijska osnova testiranja i e-testiranja znanja, problemi i izazovi pri e-testiranju znanja;
2. Koncepti i sustavi za elektroničko testiranje znanja, mehanizmi i sustavi koji osiguravaju valjanost i pouzdanost e-testiranja;
3. Metode i tehnologije udaljenog (video) nadzora i mogućnosti implementacije za e-testiranje znanja;
4. Razrada vlastitog modela ili sustava udaljenog video nadzora ispitanika pri e-testiranju znanja;
5. Metodički dio: Implementacija razvijenog sustava u okviru odabranog kurikulumu tehničkog obrazovanja, opis primjene e-testiranja, priprema za nastavu;
6. Zaključak.

Mentor: **Doc. dr. sc. Damir Purković**



(potpis mentora)

Komentor: (ime i prezime)

---

Voditelj za diplomske radove

---

Zadatak preuzet: 17.05.2021.

(potpis pristupnika)

## **IZJAVA**

Izjavljujem da sam Završni rad napisao samostalno, znanjem koje sam stekao na diplomskom studiju politehnike i informatike u Sveučilištu u Rijeci služeći se navedenim izvorima te uz stručno vodstvo doc. dr. sc. Damira Purkovića kome se srdačno zahvaljujem.

Posebno se zahvaljujem svojoj obitelji i roditeljima te djevojci Matei, koji su bili tu uz mene i davali mi podršku bez obzira bilo to u sretnim ili teškim trenutcima te bez kojih ne bi bilo moguće napisati ovaj diplomski rad.

---

Matej Luketić

# SADRŽAJ

SAŽETAK .....	III
1. Uvod.....	5
2. Bloomova taksonomija kao temelj e-testiranja znanja .....	7
2.1 Originalna Bloomova taksonomija .....	7
2.2 Revidirana Bloomova taksonomija.....	8
3. Koncepti i sustavi za elektroničko testiranje znanja .....	10
3.1 Koncept WbeTS sustava .....	12
3.1.1 Modul Baza.....	14
3.1.2 Modul Kriteriji.....	14
3.1.3 Modul Mehanizam .....	14
3.1.3 Modul Sučelje .....	15
3.1.4 Postupak WbeTS e-testiranja.....	15
3.2 Koncept AHyCo sustava .....	19
3.2.1 Arhitektura AHyCo sustava.....	20
3.2.2 Provjera znanja.....	22
3.2.3 Prva skupina (prije početka rada) .....	23
3.2.4 Druga skupina (tijekom rada) .....	25
3.2.5 Treća skupina (nakon rada).....	25
3.3 Koncept fuzzy logike .....	27
3.3.1 Usporedba s konvencionalnom logikom.....	27
3.4 Item response theory (teorija odgovora na zadatke) .....	29
3.4.1 Dimenzionalnost latentnog prostora .....	30
3.4.2 Karakteristična krivulja zadatka .....	31
3.4.3 Brzina testa.....	34
3.4.4 Primjena IRT-a.....	34
3.5 Problemi koji onemogućuju potpuno autonomno testiranje .....	35
4. Metode i tehnologije udaljenog (video) nadzora te mogućnosti implementacije za e-testiranje znanja .....	37

4.1 Sustav za prepoznavanje lica (Face Recognition System) .....	37
4.2 Sustav za detekciju pokreta (Motion Detection System) .....	40
4.3 Metoda optičkog toka.....	42
4.4 HSV metoda .....	42
5. Razrada modela udaljenog nadzora ispitanika pri e-testiranju znanja.....	44
5.1 Provjera identiteta .....	44
5.2 Model udaljenog nadzora .....	45
5.3 Princip rada modela udaljenog nadzora .....	48
5.4 Moguće poteškoće pri implementaciji .....	49
6. Metodički dio .....	51
6.1 Nastavni plan i program (kurikulum).....	51
6.2 Pitanja za provjeru znanja .....	66
7. Zaključak.....	69
Literatura.....	70
Popis slika .....	72
Popis tablica .....	73

## SAŽETAK

Svrha ovog diplomskog rada je konceptualna implementacija tehnologije udaljenog nadzora ispitanika u postojeći elektronički sustav testiranja znanja, koja ima potencijal povisiti razinu valjanosti samog postupka testiranja. Kako bi se to ostvarilo, u radu se analiziraju određeni problemi dosadašnjih sustava za elektronsko testiranje te njihova moguća rješenja a nadzor bi se trebao odvijati u uvjetima široko dostupne tehnologije Interneta uz uporabu web kamere. Uz to je predstavljena Bloomova taksonomija kao temelj sustava za testiranje znanja bilo to klasičnim ili elektroničkim putem. Prikazani su koncept WbeTS sustava i koncept AHyCo sustava za testiranje znanja te njihovi sustavi i podsustavi. Na kraju ovog rada napisan je i metodički dio koji sadrži u sebi skraćeni prikaz sadržaja nastave informatike s ciljem uklapanja značajke WbeTS sustava kao i tehnologije udaljenog nadzora prilikom vrednovanja ostvarenosti ishoda, odnosno prilikom testiranja znanja elektroničkim putem, kao i pripadajuću pripremu za izvođenje nastave.

***Ključne riječi:*** elektroničko testiranje znanja, WbeTS sustav, Bloomova taksonomija, udaljeni (video) nadzor, trivijalna valjanost



# **Application of Remote Monitoring Technology of Respondents during E-testing of Knowledge**

## **SUMMARY**

The purpose of this thesis is the conceptual implementation of remote monitoring technology of respondents in the existing electronic system of knowledge testing, which has the potential to increase the level of validity of the testing process. In order to achieve this, the paper analyzes certain problems of previous electronic testing systems and their possible solutions, and monitoring should take place in the conditions of widely available Internet technology with the use of a webcam. In addition, Bloom's taxonomy is presented as the foundation of a knowledge testing system, whether classical or electronic. The concept of WbeTS knowledge testing system and the concept of AHyCo knowledge testing system and their systems and subsystems are presented. At the end of this paper, a methodological part was written which contains a brief overview of the content of computer science teaching with the aim of fitting the features of the WbeTS system and remote monitoring technology when evaluating the achievement of outcomes, ie when testing knowledge electronically, as well as the corresponding preparation for teaching.

**Key words:** electronic knowledge testing, WbeTS system, Bloom's taxonomy, remote (video) surveillance, trivial validity

## 1. Uvod

Na početku 2020. godine smatrali smo kako pandemija COVID-19 ne može doći do nas jer se događa u dalekoj Kini. No početkom trećeg semestra diplomskog studija COVID-19 je ipak stigao i do nas, a s njim i mnoštvo novih pojmova poput Lockdown-a koji je za studente u početku bio dvosjekli mač, jer predavanja su se pratila online putem te nije bilo potrebno biti fizički u predavaonici, i sve je bilo super dok nisu stigli prvi kolokviji. Tada je i većina studenata prvi put došla u kontakt s online ispitima. Nakon takvih iskustava i studenti su dobili uvid u to koliko je razvijen sam sustav e-testiranja znanja koji koriste učilišta u Republici Hrvatskoj.

E-testiranje je područje e-ocjenjivanja koje je s vremenom polako sve više dolazilo do izražaja a pogotovo sada u ovo doba pandemije i novih pravila. E-testiranje uključuje provođenje ispitivanja i ocjenjivanja na računalo, bilo pomoću lokalnih sustava ili sustava temeljenih na webu. E-test je svaki test koji ponavlja ili zamjenjuje testove temeljene na papiru s ekranom računala, koji se nazivaju i testovi na ekranu. To uključuje prijenos papirnatih testova na računalo s malo ili bez ikakve promjene izgleda na papiru. Daljnje mogućnosti e-testiranja uključuju korištenje potpune IKT interaktivnosti. Jedna od ključnih prednosti e-testiranja za učenike je ta što se odgovori na testu odmah šalju ispitivaču (nastavniku), a rezultati se mogu odmah dobiti putem automatiziranog sustava dodjele ocjena. Također, e-testiranje nudi mnogo veću fleksibilnost samim time što učenicima daje mogućnost pisanja testa kada to oni žele, kada su oni spremni a ne u neko zakazano vrijeme koje je nastavnik odredio. Fleksibilnost lokacije testiranja i vremena testiranja dovodi do većeg angažmana učenika. Ovakav način testiranja također ima i pozitivan učinak na okoliš jer je smanjena upotreba papira te a to doprinosi i uštedi troškova. Također postoje neki problemi i prepreke prilikom implementacije automatiziranog e-testiranja znanja poput same provedbe udaljenog nadzora ispitanika koja je ključna u postizanju trivijalne valjanosti kao i autonomije samog sustava (Purković i Ban, 2014).

Glavni cilj ovog rada je predodžba pregleda dosadašnjih sustava e-testiranja te metoda i tehnika udaljenog nadzora ispitanika te prijedlog implementacije sustava udaljenog nadzora u već postojeći sustav za elektroničko testiranje znanja. Ujedno će se govoriti o Bloomovoj taksonomiji koja je temelj određivanja ishoda. Kako bi se ti ishodi mogli provjeriti potreban nam je neki sustav provjere. Unatoč klasičnom sustavu provjere ostvarenosti ishoda, u ovom

radu ćemo se baviti alternativnom elektroničkom sustavu provjere znanja. Objasniti će se razni koncepti i sustavi kao i razne metode i tehnologije udaljenog (video) nadzora koji se koriste. Nakon toga će se razraditi cjeloviti WbeTS sustav te će se u njega konceptualno implementirati kombinacija tehnologija udaljenog nadzora. Na kraju rada u metodičkom dijelu će se opisati sadržaji pogodni za implementaciju i reducirani kurikulum predmeta te sva pripadajuća nastavna priprema i dokumentacija.

## **2. Bloomova taksonomija kao temelj e-testiranja znanja**

Kako bi koncepti i sustavi e-testiranja, koje će se u nastavku rada detaljnije proučiti, bili jasniji, reći ćemo nešto o Bloomovoj taksonomiji, originalnoj i revidiranoj. Bloomova taksonomija znanja jedan je od temeljnih teoretskih okvira za pripremu, planiranje i vrednovanje svih vrsta obrazovanja. Osnovni cilj Bloomove taksonomije znanja je izrada sustava koji polazi od logičkih, pedagoških i psiholoških zakonitosti te se učvršćuje principom učenja i poučavanja (Bloom, 1956). Iz toga možemo precizirati ishode, tj., ciljeve učenja. Određivanje ishoda, odnosno ciljeva učenja je jedan od početnih koraka izrade pripreme za nastavu, te samim time dobivamo i alat, mjeru za vrednovanje ostvarenosti ishoda po završetku nastave. Ciljevi učenja i ponašanja koja učenik koristi tijekom učenja razvrstani su u 3 kategorije koje su međusobno povezane i preklapaju se. Kategorije su područja učenja:

1. Kognitivno područje – intelektualne sposobnosti, znanje
2. Afektivno područje – vrijednosti i stavovi, interesi i uvjerenja
3. Psihomotoričko područje – motoričke sposobnosti, vještine

Iako ova područja u svijesti pojedinca čine jedinstvenu cjelinu, kategorizirana su zbog jednostavnije evaluacije postignuća u nekom sustavu obrazovanja.

### **2.1 Originalna Bloomova taksonomija**

Bloomova taksonomija znanja je nastala tijekom 50.ih godina 20.stoljeća na temelju analiza intelektualnih ponašanja učenika. U početku je bila orijentirana samo na kognitivno područje, a naknadno je definirana još afektivna i psihomotorička domena. Originalna verzija taksonomije sastojala se od 6 razina kod kojih svaka iduća razina uvjetuje ovladavanjem prethodnom (Krathwohl, D.R., 2002). To su bile slijedeće razine:

1. Pamćenje – najniža razina znanja, ujedno se zove i razina dosjećanja. Dosjećanje ne znači nužno i razumijevanje. Gradivo bez razumijevanja bit će neupotrebljivo i vrlo brzo će biti zaboravljeno. Neće potaknuti ni intelektualni razvoj djece, niti će djecu potaknuti i motivirati za autonomno učenje;

2. Razumijevanje – viša razina znanja, što znači razumijevanje informacija. Informacije koje se razumiju mogu se mijenjati iz jednog oblika u drugi. Izražava se kao mogućnost prepričavanja i objašnjavanja vlastitim riječima, sažimanja ključnih informacija te navođenja primjera te koncepata ili načela. Razumijevanje je preduvjet za sljedeće četiri razine;
3. Primjena – ova razina uključuje generalizaciju i može biti neovisno primijenjena u i izvan konteksta u kojem je dobivena na konvencionalan i novi način s kojim se dijete još nije susrelo;
4. Analiza – ova razina uključuje razlikovanje važnih i nevažnih dijelova prikazanog materijala raščlambom podataka, izvođenje dokaza i zaključaka potkrijepljeni generalizacijom. Obraćanje pažnje na uzorak te prepoznavanje skrivenog značenja kao i razlikovanje činjenica i zaključaka;
5. Stvaranje – ova razina uključuje primjenu znanja s naglaskom na inovaciju ili stvaranje unaprjeđenja za nešto što postoji, kreativna ili divergentna upotreba postojećeg znanja za stvaranje nove cjeline kombinirajući poznatih dijelova u novu cjelinu a samim time i stvaranje novih ideja i rješenja;
6. Prosudba – ova razina uključuje donošenje odluka na temelju naučenog putem prosuđivanja, predviđanja, ocjenjivanja, vrednovanjem, procjenom i preporukom na temelju poznatog standarda;

## ***2.2 Revidirana Bloomova taksonomija***

Nakon što je gotovo pola stoljeća korištena originalna Bloomova taksonomija znanja koja je bila znanstveno provjeravana, korištena i dorađivana, 2001. godine revidirana je od strane Andersona i Krathwohla. Naime, izvršena je zamjena hijerarhije dviju najviših razina znanja kod kojih stvaranje preuzima vrh tablice kao najveći stupanj dok je evaluacija pomaknuta korak niže (Krathwohl, D.R., 2002). Hijerarhija je slijedeća :

1. Dosjećanje – dosjetiti se, prepoznati ili reproducirati informaciju u približno onakvom obliku u kojem su naučeni

2. Shvaćanje – uočiti i povezati glavne ideje, razumjeti, objasniti ili interpretirati naučeni sadržaj opisivanjem tijeka događaja ili procesa te izvedbom logičkog zaključka iz dostupnih informacija
3. Primjena – rješavati probleme primjenom naučenog u kontekstu učenja ili u novoj situaciji na rutinski ili na nov način putem korištenja apstrakcije kao i odabirom i primjenom podataka i principa za rješavanje problema ili zadatka u drugom području
4. Analiza – razlikovati važne od nevažnih dijelova prezentiranog materijala putem raščlambe informacija kako bi se utvrdili dijelovi cjeline, njihovi međusobni odnosi te razlikovale činjenice i zaključci
5. Evaluacija – usporediti i pronaći sličnosti i razlike među idejama kao i procijeniti valjanost ideja ili kvalitete uratka na temelju poznatih kriterija
6. Stvaranje – kreativno ili divergentno koristiti postojeće znanje za stvaranje nove cjeline kombiniranjem poznatih dijelova u nove cjeline, stvaranje novih ideja i rješenja, izvođenje generalizacije na temelju dobivenih podataka te povezivanje znanja iz različitih područja kao i uočavanje novih obrazaca

Prethodno je navedeno kako je Bloomova taksonomija temelj za pripremu nastave, odabir sadržaja, metoda i strategija kao i određivanje ishoda, ciljeva učenja. Kako bi se ti ishodi mogli provjeriti, tj. kako bi se mogla vrednovati ostvarenost tih ishoda moramo imati sustav za provjeru znanja, bio to klasični ili elektronski. U nastavku rada će se detaljnije obraditi koncepti i sustavi za elektroničko testiranje znanja kao i problemi koji ih prate.

### 3. Koncepti i sustavi za elektroničko testiranje znanja

Vrednovanje postignuća, što uključuje i ocjenjivanje, je stalan proces usmjeren na razumijevanje i poboljšanje učenja učenika. Uključuje eksplicitna i javna očekivanja, postavljanje odgovarajućih kriterija i visokih standarda za kvalitetno učenje, sustavno prikupljanje, analiziranje i tumačenje dokaza koji određuju koliko se izvođenje ocjenjivanja podudara s očekivanjima i standardima te koristeći dobivene podatke dokumentirati, objasniti i s njima poboljšati izvođenje. Kada je učinkovito ugrađen u veće institucionalne sustave, ocjenjivanje nam može pomoći usredotočiti našu kolektivnu pozornost, ispitati naše pretpostavke i stvoriti zajedničku akademsku kulturu posvećenu osiguravanju i poboljšanju kvalitete visokog obrazovanja (Thomas, 1995).

Vrednovanje ostvarenosti ishoda učenja ima važnu ulogu u učinkovitosti, poboljšanju i održivosti procesa obrazovanja koja se sve više prepoznaje i zahtijeva. Vrednovanje je sastavni dio kriterija da obrazovna ustanova ispunjava potrebne standarde, kao i ključna sredstva pružanja dokaza potrebnih za traženje i održavanje akreditacije. Kellough i Kellough (Kellough, R. D. & Kellough, N. G., 1999.) su identificirali sedam svrha vrednovanja ostvarenosti ishoda od kojih se izdvajaju zadnje dvije jer za svrhu ovog rada nisu potrebne, a to su:

1. Poboljšati proces učenja učenika;
2. Prepoznati jake i slabe strane učenika;
3. Pregledati, procijeniti i poboljšati učinkovitost različitih strategija poučavanja;
4. Pregledati, procijeniti i poboljšati učinkovitost nastavnog plana i programa;
5. Poboljšati učinkovitost poučavanja.

Za razliku od klasičnog vrednovanja ostvarenosti ishoda putem papira i olovke, kod elektroničkog ispitnog sustava se koriste računala. Prije 40 godina računala su tek počela polako ulaziti u svakodnevnu upotrebu pa se elektronički ispitni sustav nije mogao nikako koristiti, dapače, tada je i počeo njegov razvoj. Još je krajem 1980-ih Cambridge Assessment istraživao mogućnost razvoja računalnih rješenja za ocjenjivanje strukovnih kvalifikacija. Test funkcija obrade teksta je uvedena zajedno s cijenjenim RSA ispitom za pisanje tajničkim vještinama i to je identificirano kao pogodno mjesto za razvoj elektroničkog ispitnog sustava

(Craven, P. 2009). Upotreba informacijske tehnologije u ocjenjivanju je definicija elektroničkog testiranja znanja. Taj način testiranja se vrši putem različitih ispitnih pitanja poput zadataka nadopunjavanja, višestrukog izbora i ili kratkih odgovora (e-Assesment). Sami tijek pripreme i provedbe e-testiranja najlakše se prati kroz takozvani životni ciklus pitanja (Purković i Kolumbić, 2012), a taj životni ciklus se sastoji od slijedećih postupaka:

1. Analiza ishoda učenja te priprema elektroničkog testa – analizira se kognitivna razina ishoda te se kreira pitanje uzimajući u obzir teorijsku i stručno-pedagošku osnovu;
2. Provedba elektroničkog testa – putem odgovarajućeg sučelja se provodi online putem u sigurnim uvjetima uključujući i nadzor;
3. Procjena rezultata elektroničkog testa – elektronički sustav koristi model apsolutne procjene znanja prema unaprijed određenim kriterijima;
4. Vrednovanje učinkovitosti elektroničkog testa – važno kako bi odredili vjerodostojnost, pouzdanost i fleksibilnost kao i uniformnost;
5. Korekcija elektroničkog testa – na temelju vrednovanja učinkovitosti obavljaju se modifikacije pitanja i ili kriterija vrednovanja rezultata.

Kako bi elektronički ispitni sustav bio općenito prihvaćen na odgovarajući način, iz perspektive učenika, nastavnika i institucija, prilikom razvoja koncepta mnogi zahtjevi se moraju imati na umu koji se mogu podijeliti na (Purković i Kolumbić, 2012):

- sigurnost pri provedbi postupka testiranja – sigurnost sustava te nadzor ispitanika pri provedbi testiranja poput snimanja aktivnosti;
- učinkovitost – sustav bi trebao biti učinkovitiji i ekonomičniji za provedbu za razliku od klasične pismene provjere postignuća na papiru;
- usklađenost pitanja i ishoda učenja kao i postupaka testiranja s pravilnicima učilišta i zakonima države – sadržaj koji se ispituje treba pripadati određenoj razini postignuća koju želimo mjeriti sukladno već spomenutoj Bloomovoj taksonomiji;
- mogućnost integracije s postojećim LMS (Learning Management System) sustavima;



- jednostavnost, pouzdanost i dostupnost za korištenje od strane polaznika testa i nastavnika;

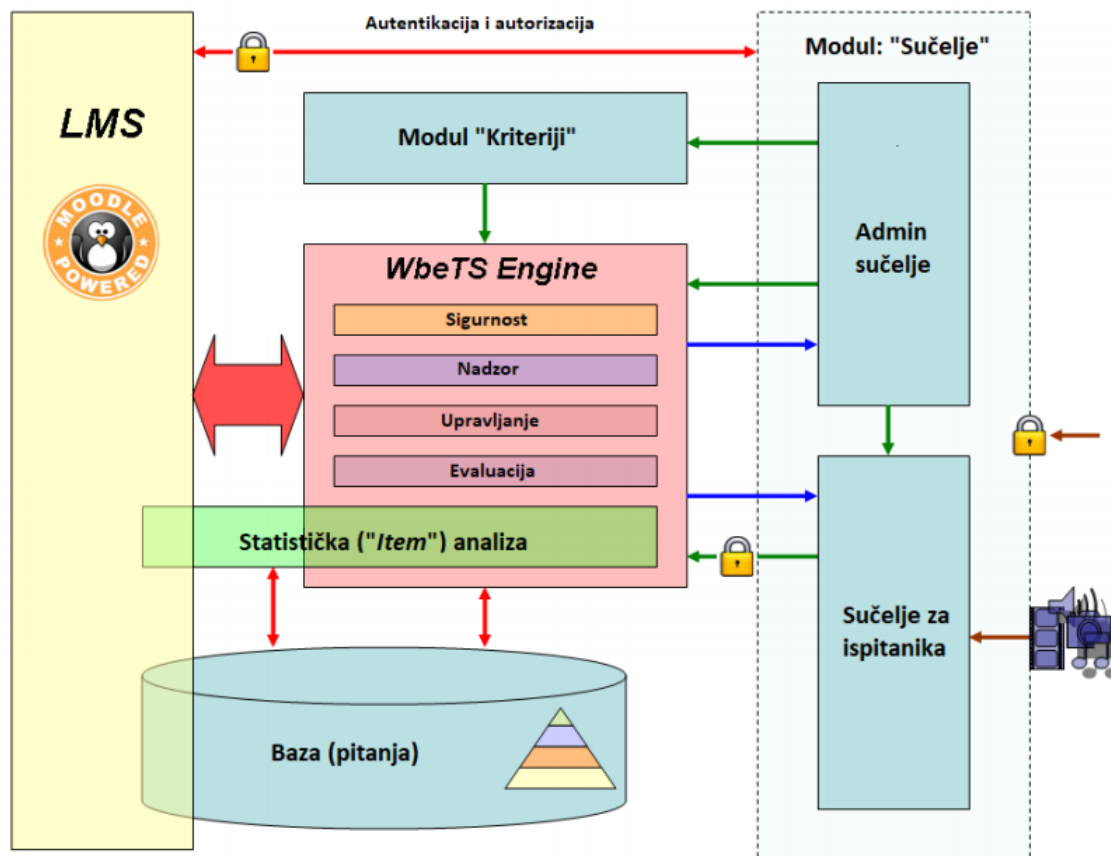
Trenutna praksa elektroničkog testiranja je u većini temeljena na utvrđenim postavkama i standardima za razvoj i realizaciju elektroničkog testiranja. Razni elektronički ispitni sustavi koriste se kao samostalne aplikacije ili se distribuiraju u raznim LMS sustavima. Postoji mnogo elektroničkih ispitnih sustava koji imaju različite prednosti, ali donose i određene nedostatke što ćemo u nastavku rada detaljnije proučiti.

### **3.1 Koncept WbeTS sustava**

Koncept WbeTS sustava e-testiranja znanja se zasniva na hijerarhijskim podjelama, tj. specifičan je po tome što su zadaci poredani na temelju težine ili složenosti, od najjednostavnijih do najsloženijih, što znači da svi polaznici započinju test identičnim pitanjima, a slijedeće pitanje ovisi o rješenju prethodnog pitanja, što zapravo ovisi o kvaliteti odgovora i vremenu potrebnom za rješavanje jednog od ponuđenih pitanja. Korištenje WbeTS sustava nudi tri velike mogućnosti (Purković i Kolumbić, 2012) :

- a) mogućnost vrednovanja veće skupine ispitanika bez potrebe za nazočnošću više nastavnika ili suradnika
- b) omogućuje ispitaniku veći vremenski prozor za rješavanje te omogućuje udaljeno pisanje testa
- c) predlaže jedinstvenu metodu vrednovanja postignuća nakon realizacije testiranja znanja

Jako visoka iskoristivost široko dostupne suvremene informacijsko-komunikacijske tehnologije također ima vrlo važnu ulogu samim time što pruža Internet i web-tehnologiju u jednom paketu. Kombinacija korištenja audio i video komponente se dolazi do prvih glavnih stavki ovog sustava. Audio komponenta se odnosi na iznošenje odgovora na višim razinama postignuća, dok video komponenta služi u svrhu djelomičnog nadzora postupka testiranja. Kako bi lakše opisali sam koncept na slici 1 je naslikana shema sustava koju ćemo u nastavku analizirati.



Slika 1. Shematski prikaz WbeTS sustava (Purković i Kolumbić, 2012)

Na slici 1. na lijevoj strani možemo uočiti LMS (Learning Management System) sustav koji je zapravo integrativni dio CMS (Content management system) sustava, pa samim time dijele neke zajedničke stavke poput autentikacije, autorizacije te djelomično i banke pitanja. Postojeći LMS sustav Moodle je odabran s razlogom, jer je koncept WbeTS zamišljen kao podsustav koji bi se integrirao u Moodle. Moodle se prvi puta pojavio 2002.godine te razvija se i danas a specifično je to što je sustav otvorenog koda. Banka pitanja, koja je se na slici nalazi u podsustavu Baza (pitanja) je zamišljena tako da se primjereno doradi i nadopuni s pitanjima te poveže s modulom Statička („Item“) analiza (Purković i Kolumbić, 2012), koji je nadogradnja na Item analizu koja je ujedno i sastavni dio Moodle-a. Item analiza Moodle sustava daje mogućnost pregleda izvještaja o odgovorima na pitanja u tabličnom obliku u kojem imamo uvid u pojedinosti svakog pojedinačnog pitanja. Mogu se analizirati odgovori svakog pojedinačnog ispitanika grupe, također se može analizira i kako su bolji učenici odgovorili na pitanje u direktnoj usporedbi s onim učenicima koji su imali lošije rezultate. Item analiza mjeri broj pokušaja i odabira ispravnog odgovora, ukupnu težinu pitanja, bodove koje je nastavnik dodijelio za odgovor te standardna odstupanja, itd.

### ***3.1.1 Modul Baza***

Ustanovljeno je kako je banka pitanja ujedno i baza pitanja a možemo reći da je to jedna baza podataka. Uz naziv modula možemo uočiti trokut koji podijeljen na razine koji simbolizira hijerarhiju pitanja. Pitanja su podijeljena u 5 skupina. Te skupine se temelje na revidiranoj Bloomovoj taksonomiji obrazovnih ciljeva. Svaka od tih pet skupina karakterizira jednu od razina ishoda. Pitanja koja se nalaze u bazi podataka se isključivo trebaju bazirati na ishodima učenja nastavne jedinice ili jedne nastavne cjeline, temeljem čega se tim pitanjima može odrediti vrijednost jedne od onih pet skupina. Prve tri skupine pitanja se odnose na pitanja dosjećanja, reprodukcije, te jednostavne logičke povezanosti i primjene dok se zadaci četvrte i pete skupine svrstavaju u skupinu složenijih pitanja za čije je rješavanje potrebna složena implementacija, analiza, procjena te kreativnost (Purković i Kolumbić, 2012). Možemo zaključiti kako je za prve tri skupine pitanja dovoljno koristiti postojeću strukturu banke pitanja dok je za četvrtu i petu skupinu nužno omogućiti, kao način za rješavanje pitanja, mogućnost snimanja zvuka te pohrana zvučnog zapisa kao i dodavanje slika i drugih multimedijских zapisa.

### ***3.1.2 Modul Kriteriji***

Na shematskom prikazu na slici 1. možemo vidjeti da se modul kriteriji nalazi odmah ispod autentikacije i autorizacije. Evaluacijske kriterije karakterizira ocjena koja ovisi o broju dobivenih ukupnih bodova te pojedinačni bodovi za svako pitanje. U samom modulu nastavnik postavlja preduvjete prelaska sa pitanja niže skupine na višu skupinu kao i evaluacijske kriterije samih odgovora na njegova pitanja. Karakteristike pitanja se najčešće odnose na vrstu te broj pitanja za svaku od skupina te način na koji će se generirati studentu što može biti automatski ili po izboru nastavnika. Za kriterij su uzeti bodovi skupljeni u pojedinoj skupini te je postavljen broj bodova koji je potreban za prijelaz u višu skupinu pitanja, a uzima se u obzir i vrijeme dostupno za rješavanje svake pojedine skupine, kao i količina dopuštenih kretnji za vrijeme rješavanja ispita koje sustav prati putem web kamere tijekom testiranja.

### ***3.1.3 Modul Mehanizam***

Na shematskom prikazu na slici 1. možemo vidjeti da se sustav WbeTS Engine tj. mehanizam nalazi odmah ispod kriterija. Ovaj sustav je zapravo poslužiteljska aplikacija koja

savršeno uklapa rad svih sustava te ga upotpunjuje s glavnim LMS sustavom Moodle. Ovako sastavljen sustav omogućuje nastavniku kontrolu pristupa i nadzora učenika a i direktno je povezan sa bankom pitanja, tj. na bazu podataka. Sustav mehanizam ima i sigurnosnu funkciju, a to je da prilikom rješavanja ispita kontrolira prikaz ekrana te „forsiranje“ samog ispita. Zbog tehničke naravi ovog problema nije pametno prepustiti potpunu kontrolu računalu pa se ovaj problem rješava putem raznih aplikacija poput Safe Exam Browser. Aplikacija poput ove „forsira“ da se na ekranu prikazuje samo ispit te zaključava bilo kakve druge radnje do isteka vremena ispita ili se prekida unosom lozinke za prekid što automatski nastavniku daje do znanja da je učenik prekinuo ispitnu proceduru. Za izradu jezgre testnog WbeTS sustava korištena je široko dostupna tehnologija zasnovana na PHP skriptnom jeziku i JavaScript-u (Purković i Kolumbić, 2012).

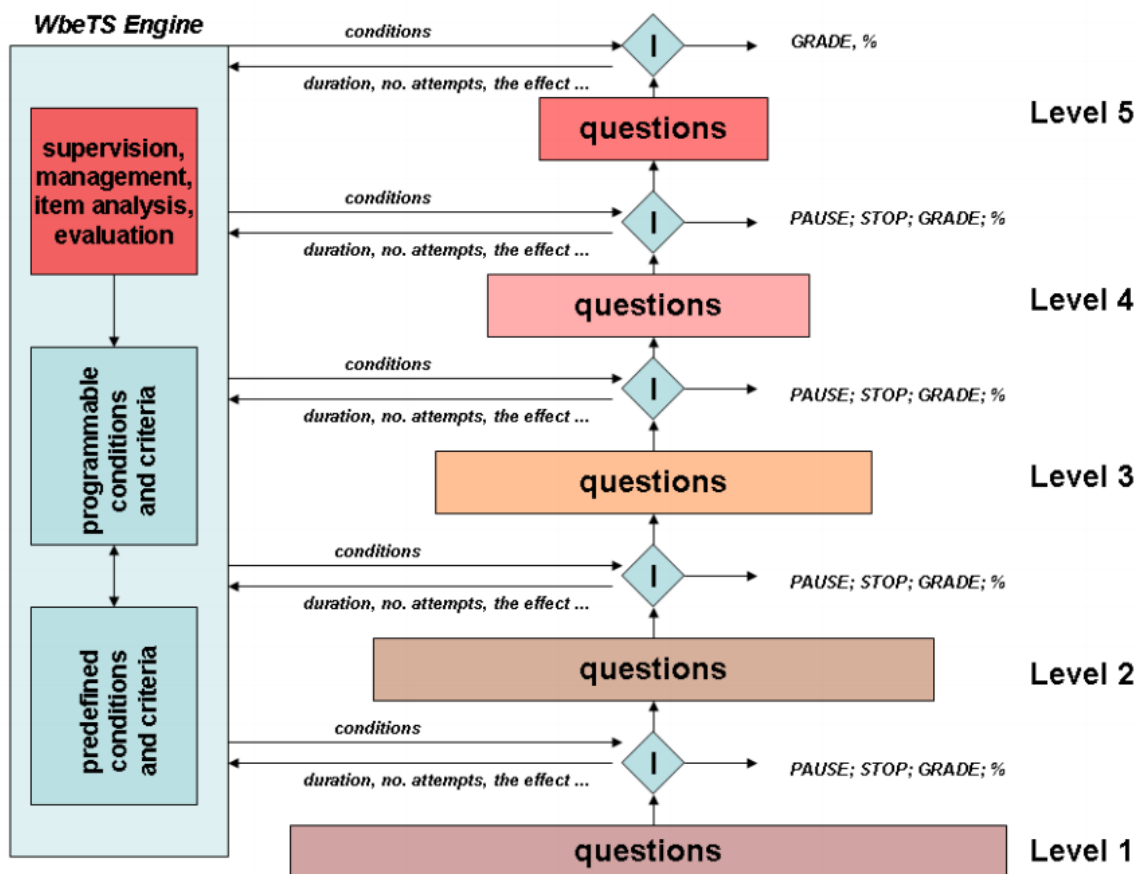
### ***3.1.3 Modul Sučelje***

Na shematskom prikazu na slici 2.1 možemo vidjeti da se modul Sučelje nalazi s desne strane te da se sastoji od nastavničkog ili tzv. administratorskog sučelja te studentskog sučelja. Samo sučelje predstavlja web-sučelje preko kojega nastavnik i ispitanici pristupaju sustavu e-testiranja. Administratorsko sučelje ima direktan pristup bazama podataka i kriterijima te nastavnik kroz njega kreira testove i prati sam tijek testa. Također može provesti i pokusni test, pokusnu evaluaciju, generirati izvješće, razgovarati s ispitanicima, vizualno preoblikovati sučelje itd. Studentsko sučelje je dizajnirano isključivo za polaznike testova. Ono omogućuje razgovor s nastavnikom, praćenje osobnih uspjeha te najvažnije polaganje, pisanje e-ispita. Za oblikovanje web-sučelja korišten je XHTML jezik čiji je izgled definiran uporabom CSS-a (Purković i Kolumbić, 2012).

### ***3.1.4 Postupak WbeTS e-testiranja***

Postupak e-testiranja sastoji se od ranije napisanih pet skupina koje ujedno predstavljaju i razine postignuća studenta. Svaka skupina predstavlja zasebnu jedinicu ili cjelinu. Nakon svake završene skupine student može ili završiti testiranje ili pauzirati ili nastaviti u drugom vremenskom terminu ako je prije samog ispitivanja tako dogovoreno sa nastavnikom. Ako student prilikom testiranja prekine testiranje bilježe se samo postignuti rezultati onih dovršenih skupina. Na slici 2. je prikazan dijagram procesa testiranja znanja u WbeTS sustavu u kojem se na lijevoj strani u crvenom kvadratu može vidjeti modul Mehanizam koji sadrži kontrolu i nadzor, item analizu te evaluaciju kao glavne stavke.

Također se ispod toga mogu vidjeti uvjeti i kriteriji koji su ili unaprijed definirani ili se modificiraju po potrebi nastavnika. Naposljetku se može vidjeti kako u sredini ima pet skupina pitanja (*eng. questions*) koje imaju svoje uvjete poput trajanja pisanja odgovora, broja pokušaja pisanja te učinak i slično. Na desnoj strani se može vidjeti razine skupina od 1 do 5 kao i proces koji poslije svake skupine pauzira i zaustavlja test te ga vrednuje.



Slika 2. Proces testiranja u WbeTS sustavu (Purković i Kolumbić, 2012)

Prethodno je istaknuto kako je u sustavu predviđeno pet skupina – razina zadataka, što zahtijeva dodatna pojašnjenja. U daljnjem tekstu iznosi se kraći opis svake od tih pet skupina.

### 1. Skupina

U prvoj skupini pitanja i zadaci se odnose na znanja na razini dosjećanja ili prepoznavanja, posjeduje veliki broj jednostavnih pitanja višestrukog izbora, nadopunjavanja i određivanja istinitosti tvrdnje a procjenjuju se isključivo činjenična znanja koja se odnose na terminologiju i specifične detalje gradiva. Vrijeme potrebno za rješavanje ove razine je

neograničeno i bez dodatnog nadzora. Ocjena za ovu skupinu je FX, što znači da je student spreman na testiranje za pozitivnu ocjenu.

## 2. Skupina

U drugoj skupini je već manji broj pitanja ali još uvijek dovoljno velik a pitanja i zadaci su tipa prepoznavanja i razumijevanja konceptualnog znanja. Posjeduje velik broj jednostavnih zadataka višestrukog izbora, nadopunjavanja ali i zadataka sređivanja i kategoriziranja. Provjerava se znanje o kategorizacijama, klasifikacijama, načelima i teorijama. Vrijeme i nadzor ne igraju veliku ulogu ali se mogu ograničiti a ovisno o uspješnosti, broju pokušaja i vremenu u kojem se riješe zadaci, zadovolji se prag za ocjenu dovoljan (2) tj. ocjena E ili D.

## 3. Skupina

U trećoj skupini je naglasak na primjeni znanja koje se odnosi na činjenično i praktično poznavanje različitih procedura, algoritama ili metodologije što znači da su zadaci složeniji a tipa su višestrukog izbora (*multiple response questions*), složenijih zadataka sređivanja, složenijih zadataka nadopunjavanja te mogu biti i numeričkog tipa. Trebaju sadržavati minimalni dodatni video (i audio) nadzor testiranja koji će detektirati samo osnovne (grube) anomalije pokreta radi sprječavanja trivijalnih metoda varanja na testu. Najčešće anomalije su razgovor, odlazak od računala na neprihvatljivo vrijeme, čitanje literature ili korištenje drugog računala (dulji gubitak fokusa) a vrijeme je ograničeno na mjeru koja bi trebala prosječnom i pripremljenom studentu jamčiti uspješno rješavanje. Za ovu skupinu zadataka nužno je provesti dodatno eksperimentalno testiranje a ocjena je dobar (3) tj. ocjena C.

## 4. Skupina

U četvrtoj skupini vrijeme postaje značajni ograničavajući čimbenik, što znači da mora biti manji broj pitanja koja su pretežno usmjerena na analizu (i sintezu) složenih materijala i sadržaja kojima je predočena neka pojava, struktura sustava, stanje, proces, tehnologija, incident. Pitanja su stoga formirana tako da od studenta traže diferencijaciju, organiziranje i određivanje karakteristika elemenata predmeta analize. Ovo su složena pitanja višestrukog izbora, nadopunjavanja, sređivanja, numeričkog i tekstualnog unosa, ali i složeni esejski tipovi na koje se odgovori osim klasičnim tekstualnim putem mogu unijeti i govorom. Očekuju se kratki i jasni odgovori čiji je obujam teksta i govora ograničen. U ovoj skupini odgovore ocjenjuje nastavnik, a audio i video nadzor prati iste anomalije kao na prethodnoj

razini ali se tolerira znatno manja učestalost tih anomalija što, u konačnici, znači stroži nadzor. Ocjena za ovu skupinu zadataka je vrlo dobar (4) tj. ocjena B.

### *5. Skupina*

U petoj skupini procjenjuje se najviša razina kognitivnih postignuća manjim brojem složenih modifikacija pitanja višestrukog izbora, nadopunjavanja i numeričkog unosa, te složenim pitanjima esejskog tipa, pitanja unosa programskog koda, multimedijalnog sadržaja, te govornog unosa rješenja. Rješavanje zadataka je relativno kratko i namijenjen je samo za najposposobnije studente. Video nadzor dopušta minimalnu razinu anomalija pokreta i zvuka a kako je malen broj studenata koji dostignu u ovu skupinu, pa nije potrebna automatizacija odgovaranja. Ocjena za ovu skupinu zadataka je izvrstan (5) tj. ocjena A.

E-testiranje kao postupak je zamišljen tako da se izvodi kao online testiranje s matične lokacije na kojoj se student nalazi, odnosno, na kojoj ima nužno minimalne uvjete za provedbu testiranja. Minimalni uvjeti koji su potrebni za provedbu e-testiranja:

- osobno računalo s nekim web-preglednikom prihvatljive verzije,
- širokopojasni pristup Internetu,
- web-kamera s mikrofonom.

Mehanizam sustava je zamišljen da automatski pomoću specijalno dizajniranog algoritma pratio anomalije a ujedno bi služio kao audio i video nadzor te bi povratnu informaciju o sumnjivim aktivnostima studenta kao i rezultatima te slao nastavniku koji onda reagira sukladno informaciji. Nastavnik vrlo često prihvaća onaj rezultat koji je automatizirano generiran na prve tri skupine ili, po pregledu tekstualnih odgovora i preslušavanju usmenih odgovora, određuje koju je od dvije najviše skupine student dostigao te položio.

### ***3.1.5 Problemi pri implementaciji WbeTS sustava***

Kako je ovaj sustav za sada samo koncept, nisu poznati apsolutno svi problemi s kojima će se budući nastavnici susresti ali kao prepreka pri uspješnoj implementaciji znamo da stoje slijedeće stavke (Purković i Kolumbić, 2012):

- izrada sustava za udaljeni nadzor ispitanika – trenutno su mnogi sustavi u izradi ali potrebno je još puno eksperimenata provesti kako bi rezultati bili valjani te kako bismo se sa velikom sigurnosti mogli osloniti na potpuno automatsku provedbu testa;
- izrada novih stavki tj. pitanja za više skupine postignuća, potrebno je osmisliti nove načine provjere;
- izrada sustava interpretacije item analize, tj. izrada sustava za poboljšanu analizu ispitnih pitanja te kao i kod sustava udaljenog nadzora potrebno je puno eksperimentiranja.

### 3.2 Koncept AHyCo sustava

Sustav AHyCo razvio se od vrlo jednostavnog alata za provjeru znanja pomoću pitanja višestrukog izbora do kompleksnog sustava za automatsko ocjenjivanje programskih zadataka. Svi nastavni sadržaji u sustavu AHyCo su podijeljeni na lekcije ili cjeline koje su grupirane u module. Kombinacijom više modula dobivamo predmet učenja. Znanje usvojeno kroz modul za učenje provjerava se putem raznih vrsta pitanja, potpitanja i raznih provjera znanja. Prije provjere znanja navodi se točan broj pitanja kao i točne cjeline čija će pripadajuća pitanja provjera znanja sadržavati. Skup konfiguracijskih pravila diktiraju koliko će najmanje, a koliko najviše pitanja iz svake jedinice biti uključeno u provjeru (Hoić-Božić, Budišćak, Botički, 2008). Upravo zbog tih konfiguracijskih pravila nije potrebno navoditi specifična pitanja jer će se prilikom svakoga novog pisanja odabirati novi skup pitanja. Unatoč tome se i odgovori na pitanja potpuno mijenjaju, čime se dodatno otežava prepisivanje, čak i kada se odgovara na dva ista pitanja. Pitanja u provjeri znanja mogu biti (Hoić-Božić, Budišćak, Botički, 2008):

1. Klasično pitanje tipa višestrukog odabira s jednim ili više točnih odgovora. Razlikuju se obična hipermedijska pitanja i parametarski zadana pitanja
2. SQL pitanje čiji je odgovor SQL upit, a evaluacija se provodi automatski putem modula za automatsko ocjenjivanje SQL zadataka

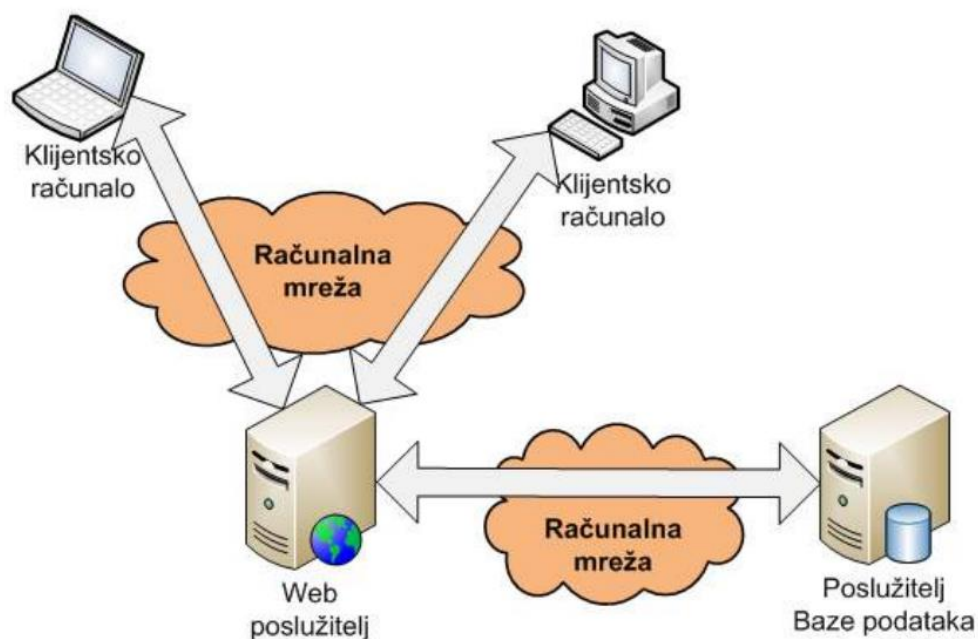


3. Programsko pitanje čiji je odgovor program pisan u programskom jeziku C++ ili C, a evaluacija se automatski provodi pomoću modula za automatsko ocjenjivanje programskih zadataka

AHyCo sustav je poseban po tome što ima mogućnost automatskog ocjenjivanja programskih zadataka te podržava nekoliko programskih jezika (SQL, C++, C#, C), a moguće je tu programsku potporu i proširiti tako da se doda par novih programskih jezika a to se postiže na način da se uvedu vanjski kompajleri. Sustav funkcionira na način da student napiše računalni program te izvorni kod računalnog programa učita u računalo koji zatim sustav sam ocjenjuje automatski tako što ga pokrene i uspoređuje izlaz izvršenog programa s unaprijed napisanim i definiranim testnim slučajevima tj. sa konačnim rješenjem zadatka.

### 3.2.1 Arhitektura AHyCo sustava

Sastavni dio sustava za učenje je upravo sustav provjere znanja. Korištenjem web-preglednika preko interneta ili intraneta, dolazimo u kontakt s AHyCo koji je zapravo web-aplikacija i implementirana je u ASP.NET tehnologiju. Sustav se sastoji od web-poslužitelja, poslužitelja baze podataka te klijentskih računala putem kojih studenti pristupaju web-aplikaciji AHyCo-a, kao što je prikazano na slici 3. dolje.



Slika 3. Arhitektura AHyCo sustava (Hoić-Božić, Budišćak, Botički, 2008)

Sustav za provjeru znanja implementiran je kao troslojna arhitektura sa sljedećim slojevima te je prikazan na slici 4. dolje (Hoić-Božić, Budišćak, Botički, 2008):

- Prezencijski sloj
- Sloj korisničkog sučelja
- Sloj poslovne logike
- Sloj pristupa bazi podataka



Slika 4. Prikaz troslojne arhitekture sustava za provjeravanje znanja (Hoić-Božić, Budišćak, Botički, 2008)

Tehnologija ADO.NET je korištena kod ostvarenja sloja pristupa bazi podataka, što je omogućilo sasvim jednostavnu integraciju sa SQL Serverom koji se koristi kao glavni sustav za upravljanje bazom podataka. Zamišljeno je da sloj pristupa bazi podataka uopće ne ovisi o

višim slojevima, tipa o sloju poslovne logike koji se nalazi iznad njega. Metode za pohranu, izmjenu i brisanje podataka te dohvaćanje podataka se nalaze u specijalnim razredima, tj. u bazi nalaze se u prostoru imena ProvjereLib.DataAccess te oni koriste isključivo razrede koje pruža već spomenuta tehnologija ADO.NET.

Model entiteta i odnosa tj. tzv. ER model podataka je korišten kao temeljni dio sloja poslovne logike. Razredi toga sloja nalaze se u prostoru pod nazivom ProvjereLib.BusinessObjects. Glavna zadaća sloja poslovne logike je ta da upravlja podacima na višoj, apstraktnoj razini u odnosu na spomenuti najniži sloj pristupa bazi podataka. Ocjenjivanje pitanja ili cijele provjere, kao i provjeravanje podataka prilikom njihova unosa je implementirano metodama ovog razrednog sloja.

Sloj korisničkog sučelja se zbog specifičnosti same web-aplikacije odvija se i na strani poslužitelja i na strani klijenta. Na strani poslužitelja razredi sloja korisničkog sučelja nalaze se u prostoru pod nazivom ProvjereLib.GUIObjects te sadrže samo one podatke koji se prezentiraju na strani klijenta u obliku obične HTML web-stranice.

### **3.2.2 Provjera znanja**

Komponente u internetskom obrazovanju mogu podijeliti u tri skupine koje se dalje mogu podijeliti u još tri manje skupine što možemo vidjeti u tablici 1. ispod (Brusilovsky i Miller, 1999):

Tablica 1. Podjela skupina komponenata internetskog obrazovanja

<b>1. Prije</b>	<b>2. Tijekom</b>	<b>3. Nakon</b>
Priprema: <ul style="list-style-type: none"> <li>• Autorizacija</li> <li>• Spremanje</li> <li>• Odabiranje</li> </ul>	Dostava <ul style="list-style-type: none"> <li>• Predstavljanje pitanja</li> <li>• Interakcija</li> <li>• Dobivanje odgovora</li> </ul>	Procjena <ul style="list-style-type: none"> <li>• Evaluacija</li> <li>• Ocjenjivanje i evidentiranje</li> <li>• Slanje povratne informacije</li> </ul>

### ***3.2.3 Prva skupina (prije početka rada)***

Prije početka rada vrši se selekcija pitanja koja čine provjeru znanja. AHyCo sustav sadržava zgodan modul za unos bilo kakve vrste pitanja kojem pristupamo preko web-sučelja. Taj modul nam omogućuje i izmjene teksta pitanja i odgovora ukoliko pitanje ima već standardne odgovore. Vrlo bitna stavka je mogućnost unošenja složenija tipa teksta te ujedno i sami uređivač teksta mora omogućiti unos slika, tablični prikaz te formatiranje teksta i sl., dok nam je .NET tehnologija to omogućila pomoću implementacije kontrole koristeći Microsoft Word kao uređivača teksta te je ujedno i omogućen i unos teksta pomoću HTML uređivača teksta.

Način unosa programskih pitanja se dijeli na dva načina: unos testova za pitanje i unos teksta pitanja s točnim rješenjem. Nakon unosa programskih pitanja slijedi unos točnih rješenja programskih zadataka koji možemo podijeliti na tri načina: prefiks, točni odgovor i sufiks. Ti se načini zatim unose u odgovarajuća polja u sučelju web-stranice te na slici 5. to možemo i vidjeti.

Polje Točan odgovor treba sadržavati točno rješenje zadatka, a Sufiks polje se koristi u zadacima koji traži implementaciju funkcije. Definiranje i uvid u obavljene provjere znanja se obavljaju pomoću modula za provjere, a unos provjere znanja se dijeli na dva dijela: 1- definiranje samih parametara provjere znanja i 2 - odabir sastava provjere znanja. Provjera znanja se može sastojati od jedne ili više cjelina kao što smo već spomenuli pri čemu su u istoj pitanja svake od onih odabranih cjelina. Kad smo unesli sva pitanja i njihove korespondirajuće odgovore, provjera znanja se može pokrenuti.

- > Povratak u glavni izbornik
- > Povratak na pregled pitanja po cjelinama
- > Unos novog pitanja

Parametri predmeta	
Akademaska godina	2007./2008. ljetni semestar
Predmet	Algoritmi i strukture podataka (FER 2)
Modul	ASP - I Ciklus

Parametri pitanja	
Pitanje	5139
Programski jezik	Cplusplus
Aktivno pitanje	<input checked="" type="checkbox"/> Pitanje se koristi u provjerama
Cjeline	Rbr Cjelina Brisanje
<a href="#">Dodavanje cjeline</a>   1 ASP - AOP - Datoteke   <a href="#">Brisanje</a>	

- > Pehrana izmjena
- > Testiranje izvodenja

> Unos/izmjena tekata pitanja

**Tekst pitanja**

Neka je zadana slijedna neformatirana datoteka u kojoj se nalaze zapisi oblika:

```
struct album{
    char naziv[30+1];
    char autor[20+1];
    int prodano;
};
```

Napisati funkciju koja će za zadanog autora vratiti broj prodanih primjeraka albuma (prodano).  
Prototip funkcije mora biti:

```
int br_prodenih(FILE *f, char *autor);
```

**Prefiks**

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

struct album
{
    char naziv[30+1];
    char autor[20+1];
    int prodano;
};

void prebaci(FILE *f, FILE *n) /* iz formatirane u neformatiranu */
```

**Točan odgovor**

```
int br_prodenih(FILE *f, char *autor)
{
    struct album a;
    int br = 0;
    while (fread (&a, sizeof(a), 1, f))
    {
        if (strcmp(a.autor,autor) == 0) br = br + a.prodano;
    }
    return br;
}
```

**Sufika**

```
int main()
{
    FILE *f, *n;
    char autor[20+1];
    f=fopen
    ("C:\\inetpub\\wwwroot\\aHyCo\\AOPDatoteke\\IN005_albumi_f.txt","r");
    n=fopen("neformatirana.txt","wb");

    if (f == NULL || n == NULL) {
        printf("pogreška prilikom otvaranja datoteka");
    }
    else
    {
        prebaci (f,n);
    }
    fclose(f);
    fclose(n);
}
```

Slika 5. Web sučelje sustava za unos AOP programskog pitanja (Hoić-Božić, Budišćak, Botički, 2008).

### ***3.2.4 Druga skupina (tijekom rada)***

Tijekom rada, tj. nakon što su studenti kliknuli na početak pisanja provjere znanja potrebno je studentu prikazati pitanje a to ćemo učiniti pomoću sučelja u kojem će studenti ujedno i odgovarati na pitanja te u pozadini toga moramo osigurati prihvrat tih odgovora kao i samo vrednovanje istih, a tu nam dolazi okolina za učenje poput Moodle-a, temeljena na webu ukomponirana sa našim sustavom AHyCO. Pisanje provjere znanja se događa u web-obrascu koji može prikazivati bilo koju vrstu pitanja a samim time prilikom promjene istih se mijenja i funkcionalnost te prikaz web-stranice. Pritiskom na gumb „Ocjenjivanje provjere“ student u bilo koje vrijeme ima mogućnost predaje provjere znanja. Netom prije predaje, studentu se pokažu svi odgovori koje je upisao te ima mogućnost nastavka pisanja ili završne predaje provjere nakon koje više nema nastavka pisanja.

### ***3.2.5 Treća skupina (nakon rada)***

Kada student završi sa pisanje provjere znanja, u web sučelju odabire opcije predaj test te ga sustav obavezno pita jeli siguran da je gotov kako bi se smanjila mogućnost slučajno-pogrešnog odabira. Time se smatra kako je provjera znanja gotova i nakon toga student više nema pravo izmjene ili nadopunjavanja svojih odgovora. Po završetku provjere znanja, sustav boduje, tj. ocjenjuje odgovore na temelju točnosti sa netočan, djelomično točan i potpuno točan, tj. ako bi pridodali brojčanu vrijednost to bi bilo 0, 0.5 ili 1 bod. Sustav nakon bodovanja odgovara daje studentima važnu povratnu informaciju koja sadrži bodove svih pitanja, popis točnih odgovora i zaključavanje te spremanje i evidentiranje testa. Na slici 6. se vidi završni izgled sučelja u kojem se nalaze podaci povratne informacije jednog od programskih pitanja.

Pitanje

1 2 3

b

b c

Mogući broj bodova: 1,00  
Ostvareni broj bodova: 1,00

Napišite funkciju koja kvadrira zadani cijeli broj.

Prototip funkcije je:

int Kvadrat(int a);

Sintaksna točnost: 😊

Rezultat ocjenjivanja: 😊

Ulaz	Očekivani izlaz	Dobiveni izlaz	Rezultat ocjenjivanja
4	16	16	😊

Slika 6. Pregled rezultata programskog testa (Hoić-Božić, Budišćak, Botički, 2008)

Studentski pregled provjere i nastavnikov se razlikuju u većoj mjeri, jer postoji mogućnost da nastavnik mora ručno pregledati neka od pitanja te ih ručno i ocijeniti, također ako je došlo do tehničkih poteškoća prilikom pisanja provjere znanja, nastavnik naknadno može bodovati odgovore koji su bili nepotpuni ili se radilo možda o sitnoj gramatičkoj grešci. Potrebno je ukratko objasniti način bodovanja programskih zadataka. U modulu se spaja kod koji upisan od strane studenta s malim dijelovima koda od strane nastavnika koji se unesao prilikom upisivanja pitanja i odgovora u prvoj skupini. Zatim se taj kod kompajlira i izvršava sa posebno pripremljenim testnim primjerima. Rezultat vrednovanja izlazi u jednom od tri završna stanja:

- program se kompajlirao ili se nije uspio kompajlirati,
- program je djelomično prošao,
- program je potpuno prošao.

Studentu se dodjeljuju završni bodovi prema formuli:

$$AR = \frac{AP}{\sum_i TP_i} \sum_i TR_i * TP_i$$

U kojoj su korištene slijedeće oznake u formuli:

- AR – rezultat vrednovanja zadatka
- AP – maksimalni broj bodova za zadatak
- $TP_i$  – maksimalni broj bodova za testni slučaj ( $0 < i < n$ , gdje  $n$  predstavlja ukupan broj testnih slučajeva za određeni zadatak)
- $TR_i$  – bodovi dodijeljeni za testni slučaj  $i$

Pri sadašnjoj implementaciji različita stanja iznose različite postotke od maksimalnih bodova pridruženih testnom slučaju (Hoić-Božić, Budišćak, Botički, 2008):

- program se nije preveo ili testiranje nije uspjelo (0%),
- testiranje programa je djelomično uspjelo (50%),
- testiranje programa je u potpunosti uspješno (100%).

### **3.3 Koncept fuzzy logike**

Umjetna inteligencija opisuje područje računalne znanosti koje se bavi razvojem inteligentnih alata bilo to strojeva, aparata ili aplikacija koje reagiraju na razne događaje. Ultimativni cilj umjetne inteligencije je stvaranje algoritama koji bi oponašali ljudski način razmišljanja. Fuzzy logiku je teško predočiti nekim od dostupnih matematičkih metoda jer predstavlja novi pristup problemima upravljanja složenih sustava. Neizraziti modeli, tj. skupovi su određeni kao osnova za prikaz U/I podataka sustava s neizrazitim zaključivanjem. Može se reći da fuzzy logika nije logika koja stvari čine nejasnima, nego logika kojom objašnjavamo nejasne stvari. Fuzzy logika je teorija nejasnoće, nejasnih skupova, skupova koji kalibriraju neodređenost, neku nejasnoću. Glavna ideja je da se stvarima koje se želi izmjeriti pristupa stupnjevanjem koje je zasnovano na ljestvicama - temperatura, visina, brzina i sl.

#### ***3.3.1 Usporedba s konvencionalnom logikom***

Konvencionalna logika koristi samo dvije vrijednosti a to su istina i laž tj. ako pridodamo brojevanu vrijednost možemo reći da koristi samo nulu i jedinicu. Za razliku od fuzzy logike kod koje je osnovna ideja da neki element pripada nejasnom setu s nekim



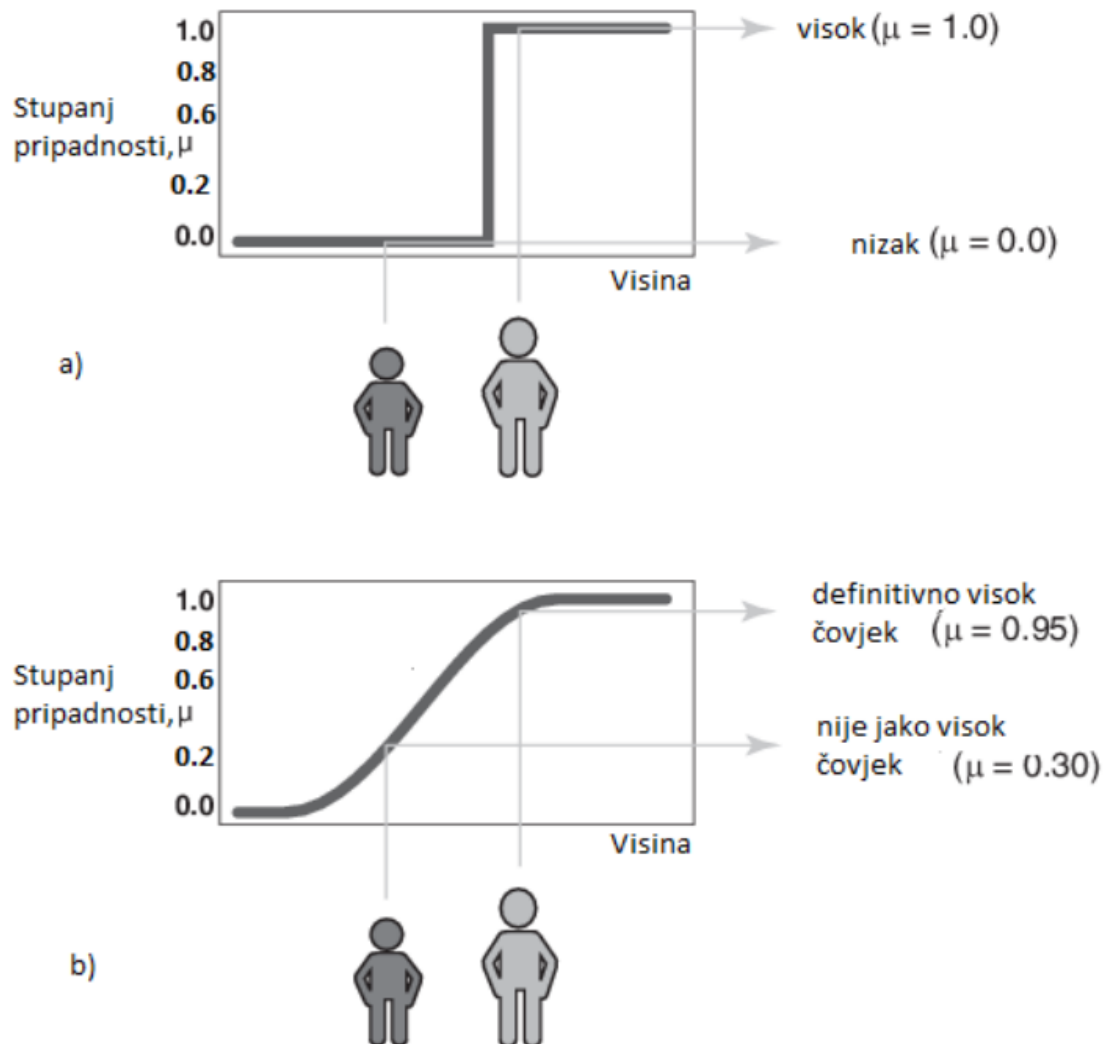
stupnjem članstva, konvencionalna logika ne može zastupati takve nejasne koncepte. Po tome možemo zaključiti kako svaka tvrdnja nije ni zasigurno točna a ni zasigurno netočna već je u nekom određenom stupnju točna ili netočna. Jednostavan primjer fuzzy logike ćemo predočiti u tablici 2. dolje.

Tablica 2. primjer stupnja pripadnosti ovisno o visini osobe

<i>Imena osoba</i>	<i>Visine osoba u centimetrima</i>	<i>Stupanj pripadnosti</i>	
		<i>Konvencionalna logika</i>	<i>Fuzzy logika</i>
Ivona	155	0	0.01
Petra	162	0	0.11
Ana	169	0	0.35
Matea	175	0	0.66
Ivano	182	0	0.75
Aleksandar	187	1	0.87
Matej	190	1	0.90
Mihael	202	1	1.00
Martino	206	1	1.00

Iz tablice se mogu uočiti dvije stavke, a to su visine Ivone i Martina, gdje je Ivoni sa njezinih 155 centimetara dodijeljena vrijednost 0 te Martinu sa njegovih 206 centimetara dodijeljena vrijednost 1. Ostalim osobama između najniže visine (155 centimetara) i najviše visine (206 centimetara) su dodijeljeni stupnjevi. Pitanje glasi: Tko je od ovih osoba visok ? U konvencionalnoj logici bi postavili granicu recimo na 185 centimetara te bi na temelju toga mogli zaključiti kako su sve osobe visinom iznad te granice visoke. Kod fuzzy logike to nije takav slučaj. Sada pitanje glasi: Koliko je osoba visoka? Odgovor bi bio u djelomičnom sklopu fuzzy skupa, recimo tiša Aleksandar je 0.82 visok. Na slici 7. možemo zaključiti kako fuzzy logika ima blagi prijelaz preko granice te kako nam fuzzy skup pod nazivom visok čovjek prikazuje vrijednost visine u korespondirajućoj vrijednosti. Možemo vidjeti također da

Ivano sa 182 centimetra, što je samo 3 centimetra manje od granice malena osoba po konvencionalnoj logici. Po tome su i duge osobe stupnjevane kao visoki ljudi sukladno njihovoj visini.



Slika 7. Razlika konvencionalne logike i fuzzy logike

### 3.4 Item response theory (teorija odgovora na zadatke)

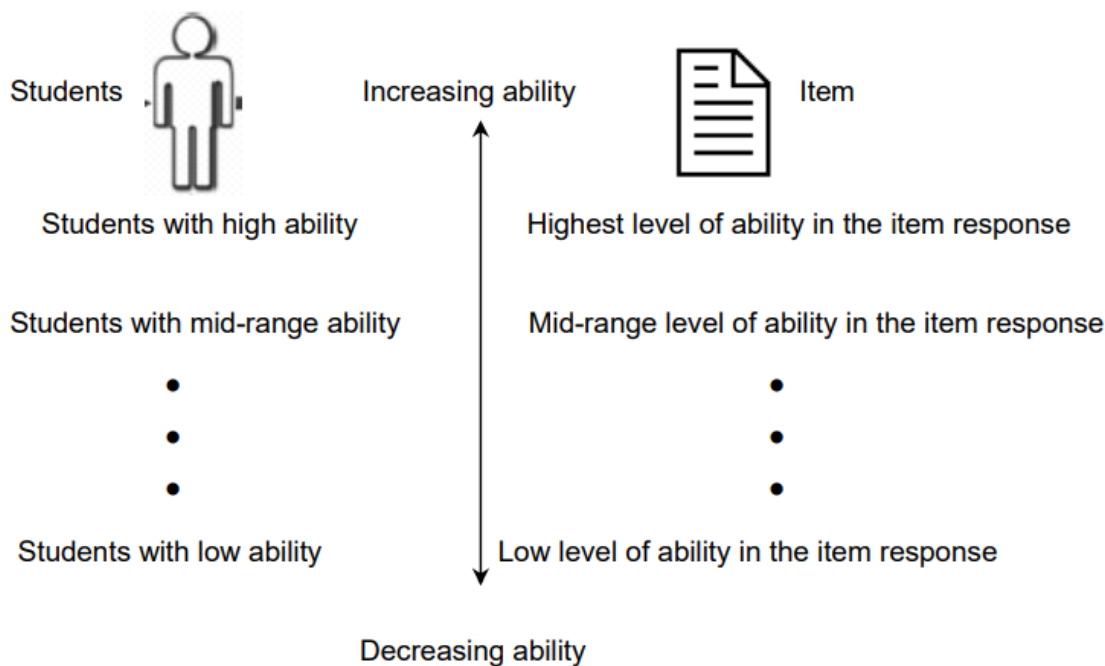
Jedno od rješenja u konstruiranju dobrog testa koji može precizno izmjeriti sposobnost je korištenje teorije odgovora na zadatke (IRT) u izradi testa. IRT modeli su matematički funkcije koje mogu specificirati vjerojatnost diskretnog ishoda, kao što je to točan odgovor na zadatak, u smislu osobe i parametra zadatka. Parametri osobe mogu

predstavljati sposobnost studenta ili snagu čovjekova stava. Zadaci mogu biti pitanja koja mogu imati netočne i točne odgovore ili izjave na ankete koji ispitanicima omogućuju da to naznače razinom dogovora. Prije nego što korisnik nastoji koristiti IRT, mora osigurati da je skup podataka ispunio pretpostavke IRT-a kako bi se osiguralo da je odabran ispravan model, u slučaju da nije, model je znatno siromašan i u konačnici upitan. Postoje četiri pretpostavke modela odziva predmeta, a to su (Desa i Latif, 2008):

- 1) dimenzionalnost latentnog prostora,
- 2) lokalna neovisnost,
- 3) karakteristične krivulje zadatka,
- 4) brzina testa koji promovira bolje razumijevanje IRT modela.

#### ***3.4.1 Dimenzionalnost latentnog prostora***

Dimenzionalnost modela odgovora na zadatke definira se na koliko je sposobnosti testirano ispitanika u testu. Sposobnost ispitanika naziva se latentnim osobinama ispitanika zbog svoje neprimjetne vrijednosti. Ako modeli odgovora na zadatke pretpostavljaju samo jednu sposobnost ili homogeni skup predmeta koji će se testirati, a zatim se dimenzije predmeta u testu nazivaju jednodimenzionalnima. Svakako, postoji puno čimbenika koji mogu utjecati na performanse ispitanika kao što su kognitivna razina, osobnost, razina ispitanikove motivacije, sposobnost brzog rada, znanje o pravilnoj upotrebi alata za odgovore itd. Ovi dominantni faktori nazivaju se sposobnostima mjerenim testom (Desa i Latif, 2008). Za svaki test, pretpostavku jednodimenzionalnosti treba provjeriti tako da primijenjeni test mjeri svaki put istu osobinu. Štoviše, ako je rad ispitanika testiran više od jedne latentne varijable, tada se IRT naziva višedimenzionalnim modelima (MIRT). Pretpostavlja se, da postoji  $k$  latentnih osobina koje definiraju  $k$ -dimenzionalni latentni prostor, a položaj ispitanika za svaku osobinu određuje se lokacijom u latentnom prostoru. Smještaj ispitanika i mjesta predmeta linearno su povezani u mjerenju izvedbe ispitanika što možemo vidjeti na slici 8. ispod. Na slici možemo vidjeti studente s lijeve strane koji su raspoređeni od studenata sa najmanjom do studenata sa najvećom sposobnosti, a posljedica linearne povezanosti da je na takav način i raspoređena stavka, od najniže do najviše sposobnosti u odgovoru na stavke.



Slika 8. Tok studentskih sposobnosti (Desa i Latif, 2008)

### 3.4.2 Karakteristična krivulja zadatka

Karakteristična krivulja zadatka (KKZ ili eng. ICC – item characteristic curve), pokazuju vezu između sredstva uvjetne vjerojatnosti s regresijom ocjene predmeta na sposobnost. Učestalost ispitnih rezultata ispitanika s fiksnom sposobnošću daje se sljedećom formulom :

$$f(x|\theta) = \sum_{\sum y_i = x} \prod_{i=1}^n P_i[\theta]^{y_i} Q_i[\theta]^{1-y_i}$$

gdje je  $x$  ispitni rezultat ispitanika te je  $x \in [0, n]$

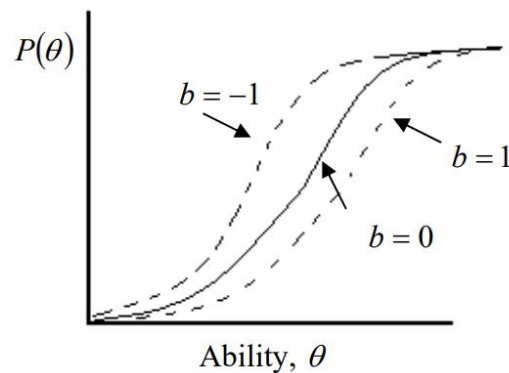
KKZ za stavku  $i$  je označena sa  $P_i(\theta)$  kao vjerojatnost da ispitanik odgovori točno na zadatak  $i$  sa sposobnošću  $\theta$ . Na primjer, 35% od svih ispitanika sa sposobnošću  $\theta = 1.5$  se očekuje da će točno odgovoriti na zadatak 1 s vjerojatnošću iznosi  $P_1(1.5) = 0.35$ .

Ne postoje fiksne matematičke funkcije KKZ-a. Ako bismo uzeli u obzir samo jednu latentnu sposobnost, onda se regresija naziva KKZ, inače, kao u višedimenzionalnim modelima, regresija se naziva funkcijom karakteristike zadatka (FKZ ili eng. IRF - Item Characteristic Function). Univerzalno, postoje tri matematička modela za KKZ koji ilustriraju povezanost KKZ-ove vjerojatnosti ispravnog odgovora sa sposobnosti. Svaki model predstavlja jedan ili

više parametara te standardni oblik modela KKZ-a je kumulativni oblik logističke funkcije . Prvi modeli poznati su kao 1-PL ili jedno-parametarski logistički modeli, a također poznati i kao Rasch modeli koji su dati iz slijedeće formule:

$$P_i(\theta) = \frac{1}{1 + e^{-(\theta - b_i)}}$$

u kojoj je  $e$  konstanta u iznosu od 2.718, a  $b_i$  je parametar težine zadatka  $i$ . Drugim riječima, možemo reći da udio zadataka na koje određeni ispitanik sa sposobnošću  $\theta$  može odgovoriti zadatak  $i$  točno je data formulom. Slika 2.8 prikazuje "S" u obliku modela 1-PL za tri različite razine težine. Što je veća vrijednost  $b_i$  to je veća vjerojatnost točnog odgovora. Tipične vrijednosti  $b_i$  se kreću u rasponu od -3 do +3. Na slici 9. dolje, možemo vidjeti kako izgleda tipična karakteristična krivulja zadataka za 1-PL mode sa tri stupnja težine.

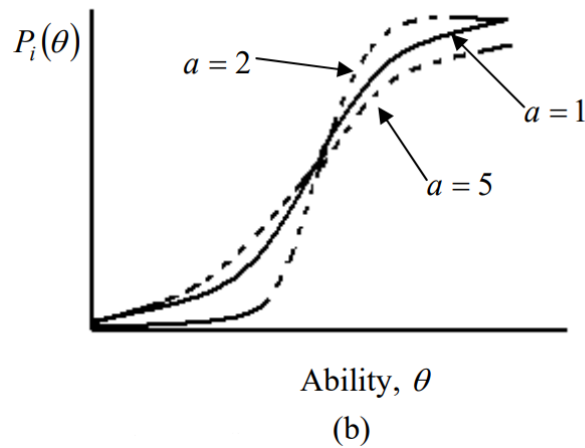


Slika 9. Tipična karakteristična krivulja zadataka za 1-PL mode sa tri stupnja težine (Desa i Latif, 2008)

U primjeni IRT 1-PL model ne odgovara uvijek dobro određenim podacima jer zadaci nisu uvijek paralelni, stoga postoje alternativni modeli koji se koriste kada se naiđe na taj problem. Možemo izbrisati zadatke na kojima krivulja prikazuje nagibe koji su divergentni ili možemo generalizirati model kako bi se omogućili različiti nagibi unutar krivulje . U dvoparametarskom logističkom modelu, KKZ se razlikuju i po nagibu i po težini (neki zadaci su teži od drugih). U ovom modelu postoji dodatni parametar za svaku pojedinačnu stavku. Ovaj parametar označen kao  $a$  i često se naziva diskriminacijom zadatka. Formula za 2-PL je slijedeća:

$$P_i(\theta) = \frac{1}{1 + e^{-a_i(\theta - b_i)}}$$

u kojoj je  $a$  razina diskriminacije zadatka  $i$ . Uobičajeni raspon razine diskriminacije  $a$  je u praski u rasponu od -2.80 do +2.80. Na slici 10. dolje, možemo vidjeti kako izgleda tipična karakteristična krivulja zadatka za 2-PL modele.

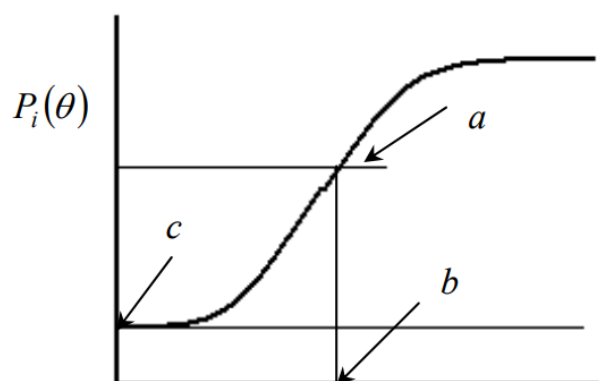


Slika 10. Tipična karakteristična krivulja zadatka za 2-PL modele (Desa i Latif, 2008)

Test opće sposobnosti, test postavljanja ili bilo koji tip testa s višestrukim izborom ostaje popularan i stoga činjenica je da će u ispitivanju ispitanici riješiti zadatak pogađanjem (Desa i Latif, 2008). Ni modeli 1-PL ni 2-PL nisu uzeli u obzir fenomen pogađanja. Moguće je da će ispitanici točno pogoditi odgovor na teži zadatak ili će odgovoriti koristeći vještinu ili znanje. Ova pojava se može ispraviti modificiranim 2-PL modelom tako da uključuje parametar koji predstavlja doprinos pogađanja vjerojatnosti ispravnog odgovora. Dodaje se treći parametar  $c$ , koji je binomni oblik parametra pogađanja. Ovaj modificirani model se naziva model s tri parametra ili model 3-PL te je prikazan u slijedećoj formuli :

$$P_i(\theta) = c_i + (1 - c_i) \frac{1}{1 + e^{-a_i(\theta - b_i)}}$$

u kojoj je  $c_i$  parametar pogađanja za zadatak  $i$  koji je također znan kao vjerojatnost ispravno riješenog zadatka  $i$  pogađajući. Raspon vrijednosti  $c_i$  je između nule i jedinice, no praktična vrijednost je između 0 i 0.35. Na slici 11. dolje, možemo vidjeti kako izgleda tipična karakteristična krivulja zadatka za 3-PL modele.



Slika 11. Tipična karakteristična krivulja zadatka za 3-PL modele (Desa i Latif, 2008)

### 3.4.3 Brzina testa

Nema puno korisnika IRT-a koji obraćaju pažnju na brzinu testa koja može utjecati na izvedbu ispitanika. Neuspjeh ispitanika da završe test u zadanom roku tako da ovaj faktor ne zagađuje procjene rezultata sposobnosti. Kada brzina postane jedan od čimbenika, tada u IRT modelima postoje najmanje dvije osobine koje utječu na uspjeh ispitanika, a to su latentne osobine i vremensko ograničenje. Brzina testa može se provjeriti utvrđivanjem broja ispitanika koji ne uspiju završiti set testova te brojevima zadataka koji nisu stigli ispuniti. Postoje tri čimbenika koje treba uzeti u obzir u odnosu na brzinu (Desa i Latif, 2008):

- proporcija testa nije dosegnuta 5%, 10% i 15%;
- odgovor na nedostignuto, prazno polje i slučajan odgovor;
- slaganje zadataka od slučajnog odabira do lakih i teških.

### 3.4.4 Primjena IRT-a

IRT je tijelo razvoja modernih psihometrijskih polja. Teorija i tehnika IRT-a za ispitivanje psihometrijskih svojstava mjera mnogo su složeniji od klasične metode, međutim, postoji mnogo prednosti kod primjene IRT-a protiv klasične teorije ispitivanja. IRT modeli su posebno važni i kao temelj za prilagodbu testiranja (uglavnom u računalno prilagodljivom testiranju, eng. *computerized adaptive testing* ili skraćeno CAT). Kad je pokrenut test putem računala, računalo može ažurirati odabir zadataka zbog trenutne izvedbe ispitanikove sposobnosti. Da bi se ispitne stavke uskladile s razinama sposobnosti, potreban je velik skup zadataka. S pravom bankom zadataka, tj., bazom podataka i velikom varijacijom sposobnosti

ispitanika, CAT može biti puno učinkovitiji od tradicionalnog testa papirom i olovkom (P&P). U adaptivnom testiranju strategija za odabir zadataka iz grupe predmeta slijede sljedeća pravila odluke:

- Ako ispitanik riješi zadatak ispravno, sljedeći bi zadatak trebao biti teži,
- Ako ispitanik netočno riješi zadatak, tada bi sljedeći bi zadatak trebao biti lakši.

Mnogo je prednosti primjene IRT-a u kompjuteriziranom prilagodljivom testiranju, kao što su (Desa i Latif, 2008):

- Kada su pretpostavke IRT modela zapravo ispunjene, IRT pruža odgovarajuće jači nalazi zbog procjene nepromjenjivih parametara, za razliku od toga, klasična teorija ispitivanja procjenjuje parametre poput težine predmeta, diskriminacije predmeta i pouzdanosti na određeni način. Rezultati pogreške se također pretpostavlja da su konstantni.
- IRT nudi nekoliko poboljšanja u skaliranju zadataka i ljudi
- Parametri IRT modela uglavnom ne ovise o uzorcima ili ispitivanjima. Dakle, IRT pruža znatno veću fleksibilnost u situacijama kada se koriste različiti se uzorci ili obrasci za ispitivanje.
- Stavke u adaptivnom testiranju IRT modela odabiru se prema trenutnoj sposobnosti ispitanika. Stoga se neobrađena ocjena temelji na ponderiranom zbroju stavke odgovori. Suprotno tome, u P&P su odgovori jednako ponderirani.
- Test proveden na računalu ispitanicima će pružiti drugačije iskustvo od klasičnog P&P testa. Neke od tih razlika su lakoća čitanja ulomka ili pregled i promjena odgovora, učinci vremenskih ograničenja, jasnoća brojki ili dijagrami i odgovaranje na tipkovnici u odnosu na odgovaranje na listu za odgovore.

### **3.5 Problemi koji onemogućuju potpuno autonomno testiranje**

Udaljeni nadzor ispitanika je najveći problem koji onemogućuje potpuno autonomno testiranje kao i validacija samih rezultata testova zbog mogućnosti varanja ili takozvane akademske nečasnosti (e-Assesment). Kod tradicionalnog pisanja testa papirom i olovkom, ispitanici varaju svakakvim različitim postupcima kao što je korištenje vrlo popularnog



„šalabahtera“ (ilegalnom papiru na kojem se nalaze informacije), ili gledanja u tuđi ispit te prepisivanje odgovora. Kod e-testiranja znanja na online ispitu ispitanici također mogu varati, štoviše, iz njihovog kuta gledanja na cjelokupnu situaciju, oni smatraju da je to vrlo lako i da će lakim putem dobiti pozitivnu ocjenu. Taj dio se donekle rješava odabirom odgovarajućih zadataka i vremenskim limitom. Način varanja koji povezuje tradicionalni i online ispit je krađa identiteta. Prilikom tradicionalnog načina slušanja predmeta nastavnik upoznaje svoje učenike i studente a prilikom dolaska na ispit se može zatražiti validacija osobnih podataka pomoću studentske iskaznice. Što se tiče online ispita situacija je malo drugačija, pa se taj problem pokušava riješiti tehnologijama udaljenog nadzora koje ćemo u nastavku rada objasniti.

## **4. Metode i tehnologije udaljenog (video) nadzora te mogućnosti implementacije za e-testiranje znanja**

Hijerarhijski autonomni sustav za e-testiranje znanja temelji svoju autonomnost na udaljenom nadzoru ispitanika, kojom se u velikoj mjeri anuliraju mogućnosti varanja ispitanika pri testiranju (Purković i Kolumbić, 2012). U današnje doba pametnih uređaja i široke uporabe računala gotovo svi ispitanici ispunjavaju minimalne tehničke uvjete koji su potrebni za provedbu e-testiranja znanja online putem, a to su :

- korištenje osobnog računala (stolno ili prijenosno računalo) ili druge kategorije uređaja (pametni telefon ili tablet),
- Internet pretraživač koji je pogodan za e-testiranje,
- web-kamera sa mikrofonom (ukoliko kamera nema mikrofonski potreban je zasebni mikrofonski),
- priključak na Internet.

Pored nabrojanih tehničkih uvjeta potrebno je osigurati i kvalitetno osvjetljenje prostorije kao i malenu razinu buke u prostoriji u kojoj se piše online ispit znanja. Proces nadzora kao i detekcije, mora se odvijati u realnom vremenu, dok je proces filtriranja i matematičke obrade valjanosti sporedan, on se može odvijati i s odgodom (Purković i Kolumbić, 2012). Zbog toga što se proces nadzora i detekcije mora odvijati u realnom vremenu, potreban nam je vrlo dobar algoritam softverske detekcije pokreta kao i zvuka jer obrada i prijenos video podataka većih rezolucija može dovesti cjelokupni sustav testiranja znanja u stanje loših performansi poput zamrzavanja i na koncu ne prihvaćanje odgovora studenata.

### **4.1 Sustav za prepoznavanje lica (*Face Recognition System*)**

Sustav prepoznavanja lica je patentiran 1992. godine u Americi. To je tehnologija koja može povezati lice s digitalnom slikom ili video okvirom s bazom podataka o licu. Najnaprednija metoda prepoznavanja lica također se koristi za autentifikaciju korisnika putem usluga provjere identiteta precizno određujući i mjereći crte lica pojedine osobe na danoj slici. Iako su izvorno računalna aplikacija, sustavi za prepoznavanje lica nedavno su se počeli naširoko koristiti i u pametnim telefonima i drugim oblicima tehnologije. Budući da

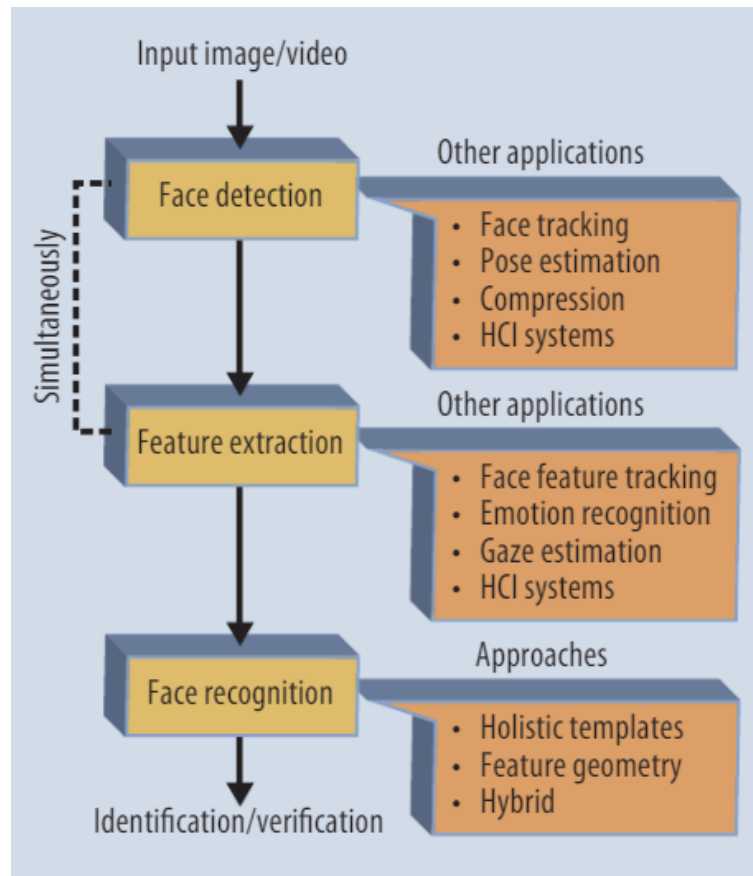
prepoznavanje lica putem računala uključuje mjerenje ljudskih fizioloških karakteristika, sustavi prepoznavanja lica mogu se kategorizirati kao biometrija. Biometrija je tehnika za autentikaciju koja koristi unikatne fizičke karakteristike svakog pojedinca poput otisaka prstiju, šarenice oka i sl., kao oblik identifikacije i kontrole pristupa. Iako je točnost sustava za prepoznavanje lica kao biometrijske tehnologije niža od one za prepoznavanje šarenice i prepoznavanje otisaka prstiju, njegovi su beskontaktni i jednostavni principi široko prihvaćeni. Sustavi za prepoznavanje lica ugrađeni su u naprednu interakciju čovjeka i računala, video nadzor i automatsko indeksiranje slika.

Kao i u svakom problemu prepoznavanja uzoraka, varijacije u uzorcima od osvjetljenja, položaja, izraza i sl., rješavaju se ili čineći značajke nepromijenjenima za ove transformacije u fazi izdvajanja obilježja, ili određujući pravila koja objašnjavaju te transformacije u fazi prepoznavanja. Pri izradi sustava za prepoznavanje lica vrlo je bitno paziti na 3 ključne stavke (Chellappa, Sinha i Phillips, 2010), a to su:

1. Verifikacija - sustav prepoznavanja utvrđuje slaže li se osoba prikazana na slici lica s položenim identitetom;
2. Identifikacija - sustav prepoznavanja utvrđuje identitet osobe na slici lica;
3. Popis praćenja - sustav prepoznavanja određuje pojavljuje li se osoba na slici lica na popisu za praćenje i, ako je tako, identificira tu osobu.

Slika 12. prikazuje shemu osnovnog sustava za prepoznavanje lica koji se sastoji od tri glavna modula a to su detekcija lica (*eng. face detection*), izdvajanje značajki (*eng. feature extraction*) i prepoznavanje lica (*eng. face recognition*). Detekcija lica (*eng. face detection*) je prvi korak kod bilo kojeg sustava za prepoznavanje lica. Drugim riječima, pitanje je može li sustav otkriti, odnosno prepoznati fizičke osobe u realnom vremenu na slikama, isječcima ili videa putem videokamera. Pod detekciju lica još spada i praćenje lica (*eng. face tracking*), procjena poze (*eng. pose estimation*), kompresija slike (*eng. compression*) te HCI sistemi. Pravilnom detekcijom lica je vrlo važno utvrditi koje je na slici ili snimci ljudsko lice i razlikovati ga od bilo čega drugog poput pozadine. Kada sustav otkrije ljudsko lice, broj lažno-pozitivnih (*eng. false positive*) rezultata mora se smanjiti pomoću izdvajanja značajki izdvajanjem gore navedenih različitih atributa te atributi koji se izdvajaju se mogu staviti u sustav klasifikacije lica (Chellappa, Sinha i Phillips, 2010). Također za izdvajanje značajki vrijedi praćenje značajki lica (*eng. face feature tracking*), prepoznavanje emocija (*eng.*

*emotion recognition*), prepoznavanje pogleda (*eng. gaze estimation*) kao i HCI sistemi. Iako to ovisi o vrsti klasifikacijskog sustava, atributi se mogu razlikovati od teksture do glavnih uvjerljivih točaka ili dijelova lica poput usta, nosa, očiju pa čak i obrva.



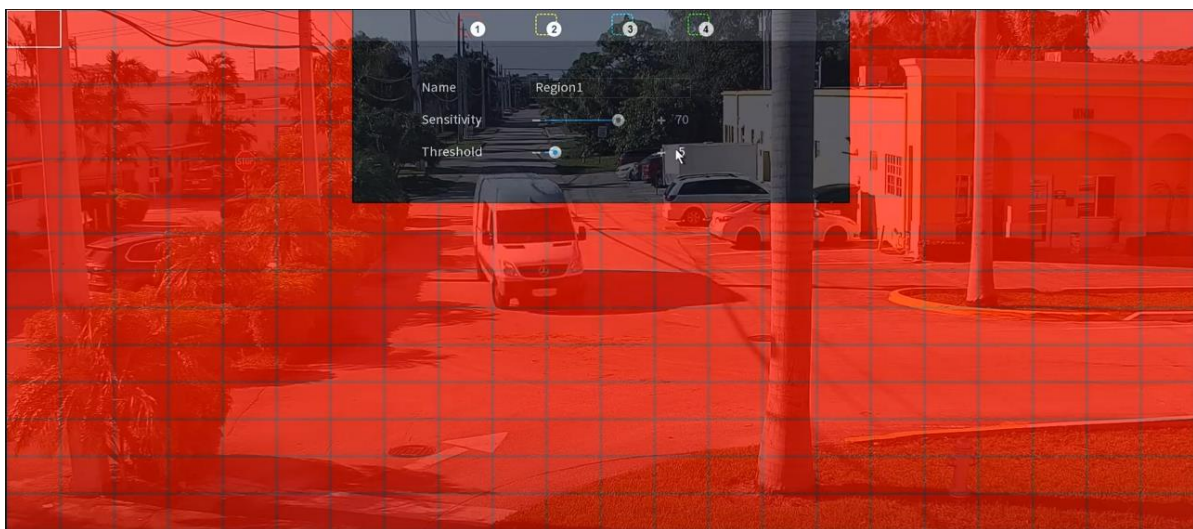
Slika 12. Shematski prikaz osnovnog sustava za prepoznavanje lica (Chellappa, Sinha i Phillips, 2010).

Praćenje lica u sustavu za prepoznavanje lica odnosi se na to može li se osoba neprestano i kontinuirano pratiti u nekim radnjama ili različitim položajima tijekom kretanja u prostoriji ili zgradi pa čak i na otvorenom terenu. Prepoznavanje lica u sustavu za prepoznavanje lica je korak koji razmatra lice koje je pronađeno detekcijom lica višestrukim snimkama praćenjem osobe pristupom holističkih predložaka (*eng. holistic templates*), geometrijom obilježja (*eng. feature geometry*) ili hibridom metodom, koja se kreće kroz prostor te ga uspoređuje s pohranjenim snimkama u bazi podataka gdje se na kraju identificira lice pozitivnim rezultatom. Najveći problem sustava za prepoznavanje lica je taj što postoji mnogo uvjeta za kvalitetno prepoznavanje. Vrlo su važni različiti uvjeti dobivanja slike ljudskih lica jer ljudi prolaze kroz mnoge promjene u životu koje bitno utječu na identifikaciju. To su različiti izrazi lica, lice se prirodno mijenja starenjem kao i fizički položaj lica u odnosu na kameru,

osvjetljenje, zamućene konture i broj piksela na području na kojem lice prikazuje detalje svoje detalje (Chellappa, Sinha i Phillips, 2010). Ostale varijante su kada osoba pokrije lice šeširom ili u današnje vrijeme zaštitnom maskom ili kada osoba nosi sunčane naočale ili kod muških osoba koji imaju veliku bradu na jednoj snimci a na drugoj su obrijane brade, također gubljenje ili dobivanje težine znatno utječe na uspješnost prepoznavanja lica.

#### 4.2 Sustav za detekciju pokreta (*Motion Detection System*)

Sustav detekcije pokreta je prvi put korišten u Drugom Svjetskom ratu kao način detektiranja neprijateljskih vozila na kopnu i moru netom prije napada. Prvi detektori pokreta su bili ultrazvučni i korišteni su u vojne svrhe sve do 1970.ih kada su postali uobičajeni dijelovi kućne sigurnosti u alarmima, dok danas imaju vrlo široku primjenu. Jedna od primjena je implementacija sustava detekcije pokreta u sustave e-testiranje znanja u svrhu udaljenog nadzora.



Slika 13. Izgled postavki nadzorne kamere

Osnovna ideja otkrivanja pokreta je ta da ako nešto uđe u vidno polje kamere koje je podijeljeno u piksele, aktivira odgovor snimača kao događaj kretanja. Pikseli su vrlo male točke ili točke na zaslonu ili slici koje zajedno čine sliku. Kako kamera snima fiksni pogled iz bilo koje scene, većina tih piksela ostat će relativno ista. Uzmimo za primjer nadzornu kameru koja u svom vidnom polju ima ulicu, ulične znakove, drveća i zgrade. Sve stavke se neće često iznenada pomaknuti, s vremenom može doći do suptilnih promjena, poput sjena i osvjetljenja, ali one su spore i postupne. Ako bi u toj istoj sceni prošao automobil ili osoba, ovo je iznenadna i kaotičnija promjena piksela u prizoru nego recimo dan koji se polako

pretvara u noć. Ova iznenadna promjena piksela mogla bi se smatrati događajem kretanja. Osim toga, postoje dvije važne postavke koje određuju točku preokreta u kojoj snimač i kamera smatraju da je bilo koji događaj događaj kretanja. Ove se postavke nazivaju osjetljivost i prag. Kada se u postavkama kreira regija, koja se može vidjeti na slici 13., te da je podijeljena na male odsjeke poput rešetke, a svaku rešetku možemo nazvati kvadratom i područjem koje se sastoji od piksela u sceni.

Osjetljivost (eng. sensitivity) je postavka koja predstavlja koliko se promjena piksela treba dogoditi u pojedinom području kako bi se moglo računati samo to područje kao da je u pokretu. Što je manji broj osjetljivosti postavljen, više se gibanja ili promjene u jednom od ovih područja zanemaruju. Što je veći broj to će lakše smatrati te promjene piksela događajem kretanja. Ovo nas dovodi do postavljanja praga (eng. threshold), kada se stvori regija kao što je prije spomenuto i što se vidi na slici 13., ono se sastoji od manjih kvadrata mreže koje nazivamo područjima. Prag određuje koji postotak ukupnog broja područja u regiji mora doživjeti kretanje kako bi kamera to mogla smatrati događajem kretanja. Na primjer, ako je regija dimenzija 4x4 što znači za ukupno 16 područja, a postavka prag je postavljena na 50 ili 50%, tada bi bilo potrebno najmanje 8 od 16 područja, od kojih 50% doživljava kretanje za aktivaciju okidača za početak snimanja.

Druga postavka u odnosu na oboje je *anti-dither*. To je količina kašnjenja koja će se dogoditi prije početka snimanja, također se može smatrati kao količina vremena u kojem nešto na sceni mora biti prisutno da bi pokrenulo se snimanje i evidentiranje, odnosno to je samo mjerač vremena u rasponu između 0 i 100 sekundi. Počinje nakon što su ispunjeni kriteriji osjetljivosti i praga. Ako je postavljena na 5 sekundi kao što je zadano, tada će morati proći 5 sekundi prije nego što se događaj koji pokreće postavke osjetljivosti i praga smatra zapisnikom kao kretanje. Sve tri ove postavke rade zajedno kako bi pomogle u uklanjanju lažnih alarma. Prednost detekcije pokreta je identificiranje sumnjive aktivnosti u zadanom trenutku. Korisna je za bilježenje promjena u sceni na način da šalje obavijest na telefon ili putem e-pošte ako se dogodi neki događaj. Sljedeća je mogućnost kasnijeg lakšeg pronalaženja snimljenih videozapisa. Kada je događaj pokreta snimljen, on je zapisan u području reprodukcije i području izvoza snimača te je označen određenom bojom na traci za traženje. Obično je to žuta boja, koja se razlikuje od standardne zelene boje pri normalnom snimanju bez pokretanja. Time je kasnije lakše pronaći te snimke. Na kraju, jedna od najtraženijih prednosti detekcije pokreta je ušteda prostora za snimanje. Može se postaviti program tako da snima samo ako otkriju događaj kretanja te se time smanjuje količina

aktivnog vremena koje snimaju, čime se povećava količina dana, tj., količina slobodnog prostora koji se nalazi na HDD ili rjeđe SDD disku. Što se tiče implementacije sustava detekcije pokreta u sustav e-testiranja znanja u svrhu udaljenog nadzora može se postići modifikacijom određenih postavki.

### **4.3 Metoda optičkog toka**

Metoda optičkog toka ima jako važnu ulogu u mnogim zadacima računalnog vida poput segmentacije, prepoznavanju određenih radnji te autonomne vožnje. Optički tok predstavlja jedan obrazac kretanja kao putanje objekata, rubova te površina koje se nalaze u fokusu scene. Putanje se stvaraju projekcijom 3D pokretom objekta kroz vrijeme na 2D površinu. Cilj metode optičkog toka je detektiranje kretanja te određivanje smjera i iznosa istog. Sama analiza pokreta promatranog objekta se može raditi na cijeloj sceni, odnosno slici ili na pojedinim dijelovima iste. Prilikom analize scene, promatrani objekt se ne razlikuje od pozadine nego se izvodi procjena optičkog toka za svaki slikovni element promatranog područja (Smolkovič, 2014). Optički tok se također koristi i u robotici pri prepoznavanju, praćenju i određivanju pokreta objekata prilikom navigacije robota. Da bi se ova metoda mogla uspješno koristiti potrebno je imati jako mali razmak između dvije scene kao slikovni okvir što bi u suštini značilo da su videozapisi sa više sličica (eng. *Frames per Second* – FPS) u sekundi bolji ali i mnogo zahtjevniji zbog količine podataka koji nose sa sobom. Jedna od pretpostavki prilikom korištenja ove metode je da se osvjetljenje nikad ne mijenja što nekada zna predstavljati problem prilikom pojave sjena. Metoda optičkog toka funkcionira na način očitavanja početnih vrijednosti određenih točaka na slici poput jačine svjetlosti, koordinata lokacije piksela te vremena, te ih uspoređujući sa novim podacima. U suštini se u pozadini rade pikselni izračuni koji se odnose na sive tonove slika te što je veća rezolucija slike to su izračuni dulji.

### **4.4 HSV metoda**

HSV metoda poznata kao i metoda oduzimanja pozadine je jedna od osnovnih metoda automatske segmentacije videozapisa. Osvjetljenje okoliša koje je vrlo promjenljivo često zna otežavati rad ove metode. Velika većina videozapisa koristi RGB prostor u boji. Dok korelacije postoje samo između tri komponente boje (eng. red, green, blue što znači crvena, zelena i plava boja), ako se osvjetljenje promijeni sve će se promijeniti te samim time

i dovesti nestabilnost samoj snimci, sličici ili sceni. Upravo zato se koristi HSV model boje (Hue, Saturation, Value) koji je bliži percepciji ljudskog oka i čije se komponente bolje učinkovitije analiziraju i obrađuju odvojeno. Za razliku od kombinaciju boja kod RGB-a, ovaj model ima tri zasebne komponente a to su nijansa (H), zasićenost (S) te vrijednost (V). Algoritam se može lakše prilagoditi različitim uvjetima osvjetljenja okoliša te samim time postaje i stabilniji. Sjene se također otkrivaju te je predstavljen novi algoritam za pozadinsko ažuriranje na temelju zapažanja da su promjene osvjetljenja privremene i da neće utjecati na sve sljedeće sličice, scene. Svi oni pridonose robusnosti metode. Rezultati dobiveni u praksi pokazuju da ova metoda oduzimanja pozadine može automatski segmentirati video objekte robusno i precizno u raznim svijetlećim okruženjima (Zhao, Ming i Bu, 2002).



## **5. Razrada modela udaljenog nadzora ispitanika pri e-testiranju znanja**

U daljnjem tekstu će se razraditi model udaljenog nadzora ispitanika pri e-testiranju znanja te ćemo u njega implementirati kombinaciju tehnologija udaljenog nadzora. Valjanost svakog sustava za provjeru znanja, uključujući klasične ispite na papiru kao i elektroničke sustave testiranja, temelji se na valjanosti sadržaja, odnosno o izboru pitanja i zadataka. Zbog pretpostavke da ispitanici prilikom pisanja ispita mogu varati, što itekako ovisi o rezultatima testiranja, ispit se mora odvijati u ograničenom vremenu i kontroliranim uvjetima. Ovakva valjanost, za potrebe ovog rada, može se nazvati trivijalnom valjanošću. Hijerarhijska autonomija elektroničkog sustava provjere znanja temelji se na daljinskom nadzoru ispitanika, čime se u velikoj mjeri eliminira mogućnost varanja ispitanika tijekom testiranja. Sustav daljinskog nadzora mora osigurati trivijalnu valjanost prihvatljive razine, a te se postiže minimalnim potrebnim tehničkim preduvjetima. Kao što je već napomenuto u poglavlju 3.1.4, minimalni uvjeti koji su potrebni za provedbu e-testiranja su osobno računalo s nekim web-preglednikom prihvatljive verzije, pristup Internetu te web-kamera s mikrofonom. Ove uvjete možemo klasificirati kao opće preduvjete za provedbu elektroničkog testiranja provjere znanja.

Podsustav za nadzor bi se trebao odvijati pod uvjetom da je mrežna veza između računala ispitanika i poslužitelja putem Interneta stalna, odnosno da ne dolazi do prekida veze, što vrijedi i u obrnutom smjeru te se podrazumijeva da je to današnji standard. Međutim, obrada i prijenos videa ili slikovnih podataka mogu predstavljati problem ako se koriste video signali visoke razlučivosti ili algoritam obrade podataka čija optimizacija ne udovoljava zahtijevanoj razini. Temelj treba biti optimizirani softverski algoritam za detekciju pokreta i zvuka, a rezultat testa matematički može biti izražen kao trivijalni brojevi pokazatelj valjanosti testa. Sam proces praćenja i detekcije trebao bi se provoditi u stvarnom vremenu, Proces filtriranja i matematičke provjere može se odgoditi kako bi veza bila stabilnija i sami ispit responsivniji.

### **5.1 Provjera identiteta**

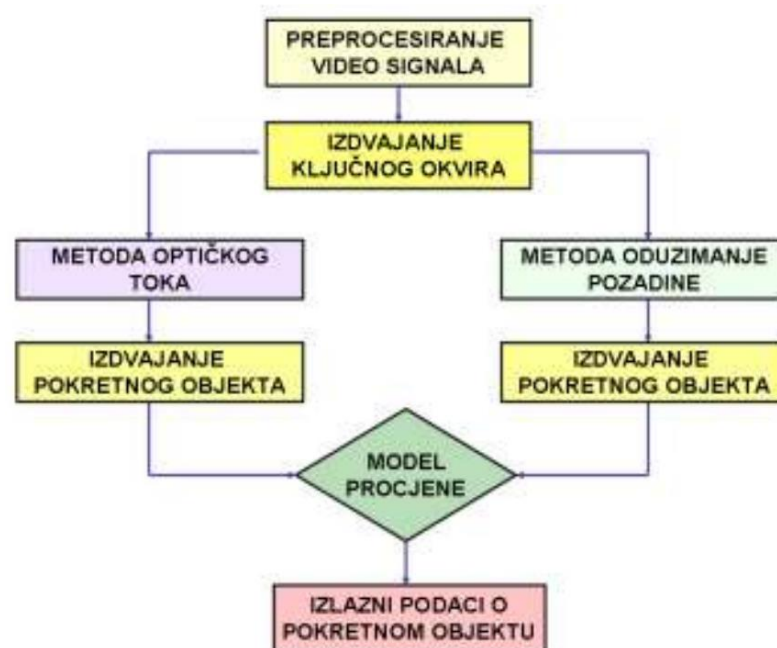
Na klasičnom testu na papiru, nastavnik i učenici su zajedno u učionici te nastavnik ima „fizičku kontrolu i nadzor“ nad učenicima. Pod time mislim kako nastavnik poznaje svoje učenike te ako bi u slučaju pisanja testa došlo do promjene osobe, nastavnik bi to uočio i

eliminirao. Kod elektroničkog sustava nastavnik nema fizičku kontrolu i nadzor, stoga potrebno je rješenje za ovaj problem a to je autentikacija korisnika te je to početni korak pri pisanju testa. Naime, za ovaj sustav klasična autentikacija ispitanika nije dovoljna putem lozinke i korisničkog imena jer i dalje nema fizičke kontrole. Stoga će se koristiti sustav za otkrivanje lica (eng. face detection) koji smo ranije u radu već spomenuli. U tu svrhu mora postojati neko polazište za referencu, odnosno fotografije ispitanika jer nam je potrebna veća preciznost, za razliku kada bi bila samo jedna slika. Rad sustava za prepoznavanje lica smo prethodno pojasnili pa ćemo se nadovezati kako bi to u principu radilo. Prilikom registracije ispitanika uzelo bi se par fotografija u različitim uvjetima poput jakog ili slabijeg svijetla te uslikane iz raznih kutova. Te fotografije bi se uparile sa korisničkim imenom i lozinkom. Prije početka ispita slijedi klasična prijava s korisničkim imenom i lozinkom te slijedi autentikacija ispitanika putem web kamere. Ispitanik bi pristupio testu klasičnim putem pomoću lozinke koju nastavnik daje netom prije ispita, zatim bi sučelje elektroničkog sustava tražilo ispitanika da upali svoju kameru kako bi se postupak autentikacije mogao izvesti. Ukoliko autentikacija bude uspješna, ispitanik može krenuti sa pisanjem testa. Ukoliko autentikacija nije uspješna, ponavlja se postupak, a u slučaju višestrukog negativnog rezultata, sustav šalje nastavniku obavijest te se traži da nastavnik obavi identifikaciju ispitanika putem web kamere. Nakon uspješne autentikacije, sustav prikazuje prvo ispitno pitanje te samim time istovremeno počinje udaljeni nadzor ispitanika za vrijeme pisanja, te se podrazumijeva da je u pitanju stalni nadzor ispitanika tijekom rješavanja testa.

## **5.2 Model udaljenog nadzora**

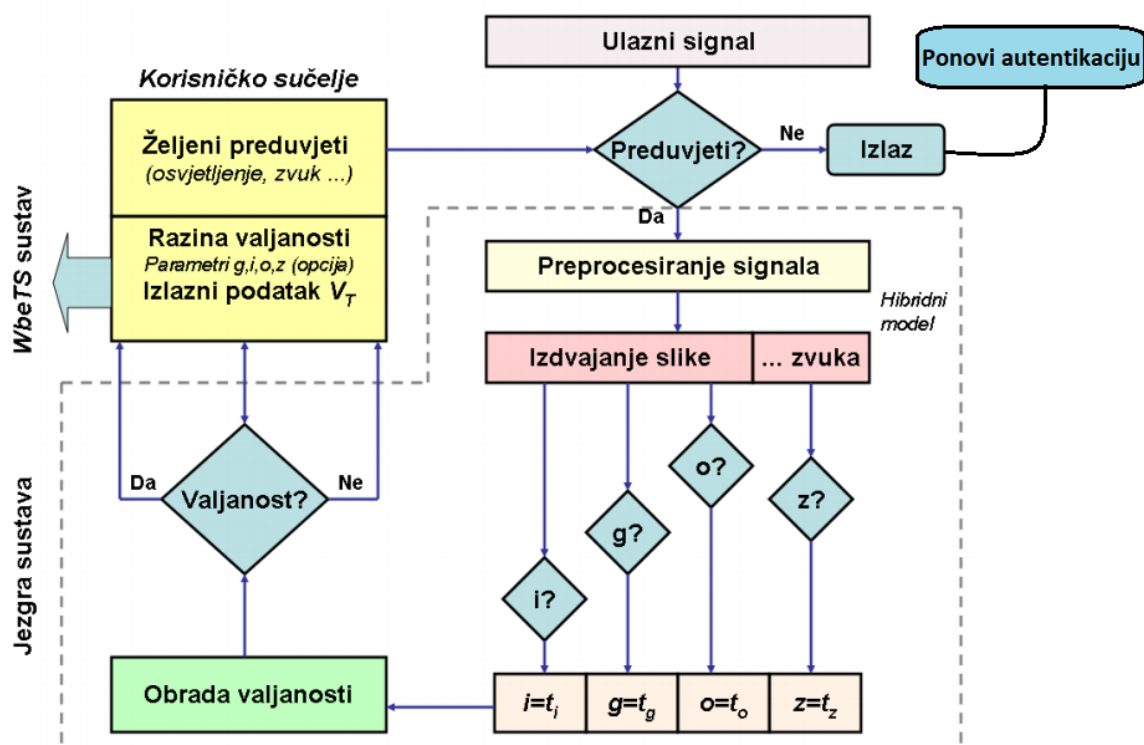
Za vrijeme rješavanja ispita potrebno je nadgledanje ispitanika kako bi se spriječilo moguće varanje. Kod klasičnog ispita to se radi fizičkom prisutnošću nastavnika na ispitu, dok se kod elektroničkog sustava testiranja to rješava sustavom praćenja pokreta. Za sustav praćenja pokreta vizualnim nadzorom razvijeni su različiti algoritmi i tehnike, koje su često prilagođene specifičnostima primjene u određenom području. U poglavlju 4. smo razradili metode i tehnologije, a osim tih navedenih metoda, koje se mogu koristiti pojedinačno, u novije se vrijeme kombiniraju njihove prednosti integriranjem u različite hibridne metode za detekciju pokreta. Svaka hibridna metoda može biti na drugačiji način dizajnirana a najčešće posjeduje metodu razlike korespondentnih točaka slikovnih okvira. Nakon razdvajanja, istaknuti su ključni kadrovi, odnosno okviri, a metode optičkog toka i oduzimanje pozadine HSV-a na temelju određivanja preciznije metode izvedene modelom ocjenjivanja, određuju

kako će se video niz obrađivati. Onda metoda koja točnije obradi procjenu, daje izlazne podatke o pokretnom objektu koji se onda dalje mogu obraditi. Primjenom ovog hibridnog pristupa, koji je za potrebe ovog rada pojednostavljen, kao što je prikazano na slici 14., mogu se postići najbolje karakteristike detekcije pokreta i kretanja sa minimalnim vremenom kao i računalnim zahtjevima za obradu tog specifičnog video signala. Za detekciju i klasifikaciju audio signala, signala dobivenim pričanjem, tj. govorom ispitanika, prikladan je za korištenje VAD tehnika i algoritam (eng. Voice Activity Detection) koji se danas koriste u gotovo svim raznim aplikacijama. Kako bi uspješno izolirali ostale zvukove od govora ispitanika, potrebno je da VAD algoritam obavi minimalan posao.



Slika 14. Dijagram tijeka hibridne metode vizualnog praćenja pokretnog objekta (Purković i Ban, 2014)

Kada sagledamo sve zahtjeve koje sustav udaljenog nadzora mora imati, a da se nalazi unutar okvira hijerarhijskog sustava e testiranja, u ovom slučaju WbeTS sustavu, te ograničenja sklopovlja, razvijen je koncept modela udaljenog nadzora ispitanika koji možemo vidjeti na slici 15. dolje. Model se zasniva na primjeni adaptirane hibridne metode udaljene detekcije i osnovne karakterizacije pokreta, te na primjeni klasičnog VAD algoritma za detektiranje audio signala te izolaciju ostalih zvukova od govora ispitanika te je nadodan korak autentikacije.



Slika 15. Pojednostavljeni dijagram tijeka modela udaljenog nadzora ispitanika

Na temelju analize pokreta za koje smo odredili da sustav treba nadzirati da bi se zadovoljio prag trivijalne valjanosti, osmišljen je model udaljenog nadzora ispitanika koji smo već spomenuli. Na uzorku od 52 studenta diplomskih i preddiplomskih studija na Filozofskom fakultetu u Rijeci na Odsjeku za politehniku, koji su pisali testove u duljini trajanja do 30 minuta, u sustavu za e-učenje, pri čemu se došlo do zaključka da su za vrijeme pisanja testa studenti u prosjeku treptali 30 puta u minuti i da je svaki treptaj trajao manje od sekunde. U prosjeku je bilo više od 20 pokreta glavom po studentu koje sustav može detektirati kao promjenu vektorske strukture kritičnih točaka postavljenih na rubnim linijama glave. Pri pisanju testa položaj glave ispitanika je uglavnom bio usmjeren prema video kameri, uz sitne anomalije u svim smjerovima, što se u algoritmu treba tretirati kao prihvatljiv pokret. To su uglavnom pokreti poput zijevanja, istezanja, dodirivanje lica i sl., što se detektira kao anomalija. Osim toga, praćene su također i oči studenata čije su u prosjeku s testa izgubile fokus 3,7 puta, što bi u prosjeku značilo nešto malo više od 15 sekundi gubitka fokusa. Zaključak je da za ispravno testiranje u 30 minuta dopušteno 200 sekundi anomalija (Purković i Ban, 2014).

Prije početka pisanja testa nastavnik je dužan upoznati učenike s anomalijama i reći im što se sve podrazumijeva kao anomalija te kolika je vjerojatnost da ju sustav zabilježi, dužan je objasniti kako će ih sustav prije početka pisanja tražiti da zauzmu početnu poziciju, odnosno poziciju pisanja u kojoj su obje ruke na tipkovnici te pogled usmjeren u web kameru, odnosno ekran. To je nužno učiniti kako bi se test mogao provoditi neometano i bez prevelike brige učenika na njihove radnje. Nakon toga sustav za testiranje od učenika traži da zauzmu početni položaj.

U poglavlju 4.2 objašnjeno je kako radi sustav detekcije pokreta, koji je predstavljen na primjeru nadzorne kamere primjer. U ovom slučaju se koristi web kamera kako bi se promatrao ispitanik i njegove radnje. Princip rada je isti ali je potrebno namjestiti postavke tako da se definira koje se točno radnje i pokreti smatraju sumnjivima. Također, u svrhu poboljšanja sigurnosti te ispravnosti kao i pravednosti pisanja testa, svaka sumnjiva radnja se snima te pohranjuje u bazu podataka kako bi se mogla poslje analizirati od strane nastavnika. Već smo ranije rekli kako se proces filtriranja i matematičke provjere može odgoditi kako bi veza bila stabilnija i sami ispit responsivniji, a sve zbog toga ako je videozapis web kamere visoke kvalitete i to što situaciju može značajno pogoršati ako se jedno računalo koristi za obradu, nadovezat ćemo se na ona 52 studenta u rijeci, 52 videozapisa u realnom vremenu.

### **5.3 Princip rada modela udaljenog nadzora**

U nastavku je potrebno objasniti pojednostavljeni i skraćeni princip rada sustava udaljenog nadzora. U sustavu na početku testiranja odredimo granice tolerancije položaja korespondentnih točaka a detekcijom pokreta na ključnim točkama scene, utvrđuje se jesu li položaji točaka, u predviđenim granicama tolerancije. Svaka točka koja nestane, izmakne ili ima poveću razliku međusobnih udaljenosti i/ili površina koje one omeđuju, klasificira se kao anomalija, utvrđuje se trajanje iste, bilježi se u zapisnik i pohranjuje u bazu podataka za daljnju analizu. Izostanak ključnih točaka se detektira kao izostanak ispitanika ili grubi pokret, dok se razlika međusobnih odnosa točaka detektira kao neprimjereni pokret glave. Također, prate se i ključne točke očiju gdje se svako spuštanje točaka ispod predviđene razine zadane u algoritmu na početku testiranja, detektira i mjeri njihovo trajanje. Ukoliko je trajanje manje od 1 sekunde, ovakva se anomalija, odnosno pokret klasificira kao treptaj i sustav ga ne evidentira. Ako je pokret dulji od 1 sekunde, svrstava se u neprimjereni fokus očiju i izdvaja se njegovo vrijeme trajanja te bilježi i evidentira za buduću analizu.

Sustav praćenja i snimanja zvuka govora ispitanika tijekom testiranja temelji se na VAD algoritmu. Kada VAD algoritam detektira zvučni signal i klasificira kao govor, mjeri se njegovo vrijeme trajanja anomalije pri testiranju. Vrijeme trajanja svake od navedenih anomalija pribraja se prethodno zabilježenom trajanju istovrsne anomalije, te se njihovo ukupno vrijeme trajanja na kraju zbraja i pretvara u cjelobrojni iznos trivijalne valjanosti

#### **5.4 Moguće poteškoće pri implementaciji**

Za ovakav model, koji smo konceptualno razradili, potrebna je softverska podrška kako bi model mogao krenuti u rad eksperimentalne provjere. Pri tom su mogući problemi na različitim poslovima i razinama razvoja i prilagodbe (Purković i Ban, 2014):

- a) Problemi pri izradi aplikacije poput kvalitete razvijenih algoritama i programa, koji će se u budućnosti razvijati za implementaciju modela, što uvelike utječe na upotrebu i ostvarenje svrhe, naime kako smo već rekli, rad s videosignalima troši mnogo računalnih resursa, zbog čega je potrebno pro izradi usredotočiti se primarno na dvije stvari, dobru optimizaciju i prilagodbu računalnih resursa te na algoritme praćenja fokusa oka, što je iznimno važno za kvalitetu ove vrste praćenja;
- b) Problemi integriranja u sustav e-učenja, jer je model zamišljen kao podsustav WbeTS elektroničkog sustava za testiranje znanja, koji je pak zamišljen kao integralni dio Moodle-a, već postojećeg sustava za e-učenje. Integracija bi trebala slijediti upute programera kao i sve preduvjete i ograničenja koja je on unaprijed postavio. Najlakše za implementaciju bi bilo model razviti u obliku modularnog dodatka (*eng. plugin*), koji se može samo priključiti na sustav e-učenja ili u elektronički sustav za testiranje znanja.
- c) Problemi pri izboru načina pokretanja i izvođenja aplikacije, jer se rad bilo koje aplikacije temelji na komunikaciji između klijenta i poslužitelja, a ono ovisi o kvaliteti i propusnosti međusobnog povezivanja. Može se odvijati na dva načina ili na poslužiteljskoj strani ili na klijentskoj. Za mali broj ispitanika poslužiteljska strana će uspjeti odraditi posao, no za veći broj ispitanika će brzo potrošiti resurse i srušiti se. Stoga je bolje rješenje rad na klijentskoj strani, gdje bi se nakon završenog testa slali podaci na obradu poslužitelju;

- d) Problemi povezani s preduvjetima za provedbu nadzora poput okruženja za elektroničko testiranje koje može i treba biti mirno i relativno tiho kao radni prostor kod kuće ili kao kod javnih računala ili Internet kafića koje nije poželjno ali je u jednu ruku i izvedivo gdje je sve bučno i dinamično. Osim toga veliku ulogu igra i rasvjeta prostora što uvelike utječe na kvalitetu slike. Zato je važno odrediti preduvjete te ih implementirati u aplikaciju.

## 6. Metodički dio

U ovom radu smo govorili o raznim konceptima i sustavima za elektroničko testiranje znanja i nabrojali smo sve prednosti, stoga ćemo se u posljednjem poglavlju ovog rada dotaknuti velikih ograničavajućih faktora koji se odnose na testiranje znanja u tehničkim područjima. Kako bi se mogli vrednovati ostvarenost ishoda učenja, možemo testirati klasičnim ili elektroničkim putem teorijsko znanje bez ikakvog problema. Naime, do problema dolazi kada bi se trebalo testirati praktično znanje učenika. U jednoj mjeri i možemo zbog prirode same djelatnosti, tj. aktivnosti kao što su programiranje, modeliranje ili tehničko crtanje. Ograničavajući faktor u ovom slučaju je taj da su nabrojane aktivnosti samo dio procesa razvoja proizvoda koji se traži.

Pitanja u WbeTS sustavu koja traže video odgovor učenika, a kako već imamo kameru u prvom planu, dobiva se mogućnost demonstracije nekih kraćih radnji. Ograničavajući faktor u ovom slučaju bi bio opremanje učenika sa potrebnim alatima kao i materijalima što je za početak vrlo nepraktično a logistički gotovo pa neizvedivo te se također ne može u potpunosti kao i na pravedan način evaluirati gotov proizvod. Ukoliko bi učenici bili opremljeni alatom i materijalima, kao alternativa moglo bi se vrednovati učenikovo manipuliranje istih, također kod tehničkog crtanja bi se moglo vrednovati učenikovo snalaženje s priborom za crtanje. Možemo zaključiti kako manipuliranje alatom i materijalom u konačnici možemo vrednovati samo na licu mjesta izvođenja tih operacija.

### 6.1 Nastavni plan i program (kurikulum)

Zbog ograničenih mogućnosti provođenja odgovarajućeg elektroničkog testiranja znanja i vještina u tehničkim područjima, za razradu provedbe e-testiranja korišten je sadržaj iz područja informatike. Područje informatike se odabire zbog prirode sadržaja, što znači da se znanje u potpunosti može provjeriti elektroničkim sustavom testiranja znanja. U nastavku rada, u tablici 3. se može vidjeti prikaz sadržaja informatike u prvom razredu gimnazije. Broj sati informatike u prvom razredu gimnazije iznosi 70, a izvodi se 2 sata tjedno. Nastavni sadržaji ovog predmeta su modificirani zbog uklapanja sadržaja ovog rada, što znači da kada se bude obrađivalo područje programiranja, govoriti će se o konceptu WbeTS sustava. Potrebe sustava dolaze u prvi plan učenja. Modificirani sadržaji povezani s ovim radom su



podvučeni. U nastavku rada nakon tabličnog prikaza modificiranog predmeta će biti prikazana priprema za izvođenje nastave.

Tablica 3. Skraćena i modificirana verzija nastavnih sadržaja informatike za 1. razred gimnazije

<i>Naziv nastavne cjeline</i> <i>Nastavna tema/vježba/projekt</i>	<i>Cilj (teme/vježbe/projekta)</i> <i>Glavne zadaće učenika</i>
Uvod u nastavni predmet	Upoznavanje učenika s nastavnim planom i programom, načinom realizacije i bitnim obavezama koje se rješavaju tokom godine
Matematičke osnove rada računala	Upoznavanje sa bitom, bajtom Analiziranje uloge binarnoga i heksadekadskega brojevnog sustava različitih tipova podataka
Uvod u programski jezik Python	Definiranje logičkih izraza za zadane probleme Analizirati problema, definiranje U/I vrijednosti te uočavanje koraka u svrhu rješavanja problema Primjena jednostavnih tipova podataka
<u>Upoznavanje s WbeTS sustavom</u>	Upoznavanje sa principom rada sustava Upoznavanje sa izgledom sučelja Upoznavanje s tehnologijom udaljenog nadzora
Slijed naredaba	Upoznavanje sa algoritmima Rješavanje zadataka pseudokodom i dijagramom tijeka
Grananje u Pythonu	Upoznavanje sa grananjem Spoznaja uvjeta grananja
Naredbe za ponavljanje	Spoznaja kada i zašto se koriste petlje u programiranju

	<p>Upoznavanje i rad s naredbama for, while, continue, break</p> <p>Razlikovanje rada različitih petlji</p> <p>Samostalno korištenje petlji u programiranju</p>
<p>Programiranje u Pythonu – ponavljanje</p>	<p>Ponavljjanje gradiva rješavanjem zadataka</p> <p>Rješavanje kvizova</p>
<p>Rad s mapama i datotekama</p>	<p>Korištenje mogućnosti sustava za pohranjivanje i organizaciju datoteka</p>
<p>Obrada teksta</p> <p>Obrada tekstualnih rješenja</p> <p>WbeTS sustava</p>	<p>Oblikovanje dokumenta na razini odlomka, stranice</p> <p>Rad s naslovima, tablicama, jednadžbama kao i umetanje sadržaja</p> <p>Izrada prezentacija i dokumenata kao i pripreme za prezentiranje</p>
<p>EU Code Week</p>	<p>Sudjelovanje na aktivnostima</p>

SVEUČILIŠTE U RIJECI  
**STUDIJ POLITEHNIKE I INFORMATIKE**

Ime i prezime: **Matej Luketić**

**PRIPREMA ZA IZVOĐENJE NASTAVE**

Mjesto: Rijeka

Škola: Prva sušačka hrvatska gimnazija u Rijeci

Razredni odjel: 1. A

Nastavni predmet: Informatika

Naziv teme: Naredbe ponavljanja

Naziv aktivnosti: While petlja u programskom jeziku Python

Datum izvođenja: Rijeka, 4. svibnja 2021.

**Odgojno-obrazovni ishodi:**

**A. 1. 5** definira logički izraz za zadani problem.

**B. 1. 2** primjenjuje jednostavne tipove podataka te argumentira njihov odabir, primjenjuje različite vrste izraza, operacija, relacija i standardnih funkcija za modeliranje jednostavnoga problema u odabranome programskom jeziku.

**B. 1. 3** razvija algoritam i stvara program u odabranome programskom jeziku rješavajući problem uporabom strukture grananja i ponavljanja.

**Ishodi učenja - razrada:** Učenici će moći:

- analizirati problem i razmatrati načine rješavanja problema programskog zadatka pomoću while petlje
- određuje vrstu ulaznih i izlaznih vrijednosti
- primijeniti funkcije unosa i ispisa
- prepoznaje strukturu ponavljanja
- odabrati adekvatan tip podataka za rješavanje problema
- razviti algoritam koristeći se while petljom za ponavljanje

**Očekivanja međupredmetnih tema:**

- **IKT D.4.1.** Učenik samostalno ili u suradnji s drugima stvara nove sadržaje i ideje ili preoblikuje postojeća digitalna rješenja primjenjujući različite načine za poticanje kreativnosti.

**Nastavna načela:**

- Načelo postupnosti – zadaci počinju od jednostavnijih prema složenijim.
- Načelo primjerenosti – odabrani zadaci i primjeri uvažavaju predznanje i učenikovu dob te ukazuje na korelaciju s drugim predmetima.
- Načelo zornosti – pojmovi koji se odrađuju prikazuju se grafičkim prikazima i slikama tijekom demonstriranja.
- Načelo individualizacije – tokom izvedbe praktičnih radova prati je situacija i tempo u razredu te se prilagođava učenicima i pruža individualna pomoć učenicima tokom rješavanja zadataka.

**Metodički oblici koji će se primjenjivati tijekom rada:**

- **Uvodni dio:**

Dijalog s učenicima kroz koji ponavljamo osnovne pojmove programiranja

Dijalog s učenicima gdje utvrđujemo problemski koncept WbeTS sustava

- **Glavni dio:**

Predavanje o while petlji u programskom jeziku Python

Demonstracija while petlje

Individualni rad učenika – izrada vlastitih programa ( G koda )

- **Završni dio:**  
 Evaluacija učeničkih radova  
 Letimični prolaz obrađenog gradiva

### **ORGANIZACIJA NASTAVNOG SATA**

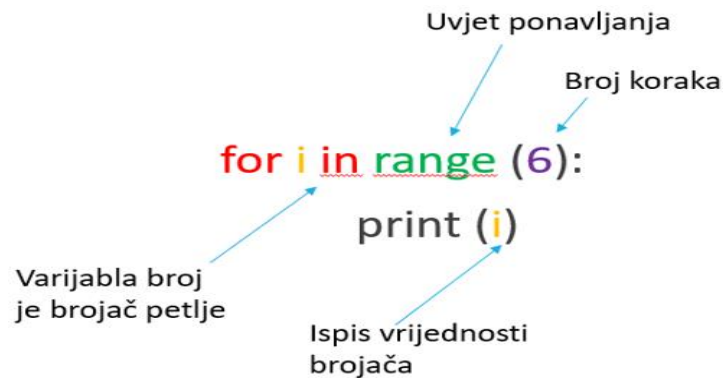
<b>ETAPA</b>	<b>SADRŽAJ</b>	<b>METODIČKO OBLIKOVANJE</b>	<b>VRIJEME</b>
Uvodni dio	Predstavljanje teme  Motivacija učenika	Razgovor o petljama u programiranju i zašto su nam bitne  Razgovor o specifičnim problemima WbeTS sustava Demonstracija while petlje	10 min
Središnji dio	Ponavljjanje prethodne nastavne jedinice  Obrada novog gradiva Rješavanje zadataka	Razgovor o različitim petljama, razlikama i svrsi  Demonstracija nastavnika Praktični rad učenika na računalu	70 min
Završni dio	Pitanja učenika Sistematizacija obrađene teme  Zadavanje zadataka za vježbu	Analiza ključnih točaka  Iznošenje bitnih informacija potrebnih za uspješno rješavanje zadataka za vježbu	10 min

### **TIJEK NASTAVNOG SATA**

#### **Uvodni dio**

Na početku sata najavljujem današnju temu i motiviram učenike za rad. Primjeri koje pokazujem učenicima za početak su problemi povezani s WbeTS sustavom a to su pisanje programa koji će u WbeTS sustavu ispisati listu testova te programa koji će na ekranu ispisati sve polaznike koji su ostvarili mogućnost pristupa trećoj i četvrtoj razini testiranja.

Ukratko opisujem učenicima da su se do sada susreli s naredbama za ponavljanje u prošlom satu te da ćemo sada zajednički riješiti jedan zadatak u svrhu ponavljanja naučenog gradiva te ćemo krenuti na while petlju. Odrađujem s učenicima teorijski dio sata te nije potrebno da učenici koriste svoja računala i Python program. S učenicima ću usmeno proći zadatak za vježbu.



Slika 1. Zadatak za ponavljanje

### Središnji dio

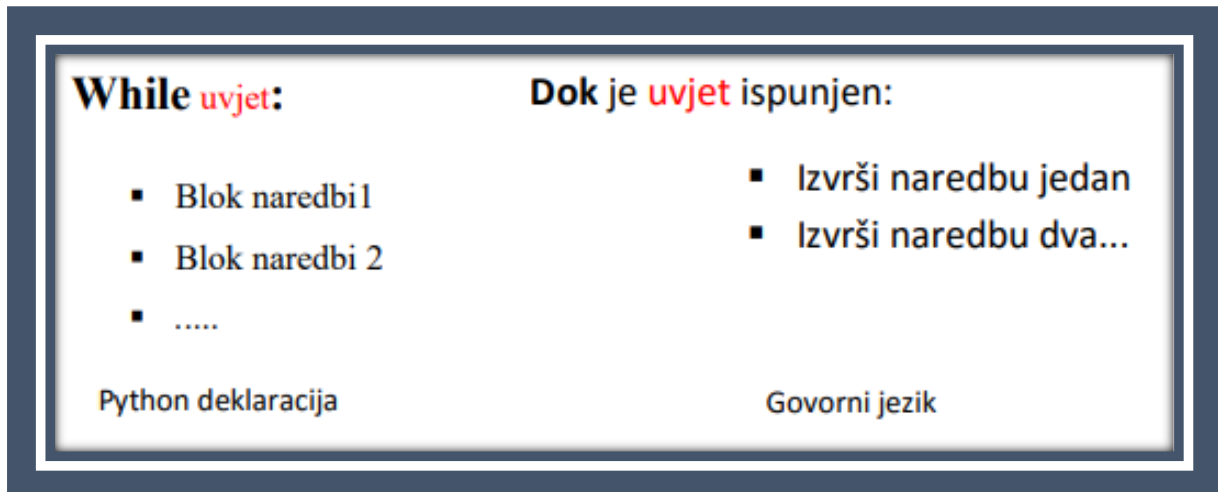
Nakon kratkog ponavljanja, prelazimo na današnje gradivo, programska petlja while. Nadovezujem se na primjer ispisivanja rečenice 10 puta te na korištenje for petlje. Ponavljam učenicima kako su programske petlje programske strukture koje omogućavaju višestruko ponavljanje određenog dijela programskog koda. Razlikujemo programsku petlju for i while. Ukratko ponavljam koja je osnovna razlika između for i while petlje.

Ponavljam učenicima kako programska petlja, identično kao i programska konstrukcija grananja i ili odluke, sadrži uvjet. Kada se taj uvjet ispuni, tj. bude istinit, program nastavlja dalje sa svojim naredbama.

```
1 print(" Ispis brojeva od 0 do 5")  
2 i=0  
3 while i<=5:  
4     print("i= ",i)  
5     i=i+1
```

Slika 2. Primjer while petlje

While petlja je programska struktura koja ima istu namjenu kao i for petlja. To znači da while petlja isto omogućava ponavljanje određenog bloka naredbi. Naravno razlika ove petlje i for petlje je u tome što while petlja ne sadržava brojač kojeg sadržava for petlja. Brojač for petlji omogućava određivanje određenog broja ponavljanja nekog bloka naredbi. While petlja nema brojač, ali kao i for petlja ima uvjet. Taj uvjet joj omogućava da se izvršavaju naredbe, tj. ponavlja blok naredbi tako dugo dok je uvjet ispunjen. Ako uvjet nije ispunjen, nemoj ponavljati taj blok naredbi. Tako bismo mogli prevesti funkcionalnost while petlje.

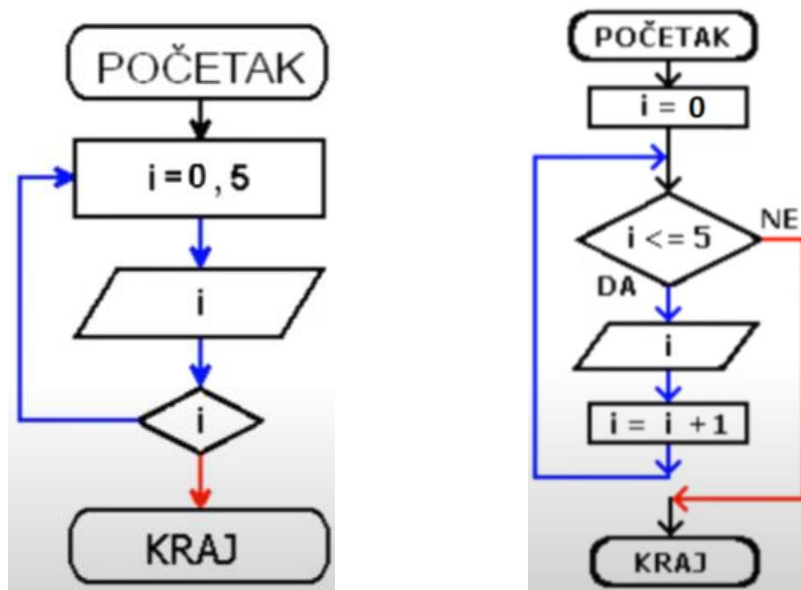


Slika 3. While sintaksa

While petlja je vrlo jednostavna petlja koja samo omogućava ponavljanje kada je određeni uvjet ispunjen. Uvjet u while petlji može biti određen operatorima: usporedbe, logičkim operatorima i aritmetičkim operatorima.

Nakon što smo učenike upoznali sa sintaksom while petlje krećemo na praktične primjere.

U prvom primjeru ćemo napisati program koji će ispisati cijele brojeve od 0 do 5 uporabom petlje for a zatim taj isti program napisati uporabom petlje while. Na ovom primjeru ćemo direktno usporediti kako funkcioniraju for i while petlja. Kako bi učenicima bilo lakše napisati program na desnoj strani prezentacije se nalaze dijagrami tokova za oba programa. Lijevi dijagram toka je za for petlju a desni za while petlju.



Slika 4. Dijagrami tokova

Nakon direktne usporedbe oba primjera krećemo na malo kompliciraniji zadatak u kojem treba napisati program koji ispisuje zbroj unesenih brojeva do broja  $n$  koji je zadan od strane korisnika. Kako bi učenicima bilo lakše, napisani su koraci pri rješavanju, tj. tekstualni algoritam koji treba prepisati u programskom jeziku. Prije nego što učenici krenu sa pisanjem programa napominjem im kako je početna vrijednost varijable zbroj postavljena na nulu dok je varijabla  $i$  postavljena na jedan. Objašnjavam učenicima kako logički uvjet  $i \leq n$  osigurava ponavljanje while petlje dok se izrazom  $zbroj = zbroj + i$  izvodi samo formiranje zbroja. Po izvedenom zbrajanju  $zbroj + i$ , izračunata vrijednost pridružuje se samom zbroju a zatim se uvećava vrijednost varijable  $i$  ( $i = i + 1$ ) i tako sve u krug dokle god varijabla  $i$  ne postane identične vrijednosti kao  $i$  i  $n$ , te se izračunata vrijednost zbroja ispisuje na kraju programa.

Nakon što su učenici napisali program, pokrenuli ga te ispitali njegovu funkcionalnost, svi zajedno komentiramo rješenje programa.



```

print("Zbroj brojeva do unesenog broja n")
n=int(input("Unesi broj n"))
zbroj = 0
i=1
while i<= n:
    zbroj = zbroj + i
    i=i+1
print ("zbroj je", zbroj)

```

Slika 5. Rješenje prvog programa

Nakon toga učenici više neće pisati programe nego ću ja nadograditi prošli program te ga zajedno s učenicima prokomentirati, a radi se o programu koji izračunava zbroj prirodnih brojeva u intervalu od 1 do n koji su djeljivi s 3.

```

print("Zbroj prirodnih brojeva u intervalu od 1 do n koji su djeljivi s 3")
n=int(input("Zbrajaj do broja : "))
s=0
i=1
while i<=n:
    if i%3==0:
        s=s+i
    i=i+1
print("Zbroj je : ",s)

```

Slika 6. Drugi primjer

Nakon što smo prokomentirali drugi primjer tako ćemo na isti način prokomentirati i treći primjer. Modificirati ćemo drugi primjer tako da imamo program u koji unosimo neki prirodni broj u kojem će se zbrojiti sve njegove znamenke.

```

print("Zbroj znamenki unesenog prirodnog broja je : ")
n=int(input ("Unesi prirodni broj : "))
zbroj=0
while n>0:
    zbroj=zbroj+n%10
    n=n//10
print ("Zbroj znamenaka iznosi : ",zbroj)

```

Slika 7. Treći primjer

Nakon što smo prokomentirali i treći primjer prelazimo na Break i Continue. Učenicima govorim kako je break naredba za izlaz iz (najbliže) petlje unutar koje se nalazi a da se nastavak programa izvršava kao da je petlja završila te im pokazujem kako bi to izgledalo na slici 8.

while	< test1>:		
	< naredbe1>		
	if < test2>:	break	# izlaz
	if < test1>:	continue	# na početak
else:			
	< naredbe2>		# ako nije bio break

Slika 8. Izgled Break naredbe

Nakon toga pitam učenike znaju li što je to beskonačna petlja i kako bi mogli doći do nje i zašto nam ona u nekim slučajevima odgovora ili ne odgovara. Zatim učenicima dajem jedan primjer beskonačne petlje u kojoj se traži broj ili riječ stop kako bi se program završio.

Nakon toga učenicima govorim kako je Continue naredba za izlaz iz petlje da se zapravo pokreće idući prolaz kroz petlju što znači prijevremeni prelazak na sljedeću iteraciju petlje u kojoj se nalazi te na vrlo jednostavnom primjeru to i pokazujem.

**Kviz:** <https://wordwall.net/play/15547/978/279>

Nakon što smo obradili teorijski dio dajem učenicima kviz za ponavljanje naučenog gradiva koji nije za ocjenu već će se ostvarenost ishoda provjeravati pitanjima za provjeru znanja koji su navedeni u sljedećem poglavlju.-+

# While petlja ima brojač



Slika 9. Prvi zadatak u kvizu

# While petlja nema uvjet



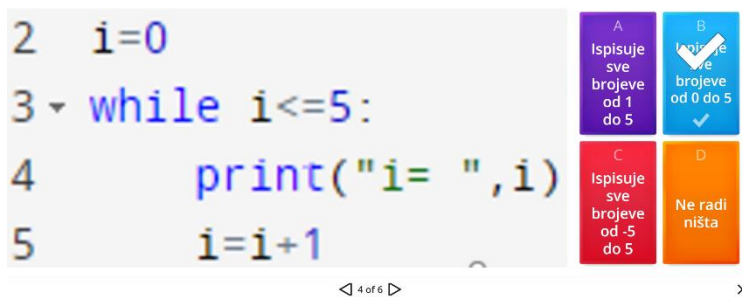
Slika 10. Drugi zadatak u kvizu

# While petlja se ponavlja dokle god :



Slika 11. Treći zadatak u kvizu

Što radi program sa slike ?



Slika 12. Četvrti zadatak u kvizu

# Continue je naredba za ?



Slika 13. Peti zadatak u kvizu

## Koji program sa slike računa zbroj brojeva djeljivih sa 3?



Slika 14. Šesti zadatak u kvizu

Nakon što su učenici završili sa kvizom kreću na rješavanje zadataka.

### Zadaci za vježbu:

1. Napisati program za ispis parnih prirodnih brojeva od 1 do 37 WHILE petljom

Rješenje:

```
1 # Ispis parnih prirodnih brojeva od 1 do 37
2 print("Parni od 1 do 37")
3 i=1 # pocetna vrijednost
4 while i <= 37: # dok je i <= 37 ponavljaj
5     if i % 2 == 0: # provjera da li je broj paran
6         print(i, end=' ') # ispis parnog
7     i = i + 1 # sljedeca vrijednost kontrolne varijable i
```

2. Napisati program za ispis kvadrata prirodnih brojeva od 1 do 5 WHILE petljom

Rješenje:

```
1 # Napisati program za ispis kvadrata prirodnih brojeva od 1 do 5 WHILE petljom.
2
3 print("Ispi kvadrata prirodnih brojeva od 1 do 5")
4
5 i=1 # pocetna vrijednost i=1
6 while i <= 5: # dok je i <= 5 ponavljaj - pocetak petlje
7     print("broj=", i, " kvadrta broja=", i*i) # ako je ispunjen uslov izračunaj proizvod
8     i = i + 1 # sljedeca vrijednost kontrolne varijable i
```

3. Napisati program za ispis prirodnih brojeva djeljivih sa 5 od 1 do n WHILE petljom

Rješenje:

```

1 # Ispis prirodnih brojeva djeljivih sa 5 od 1 do n.
2 print("Ispis prirodnih brojeva do n koji su djeljivi sa 5")
3 n = int(input("Unesite broj n = "))
4 i=1 # pocetna vrijednost
5 print("Brojevi su : ") # naslov
6 while i <= n: # dok je i <= n ponavljaj - pocetak petlje
7     if i % 5 == 0: # i djeljivo sa 5?
8         print(i) # ispis djeljivog sa 5
9     i = i + 1 # slijedeca vrijednost kontrolne varijable i

```

### Dodatni zadaci:

Za one učenike koji žele znati više zadajem jedan zadatak.

1. Napraviti program koji će korisniku omogućiti pogađanje brojeva. Ako korisnik upiše bilo koji broj, izvršavati će se blok naredbi ispod while petlje. Ako odabere 0, dogodi se prekid programa. Prekid programa omogućiti sa naredbom break. Ako korisnik upiše točan broj, ispiše se poruka o pogođenom broju i program se dalje izvršava. Ako korisnik napiše prevelik ili premali broj od traženog, ispisati prigodnu poruku korisniku i dalje izvršavati program, sve dok korisnik sam ne odabere opciju 0

Rješenje:

```

print ("Igra pogađanja brojeva")
print ("Za prekid igre unesite 0")

pogodi=17

while 1:
    broj=int(input("Unesite broj: "))
    if broj==pogodi:
        print ("Pogodili ste zadani broj cestitamo!")
        broj=0
    elif broj==0:
        break
    elif broj > pogodi:
        print ("Broj je veci od zadanog, pokusajte ponovo")
    elif broj < pogodi:
        print ("Broj je manji od zadanog, pokusajte ponovo")

```

### Završni dio

Nakon što smo završili s rješavanjem zadatka, učenike potičem da postavljaju pitanja vezana uz današnje gradivo. Učenici na kraju sata rješavaju kviz za ponavljanje gradiva. Zadajem učenicima zadatke za domaću zadaću za one koji žele više.

## Evaluacija učenika

Evaluacija sadržaja i vježbi koje slijede nakon obrade ove teme, provesti će se elektroničkim putem pomoću WbeTS sustava. Elektroničko ispitivanje provoditi će se pitanjima koja su podijeljena prema skupinama, odnosno razinama u Bloomovoj taksonomiji koju smo na početku ovog rada pojasnili. Prilikom provjere znanja biti će korištena tehnologija udaljenog nadzora koju smo u 5. poglavlju razradili. Primjeri tih pitanja, kao i elementi vrednovanja i kriteriji, navedeni su u slijedećem poglavlju ovog rada.

### 6.2 Pitanja za provjeru znanja

Slijede pitanja odabrana za svaku skupinu elektroničkog testiranja pomoću WbeTS sustava za testiranje znanja. Ovaj primjer će koristiti pitanja programskih petlji u programskom jeziku Python. Prva skupina pitanja sadrži najviše pitanja od svih skupina, a ima ih oko 30. U tim će se pitanjima provjeravati činjenično znanje, terminologija i posebni detalji. Za ovu skupinu pitanja ne postoji vremensko ograničenje, a ta skupina služi, kako smo to već rekli u točki 3.1.4, kao ulaznica za test.

Primjer pitanja za prvu skupinu:

- Koje slijedeće naredbe služe za implementaciju petlji?
  - a) true, false
  - b) for, while
  - c) cout, cin
  - d) endl, wsd
  - e) do while, int

- Nadopuni slijedeće tvrdnje:

Kada ne znamo koliko je puta potrebno izvršiti neki dio koda, koristimo \_\_\_\_\_ petlju.

- Utvrdi točnost tvrdnje

Sve petlje napisane FOR petljom možemo napisati i WHILE petljom.

- 1) točno
- 2) netočno

U drugu skupina pitanja se postavljaju odgovori na pitanja. Druga skupina pitanja sadržava manji broj pitanja od prve skupine pitanja. Pitanja se sastoje od pitanja kategoriziranja, višestrukog izbora, i ili sređivanja a vrijeme se može a i ne mora ograničiti za ovu skupinu.

Primjer pitanja za drugu skupinu:

- Poveži pojmove na lijevoj strani s pripadajućom sintaksom na desnoj strani.
  - a)do while                    1)provjeravanje zadanog uvjeta se vrši na ulazu u petlju
  - b)while                        2)provjeravanje zadanog uvjeta se vrši na kraju svake iteracije
  - d)for                            3)provjeravanje zadanog uvjeta se točno definira za vrijeme pisanja programskog koda
- Što će slijedeći program ispisati kao rješenje?

```
for j in range (4):
```

```
    print (j)
```

- a) j j j j
- b) 0 1 2 3
- c) 1 2 3 4
- d) 4 4 4 4

Treća skupina pitanja ima još manji broj pitanja nego druga skupina te su pitanja kompleksnija. Pitanja u trećoj skupini se odnose na numeričke zadatke te složene zadatke sređivanja i nadopunjavanja, a vrijeme se mora ograničiti za ovu skupinu.

Primjer pitanja za treću skupinu:

- Nadopuni petlju tako tvrdnja bude točna

```
For val in „string“:
```



```
if _____==_____:
```

```
_____
```

```
print(„Kraj“)
```

blok naredbi unutar dane petlje će se izvršiti tako da će ispisati sva slova osim slova i te na kraju ispisati Kraj.

- Poveži pojmove tako da tvore točnu tvrdnju.

```
for j in range (1,9,3) print (j)
```

1) Ispisat će se 1,3,5,7

```
for j in range (1,8,2) print (j)
```

2) Ispisat će se 1,4,7

```
for j in range (1,3,9) print (j)
```

3) Ispisat će se 1

Četvrta skupina pitanja posjeduje još složenije zadatke sređivanja, nadopunjavanja numeričkim i tekstualnim rješenjima kao i neka esejska pitanja. Vrijeme je ograničeno.

Primjer pitanja za četvrtu skupinu:

- Napiši programski kod koji će izračunati sumu prirodnih brojeva u intervalu od 1 do n koji su djeljivi sa 5 sa for i while petljom
- Objasni zašto sve petlje napisane pomoću while petlje ne možemo napisati pomoću for petlje.

Pet skupina pitanja, koja je ujedno i najteža, posjeduje složena esejska pitanja, multimedije i govora te unos programskog koda, a vrijeme je ograničeno i relativno je kratko.

Primjer pitanja za petu skupinu:

- Napiši programski kod koji sadrži četiri opcije kalkulatora (zbrajanje, oduzimanje, množenje i dijeljenje) pomoću for, while i do while petlje.
- Koristeći web kameru, snimi kako svojim riječima objašnjavaš zašto i u kojim slučajevima se koristi break i continue naredba i navedi par primjera.

## 7. Zaključak

Najveća prednost e-testiranja za učenike je ta što se odgovori na testu odmah šalju ispitivaču (nastavniku), a rezultati se dobivaju odmah po završetku pisanja testa. Prednost je e-testiranja ta što omogućava povećanje fleksibilnosti samim time što učenici imaju mogućnost pisanja testa kad god i gdje god to poželeva, a fleksibilnost lokacije i vremena testiranja dovodi do većeg angažmana učenika. Pozitivna stvar je sveopći učinak na okoliš jer je smanjena upotreba papira te to doprinosi i na uštedi troškova. Također postoje i neki problemi te prepreke prilikom implementacije automatiziranog e-testiranja znanja poput same provedbe udaljenog nadzora ispitanika koja je ključna u postizanju trivijalne valjanosti kao i autonomije samog sustava.

U ovom diplomskom radu razrađen je i predstavljen konceptualni model udaljenog nadzora ispitanika kojim se pokušava riješiti problema trivijalne valjanosti. Model se temelji na primjeni hibridne metode za detekciju i klasifikaciju pokreta, sustavu za prepoznavanje, algoritmu filtriranja detektiranih pokreta, te na uporabi standardnog računala s web kamerom te pristupom na Internet. Model je zamišljen u sklopu WbeTS sustava za elektroničko testiranje znanja. Kako bi ovaj model „zaživio“, potrebno ga je softverski razraditi kako bi se uspješno mogao integrirati u takvo okruženje. Primjenom i testiranjem u stvarnim uvjetima bit će moguće obaviti potrebne dorade i poboljšanja, kako sustava tako i samog modela, te stvarnu primjenu i prihvaćenost od strane korisnika. Bitno je napomenuti kako se tehnologija udaljenog (video) nadzora konstanto poboljšava te se zbog pandemije galopirajuće radi na novim tehnologijama i metodama.

Pandemija uzrokovana virusom COVID-19 nam je donijela mnoštvo novih stvari kako onih dobrih, tako i onih loših, a kroz proteklih nekoliko mjeseci praktičnog korištenja elektroničkih sustava za učenje te testiranje znanja, osjeti se velika potreba za učinkovitijim digitalnim sustavima i u upravljanju i u obrazovanju. Nadam se da će ova pandemija i sva događanja vezana za školu i obrazovanje nekako potaknuti Republiku Hrvatsku na poboljšanje svojih digitalnih sustava.

## Literatura

Brusilovsky, P., Miller, P., Web-based Testing for Distance Education, Proceedings of WebNet 99, Honolulu, Hawaii, United States , 1999

Chellappa, Rama & Sinha, Pawan & Phillips, P. Jonathon. (2010). Face Recognition by Computers and Humans. Computer. 43. 46 - 55. 10.1109/MC.2010.37.

Craven, P.(2009) History and Challenges of e-assessment The 'Cambridge Approach' perspective - e-assessment research and development 1989 to 2009, Cambridge Assessment - Assessment Research and Development Division

Du, Y., Wang, X., Wang, S-T., Xue, S., Qin, Z. (2011). Hybrid moving object detection system based on key frame extraction, Industrial Electronics and Applications (ICIEA), 2011 6th IEEE Conference on, 441-445, (ISBN: 978-1-4244-8754-7)

E-Assessment, [https://en.wikipedia.org/wiki/Electronic\\_assessment](https://en.wikipedia.org/wiki/Electronic_assessment) (11.7.2021)

Hoić-Božić, N., Budišćak, I., Botički, I. (2008). Online provjeravanje programskih zadataka unutar sustava za učenje AHyCo. Engineering Review, 28(1), 51-60.

Kellough, R. D. , Kellough, N. G. (1999). Secondary school teaching: A guide to methods and resources, planning for competence. New Jersey: Prentice Hall.

Krathwohl, D. R. (2002). A Revision of Bloom's Taxonomy: An Overview. Theory into Practice, 41(4), 213-218

Kunštek, A. (2016), Diplomski rad, Algoritmi za prepoznavanje pokreta i primjena na automatsko praćenje objekata, Zagreb

Nimac, E. (2011.), Primjena Bloomove Taksonomije U Nastavi, Stručni skup Razine učeničkih postignuća – Bloomova taksonomija znanja za učitelje hrvatskog jezika Primorsko-goranske, Ličko-senjske i Istarske županije

Purković, D., Kolumbić, Z. (2012), A Concept of the Web-Based E-Testing system, MIPRO 2012, Biljanović, Petar (ur.), Rijeka: GRAFIK, 2012. 1514-1520(ISBN: 978-953-233-069-4)

Purković, D., Ban, E. (2014). The Model of Remote Video Surveillance in Hierarchical Autonomous E-testing WbeTS System for Knowledge Testing, MIPRO 2014, Biljanović, Petar (ur.), Rijeka: Prospekt, 1001-1006, ISSN: 1847-3946.

Smolković I., (2014.), Diplomski rad, Analiza i vrednovanje odabranih izvedbenih detalja postupaka optičkog toka, Zagreb

Videosurveillance,

[https://www.videosurveillance.com/blog/technology/how\\_does\\_motion\\_detection\\_work.asp](https://www.videosurveillance.com/blog/technology/how_does_motion_detection_work.asp)

(28.8.2021)

Zhao, Ming & Bu, Jiajun & Chen, C.. (2002). Robust background subtraction in HSV color space. Proceedings of SPIE MSAV, vol. 1. 4861. 10.1117/12.456333.

ZNDM Desa, AA Latif (2008), PROBABILITY THEORY AND APPLICATION OF ITEM RESPONSE THEORY, Julnar Pendidikan Universiti Teknologi Malaysia, Jilid 13 Okrober 2008 hal. 122-133.

## **Popis slika**

Slika 1. Shematski prikaz WbeTS sustava

Slika 2. Proces testiranja u WbeTS sustavu

Slika 3. Arhitektura AHyCo sustava

Slika 4. Prikaz troslojne arhitekture sustava za provjeravanje znanja

Slika 5. Web sučelje sustava za unos AOP programskog pitanja

Slika 6. Pregled rezultata programskog testa

Slika 7. Razlika konvencionalne logike i fuzzy logike

Slika 8. Tok studentskih sposobnosti

Slika 9. Tipična karakteristična krivulja zadatka za 1-PL mode sa tri stupnja težine

Slika 10. Tipična karakteristična krivulja zadatka za 2-PL modele

Slika 11. Tipična karakteristična krivulja zadatka za 3-PL modele

Slika 12. Shematski prikaz osnovnog sustava za prepoznavanje lica

Slika 13. Izgled postavki nadzorne kamere

Slika 14. Dijagram tijeka hibridne metode vizualnog praćenja pokretnog objekta

Slika 15. Pojednostavljeni dijagram tijeka modela udaljenog nadzora ispitanika

## Popis tablica

Tablica 1. Podjela skupina komponenata internetskog obrazovanja

Tablica 2. primjer stupnja pripadnosti ovisno o visi osobe

Tablica 3. Skraćena i modificirana verzija nastavnih sadržaja informatike za 1. razred gimnazije

### PREGLED VELIČINA, OZNAKA I JEDINICA

Oznaka	Opis
AR	rezultat vrednovanja i evaluacije zadatka
AP	maksimalan broj bodova za zadatak
$TP_i$	maksimalni broj bodova za testni slučaj
$TR_i$	bodovi dodijeljeni za testni slučaj