**Projekt Laboratorija računalnog inženjerstva iz Multimedijskih arhitektura i sustava**

1. **UVOD**

Zadatak projekta Laboratorija računalnog inženjerstva iz Multimedijskih arhitektura i sustava bio je razviti sustav za snimanje, obradu, prijenos i prikaz slika. Sam sustav podijeljen je na dva dijela – Ugradbeni sustav za snimanje i prijemni sustav. Ugradbeni sustav za snimanje zadužen je za čitanje slike s kamere, kompresiju i dekripciju slike. Prijemni sustav izvodi dekripciju, dekodiranje i prikaz slike.

Za ostvarivanje tog projekta koristili smo razvojnu pločicu PYNQ-Z1 i modul kamere OV7670. Projekt je razvijan u alatima Vivado i Vitis tvrtke Xilinx.

1. **Ugradbeni sustav za snimanje**
   1. **Block Design**

Prvi korak u razvoju ugradbenog sustava bio je razvoj *block design*-a koji podržava sve zahtjeve i ograničenja zadatka. U tu svrhu razvijen je sljedeći *block design*:

A screenshot of a cell phone

Description automatically generated

Slika 1 Block design

* ***Zynq7 Processing System*** *–* Procesni sustav Zynq Z1 *SoC*-a. Sadrži 2 ARM jezgre te je centralni element sustava. Generira *XCLK*, te izravno komunicira sa OV7670 modulom putem *IIC* sučelja.
  + *XCLK* je frekvencije 12 MHz
* ***AXI GPIO 0*** *–* Komponenta zadužena za održavanje komunikacije sa sklopovima na sustav povezanih putem GPIO sučelja. Ovaj konkretan AXI GPIO komunicira sa gumbima i led diodama na Zynq Z1 pločici. Gumbe i led diode koristimo za korisničko sučelje prezentirano putem UART terminala. Obje konekcije su 4-bitne.
* ***AXI GPIO 1 –*** Komunicira sa Video portom na OV7670 modulu. Konekcija je 11 bitna.
  + *AXI\_GPIO\_1[7-0] – Data*
  + *AXI\_GPIO\_1[8] – PCLK*
  + *AXI\_GPIO\_1[9] – HREF*
  + *AXI\_GPIO\_1[10] - VSYNC*
* ***AXI Interconnect –*** Spaja *AXI GPIO* komponente sa *Zynq7 Processing System*-om.
  1. **Stvaranje platforme u razvojnom okruženju Vitis**

Nakon stvaranja block *design*-a potrebno je stvoriti platformu koju ga implementira unutar razvojnog okruženja Vitis. Prvo je potrebno stvoriti *wrapper* za *block design*, a zatim je potrebno pokrenuti lanac operacija od sinteze do generiranja *bitstreama.* Po završetkutog niza operacija, potrebno je odabrati opciju *Export hardware* koja će stvoriti .xsa datoteku koja sadrži opis hardvera iz koje će unutar sustava vitis biti moguće stvoriti *Platform project.*

Unutar razvojnog okruženja Vitis potrebno je odabrati opciju New platform project, te unutar čarobnjaka za njezino stvaranje odabrati ranije stvorenu .xsa datoteku.

Sada je moguće stvoriti novi Application project koji kao platformu koristi platformu generiranu iz *block design*-a

* 1. **Dohvaćanje slike s OV7670**

Za komunikacija sa GPIO povezanim komponentama koristi se biblioteka **xgpio.h** tvrtke Xilinx. Ta biblioteka omogućava pisanje i čitanje na memorijski mapirane GPIO lokacije. Pomoću funkcija XGpio\_DiscreteWrite i XGpio\_DiscreteRead komuniciramo sa gumbima, led diodama i Video portom modula OV7670.

Prije nego što je uspješno moguće dohvatiti sliku sa kamere, potrebno je postaviti potrebne registre modula OV7670 na prave vrijednosti. Registri se postavljaju pomoću SCCB sučelja. SCCB sučelje razvijeno je od strane tvrtke OmniVision te se temelji na IIC standardu. Iz tog razloga, za postavljanje registara kamere koristimo biblioteke za IIC komunikaciju. To je biblioteka xiicps.h.

Unutar datoteke iic.c razvijene su potrebne metode za inicijalizaciju IIC konekcije i postavljanje željenih registara.

Nakon inicijaliziranja IIC veze i registara kamere, moguće je dohvatiti sliku s kamere. Za to koristimo isječak koda dan u repozitoriju:



Slika 2: Odsječak koda za dohvat slike

* 1. **Upute za korištenje**

Dani Application i Platform project treba dodati u Workspace Vitis razvojnog okruženja. Oba projekta je preporučljivo ponovno izgraditi. Potom, treba spojiti Zynq Z1 s računalom putem USB kabla te je upaliti. Desnim klikom na Application project odabrati Launch on local hardware te tako pokrenuti Aplikaciju na pločici.

Pošto pločica nema vizualno sučelje preko kojeg bi mogli dobiti povratne informacije, potrebno je s njom komunicirati preko konzole koja otvara vezu s pločicom. Za potrebe ovog projekta, korišten je besplatni terminalni emulator PuTTY. Prije nego što je uspostava veze moguća, potrebno je saznati na kojoj je serijskoj liniji pločica spojena s računalom. Nakon toga potrebno je namjestiti vrstu veze na *serial* te upisati ime UART serijske linije i postaviti brzinu komunikacije na 115200. Još je dodatno potrebno isključiti opciju *flow control* . Nakon što su parametri veze postavljeni, potrebno je stisnuti na *Open* pri čemu se otvara prozor konzole.

Pri pokretanju UART konzole pojavljuju se poruke o inicijalizaciji kamere, a zatim glavni izbornik. U glavnom izborniku pritiskom na jedan od četiri gumba odabire se mode fotografije koji želite. Pritiskom na gumb pokreće se dohvaćanje slike s kamere te se na terminalu ispisuju poruke o statusu izvođenja. Sa završetkom prihvaćanja slike, ponovo se ispisuje glavni izbornik.

* 1. **Web aplikacija**

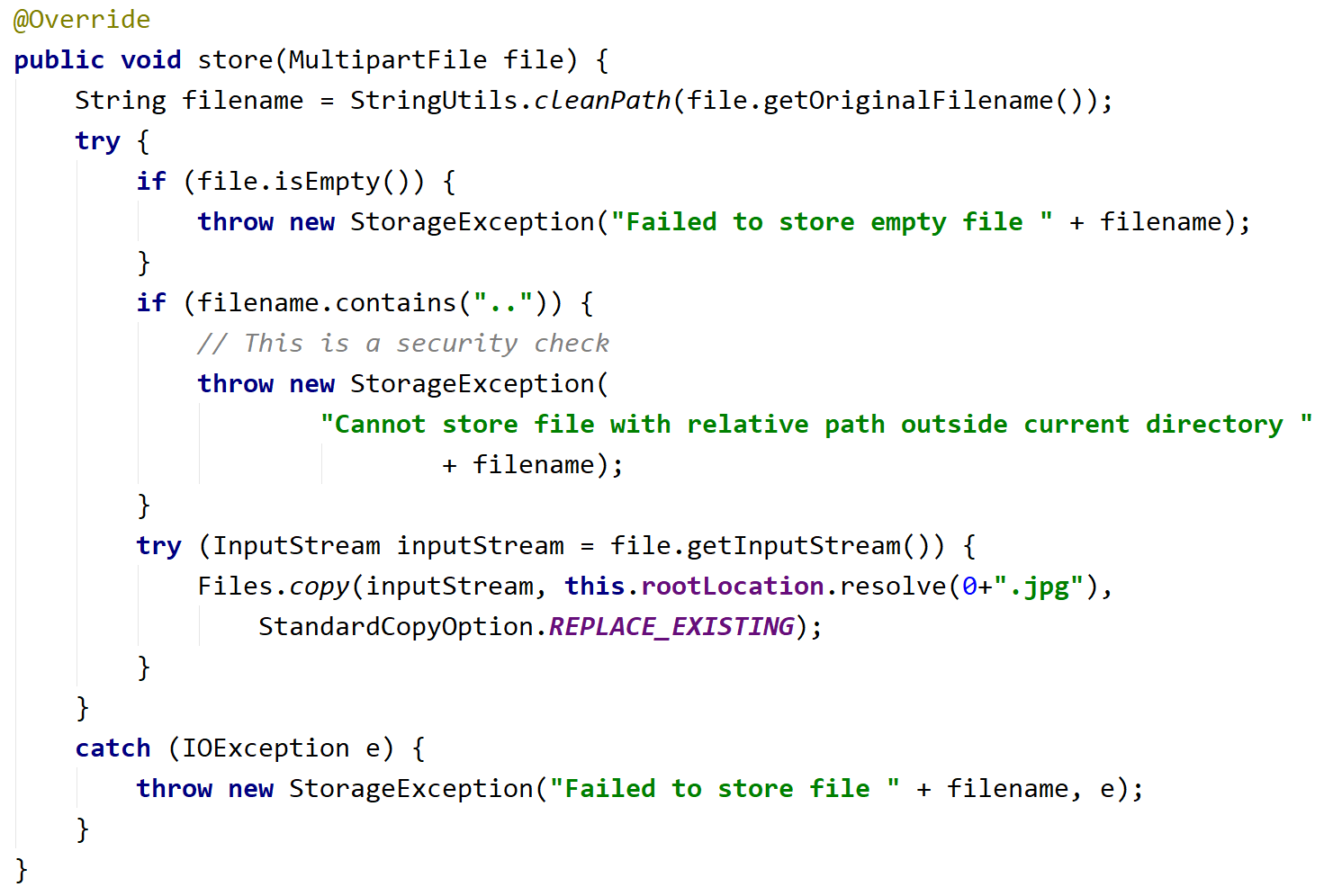
Aplikacija je napisana koristeći Spring razvojni okvir koji koristi programski jezik Java. Kao što je već navedeno poslužitelj na kojem se nalazi aplikacija spojena je Ethernet kablom sa PYNQ-Z1 pločicom. Adresa pločice je 192.168.1.100 što znači da je potrebno na poslužitelju izmijeniti IP adresu kako bi bili u istom rasponu. Klijent (pločica) šalje POST zahtjev preko mreže poslužitelju koji uzima podatke iz zahtjeva, sprema ih u svoj sustav kao datoteku.jpg te ju prikazuje na stanici.

POST http zahtjev prima se na adresu localhost:8080/upload2 te se tamo sprema na disk, rotiraju slike na stranici te se vraća klijentu da je njegova slika uspješno stvorena.



Slika 3 Otvaranje slike

Metoda „store“ opisana je u kodu ispod. Za svaki *file* koji joj pošaljemo ona provjerava je li *file* prazan, ako nije onda ga sprema na definiranu lokaciju, u našem slučaju „/static/uploaded“ te joj dodjeljuje ime 0.jpg i stavlja ju na prvo mjesto za slike na stranici. Prije ove funkcije vrši se funkcija „rotatePictures“ koja samo zamjeni imena slikama kako bi se one mogle rotirati i biti vidljive sa lijeve strane stranice što će biti prikazano u korisničkim uputama.



Slika 4 Spremanje slike

Sav *frontend* web aplikacije opisan je HTML-om i CSS-om uz pomoć Thymleaf biblioteke. Ispod teksta nalazi se isječak iz HTML-a koji pomoću skripte opisuje rad gumba „refresh“. Na pritisak gumba „refresh“ stranica se ponovno učita te se ispiše zadnje vrijeme osvježavanja stranice.



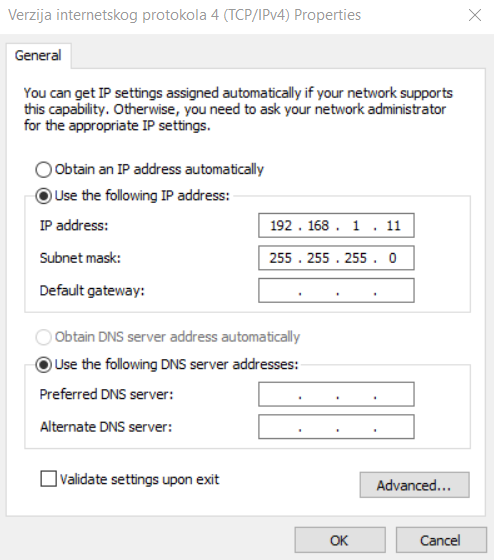
Slika 5 Ispis zadnjeg osvježavanja

**KORISNIČKE UPUTE**

1. Spajanje klijenta i poslužitelja Ethernet kablom.

2. Mijenjanje IP adrese poslužitelja odlaskom u control panel -> network -> change adapters ->

Ethernet -> properties -> IPv4 -> gdje se upisuje proizvoljna adresa u range-u 0-255 osim adrese 10 jer je to adresa klijenta.



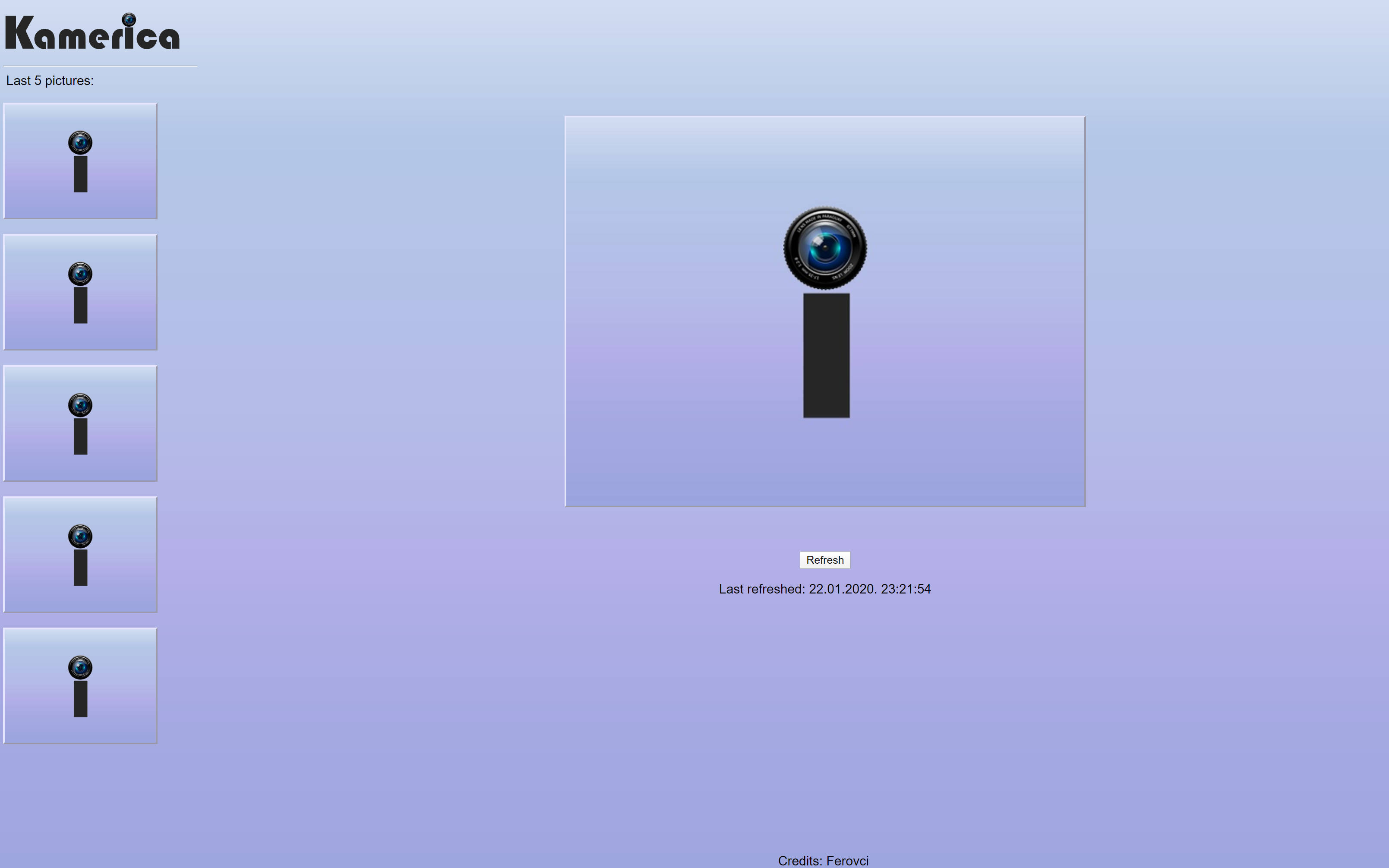
Slika 6 Podešavanje okružja

3. Instaliranje maven, gradel, java jdk8+ te IntelliJ okruženja.

4. Kloniranje projekta sa git repozitorija te otvaranje kroz IntelliJ pomoću build.gradle.

5. Pokretanje aplikacije sa naredbom run->application.

6. Otvaranje Web preglednika te upis „localhost:8080“.



Slika 7 Prikaz stranice

7. Pritiskom gumba na pločici slika se očita te pošalje na web aplikaciju na kojoj je potrebno stisnuti gumb „refresh“ kako bi se nova slika prikazala a ostale slike se zarotirale za jedno mjesto.

* 1. **Slanje slike na server**

Za uspješno slanje slike prethodno je potrebno uspostaviti TCP vezu sa serverom. Port na kojemu server „sluša“ je 8080. Korištene su funkcije iz Xilinx-ove *lwip* biblioteke. Kako bi se *lwip* biblioteka ispravno mogla koristiti, prvo je potrebno dodati određenu biblioteku u projekt te eng. *buildati* projekt. Prije slanja paketa potrebno je inicijalizirati platformu i module lwip. Unutar koda potrebno je podesiti IP adresu servera te IP adresu klijenta, u ovom slučaju je to pločica. Pozivom funkcije *client\_write()* pokreće se samo slanje HTTP paketa na server. Unutar funkcije stvara se nova TCP veza i kada je uspješno stvorena, konkatenira se zaglavlje HTTP zahtjeva i unutar tijela zahtjeva se šalje slika.



Slika 8 Kod za slanje slike na server