SVEUČILIŠTE U ZAGREBU
**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

ZAVRŠNI RAD br. 927

# SIMULACIJA I PRIMJENA MODELA TEORIJE POSLUŽIVANJA

Jure Rajčić

Zagreb, lipanj 2023.

SVEUČILIŠTE U ZAGREBU
**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

ZAVRŠNI RAD br. 927

# SIMULACIJA I PRIMJENA MODELA TEORIJE POSLUŽIVANJA

Jure Rajčić

Zagreb, lipanj 2023.

**SVEUČILIŠTE U ZAGREBU**
**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

Zagreb, 10. ožujka 2023.

# ZAVRŠNI ZADATAK br. 927

| | |
|---|---|
| Pristupnik: | **Jure Rajčić (0036536053)** |
| Studij: | Elektrotehnika i informacijska tehnologija i Računarstvo |
| Modul: | Računarstvo |
| Mentorica: | doc. dr. sc. Lenka Mihoković |

Zadatak: **Simulacija i primjena modela teorije posluživanja**

Opis zadatka:

Sustav posluživanja se uobičajeno prema Kendallovoj notaciji zadaje s četiri parametra, razdiobom vremena između dolazaka, razdiobom vremena posluživanja, brojem poslužitelja te kapacitetom spremnika. Proučava se razdioba broja ljudi u trenutku t, a ovisno o odabranim parametrima mogu promatrati i neke druge vjerojatnosti. Cilj rada je objasniti osnovne pojmove iz teorije posluživanja s naglaskom na M/M/s/m modele te izraditi aplikaciju koja omogućava unos potrebnih parametara a zatim simulira slučajne dolaske te vremena obrade. Aplikacija prikazuje stanje sustava u ovisnosti o vremenu. U sklopu rada je potrebno obraditi jednu konkretnu primjenu, primjerice sustav pružanja medicinske pomoći s obzirom na slučajne dolaske pacijenata u bolnicu.

Rok za predaju rada: 9. lipnja 2023.

*Zahvaljujem se mentorici doc. dr. sc. Lenki Mihoković na potpori i pomoći pri izradi ovog završnog rada.*

# SADRŽAJ

# 1. Uvod

Sustav posluživanja predstavlja temeljnu strukturu u mnogim aspektima svakodnevnog života. Bilo da se nalazite u redu za čekanje na kasi ili se bavite organizacijom medicinskog osoblja u bolnici, oslanjate se na efikasnost sustava posluživanja. Međutim, optimizacija ovih sustava često predstavlja značajan izazov, pogotovo u medicinskom okruženju gdje su resursi ograničeni, a potrebe pacijenata stalno rastu. Matematičke metode i modeliranje igraju ključnu ulogu u rješavanju ovih izazova, omogućavajući nam da predvidimo i unaprijedimo performanse sustava. Jedan od načina kako se matematički koncepti mogu primijeniti na ovakve probleme je kroz simulaciju. Simulacije nam omogućavaju da testiramo različite scenarije i da bolje razumijemo kako različiti parametri utječu na performanse sustava.

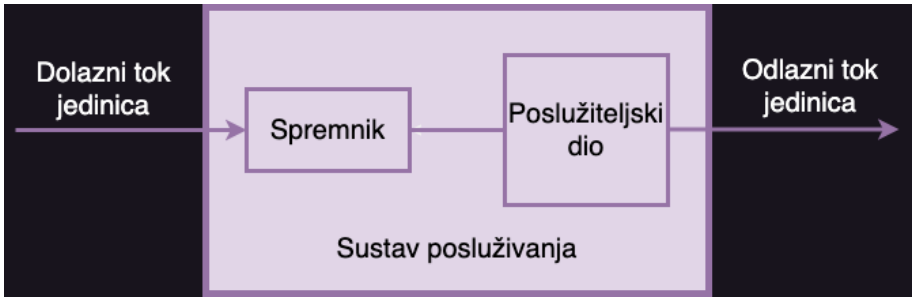
U ovom radu, naš primarni fokus će biti na simulaciji sustava teorije posluživanja. Konkretno, bavit ćemo se simulacijom sustava posluživanja u bolnici, gdje pacijenti dolaze, a medicinsko osoblje ih poslužuje. Kroz modeliranje i simulaciju ovog sustava, cilj nam je pružiti korisne uvide u unaprjeđenje postojećih sustava te poboljšanje njihove učinkovitosti. Koristit ćemo mobilnu aplikaciju za provedbu simulacija. Korisnik pri ulasku na aplikaciju odabire parametre sustava koji se simulira, pokreće simulaciju i prati stanje sustava. Nakon završetka simulacije, korisnik može vidjeti rezultate simulacije te ih usporediti s teorijskim vrijednostima. Matematički koncepti koji se istražuju u ovom radu uključuju Poissonov proces i eksponencijalnu razdiobu. U cilju modeliranja sustava posluživanja, koristit će se Kendalll-ova notacija koja opisuje ključne karakteristike sustava posluživanja. Detaljno će biti objašnjeni matematičke formule i izvodi kako bi se olakšalo razumijevanje teorije posluživanja.

# 2. Osnovni sustavi posluživanja

## 2.1. Opis sustava posluživanja

Osnovni sustav posluživanja predstavlja sustav usluge u kojem skupina poslužitelja prima zahtjeve koji od poslužitelja traže pružanje određene usluge. Inicijalno, kada zahtjev stigne, poslužitelj ga prvo prima. Ako je poslužitelj dostupan, tada zahtjev odmah prelazi na posluživanje. Ako su svi poslužitelji zauzeti kada zahtjev pristupi, zahtjev se prosljeđuje na spremnik. Disciplina sustava određuje redoslijed kojim se zahtjevi iz reda čekanja poslužuju, najčešća disciplina je poznata kao "tko prvi dođe, prvi se poslužuje" (FCFS). Međutim, često se koriste i druge discipline s ciljem povećanja učinkovitosti ili smanjenja čekanja za osjetljivije zahtjeve. Primjerice, u hitnoj službi se primjenjuje sustav trijaže kao disciplina reda čekanja s prioritetom. Po završetku posluživanja zahtjeva u poslužitelju, zahtjev napušta sustav, a poslužitelj se oslobađa za obradu sljedećeg zahtjeva. Ovaj ciklus se ponavlja za svaki dolazni zahtjev, omogućujući sustavu posluživanja da kontinuirano pruža usluge unatoč promjenama u brzini dolaznih zahtjeva ili dostupnosti poslužitelja.

## 2.2. Model sustava i Kendallova notacija




**Slika 2.1:** Općeniti prikaz sustava posluživanja

Sustav posluživanja se sastoji od spremnika i poslužiteljskog dijela. Prilikom kla-

sifikacije takvih sustava koristi se Kendallova notacija:

$$A/B/s/m.$$

1. $A$ označava ponašanje na ulazu, odnosno dolazak klijenata, s mogućim opcijama:

   $M$ - Klijenti dolaze prema homogenom Poissonovom procesu.

   $G$ - Klijenti dolaze prema jednostavnom procesu obnavljanja.

   $D$ - Vremena međudolaska su deterministička.

2. $B$ predstavlja proces obrade zahtjeva s mogućim opcijama:

   $M$ - Vremena obrade su nezavisna i jednako distribuirana s eksponencijalnom razdiobom.

   $G$ - Vremena obrade su nezavisna i jednako distribuirana s proizvoljnom razdiobom.

3. $s$ - predstavlja broj poslužitelja.

4. $m$ - predstavlja kapacitet spremnika, ako je kapacitet spremnika neograničen parametar $m$ možemo izostaviti u notaciji.

## 2.3. Oznake i osnovne relacije

Promatrajmo opći sustav posluživanja s $s$ poslužitelja neograničenog kapaciteta. Neka $C_n$ označava $n$-tog zahtjeva u slijedu zahtjeva koje dolaze u sustav posluživanja. Definirajmo sljedeće slučajne varijable:

$$\begin{aligned}
\theta_n &= \text{vrijeme dolaska zahtjeva } C_n, \\
t_n &= \theta_n - \theta_{n-1} = \text{vrijeme između dolaska zahtjeva } C_n \text{ i } C_{n-1}, \\
x_n &= \text{vrijeme posluživanja zahtjeva } C_n.
\end{aligned}$$

U nekom trenutku $t$ zanima nas ukupan broj zahtjeva u sustavu posluživanja $N(t)$. Broj zahtjeva u sustavu $N(t)$ jednak je razlici između broja zahtjeva $\alpha(t)$ koje su pristigle u sustav do trenutka $t$ i broja zahtjeva $\beta(t)$ koje su poslužene do trenutka $t$. Dakle, imamo:

$$N(t) = \alpha(t) - \beta(t).$$

Cilj teorije čekanja i posluživanja nije samo predvidjeti broj zahtjeva u sustavu posluživanja (uključujući broj zahtjeva u spremniku), već i vrijeme koje zahtjev provode u sustavu. Stoga definiramo još jednu slučajnu varijablu:

$$s_n = \text{vrijeme koje zahtjev } n \text{ provede u sustavu posluživanja.}$$

Ukupno vrijeme koje neka zahtjeva provede u sustavu posluživanja sastoji se od vremena koje provodi u redu čekanja (spremniku) i vremena posluživanja zahtjeva u poslužitelju. Vrijeme čekanja označavamo s:

$$w_n = \text{vrijeme čekanja zahtjeva } C_n \text{ u spremniku.}$$

Zaključujemo da vrijedi:

$$s_n = w_n + x_n.$$

Za navedene varijable definiramo funkcije razdiobe i njihove gustoće:

$$\begin{aligned} A_n(t) = \mathbb{P}(t_n \leq t), \qquad & a_n(t) = \frac{dA_n(t)}{dt}, \\ B_n(t) = \mathbb{P}(x_n \leq x), \qquad & b_n(x) = \frac{dB_n(x)}{dx}, \\ W_n(t) = \mathbb{P}(w_n \leq w), \qquad & w_n(w) = \frac{dW_n(w)}{dw}, \\ S_n(y) = \mathbb{P}(s_n \leq y), \qquad & s_n(y) = \frac{dS_n(y)}{dy}. \end{aligned}$$

Također odredimo očekivanja navedenih varijabli na sljedeći način:

$$\begin{aligned} \mathbb{E}[t_n] = \overline{t}_n, \qquad & \mathbb{E}[x_n] = \overline{x}_n, \\ \mathbb{E}[w_n] = \overline{w}_n, \qquad & \mathbb{E}[s_n] = \overline{s}_n. \end{aligned}$$

Varijable koje smo naveli ne ovise o parametru $n$, stoga promatramo sustav posluživanja isključivo u stacionarnom stanju. Gdje ne ćemo promatrati varijable kao funkciju parametra $n$, već kao funkciju vremena $t$. U slučaju međudolaznog vremena, promatramo varijablu $\overline{t}$ koja je definirana na sljedeći način:

$$\overline{t} = \lim_{n \to \infty} t_n.$$

Tada za sve definirane varijable možemo definirati varijable na sljedeći način:

$$\begin{aligned} t_n \to \tilde{t}, \qquad & A_n(t) \to A(t), \qquad a_n(t) \to a(t), \qquad \overline{t}_n \to \overline{t} = \frac{1}{\lambda} = a, \\ x_n \to \tilde{x}, \qquad & B_n(x) \to B(x), \qquad b_n(x) \to b(x), \qquad \overline{x}_n \to \overline{x} = \frac{1}{\mu} = b, \\ w_n \to \tilde{t}, \qquad & W_n(t) \to W(t), \qquad w_n(t) \to w(t), \qquad \overline{w}_n \to \overline{w} = W, \\ s_n \to \tilde{s}, \qquad & S_n(y) \to S(y), \qquad s_n(y) \to s(y), \qquad \overline{s}_n \to \overline{s} = T. \end{aligned}$$

Očekivanje varijable $\tilde{t}$ koja mjeri međudolazno vrijeme zahtjeva jednako je $\frac{1}{\lambda}$. Parametar $\lambda$ nazivamo prosječnim intenzitetom dolazaka zahtjeva. Slično, očekivanje varijable $\tilde{x}$ koja mjeri vrijeme obrade zahtjeva jednako je $\frac{1}{\mu}$. Parametar $\mu$ nazivamo prosječnim intenzitetom posluživanja.

Prometni intenzitet nekog sustava posluživanja definira se kao omjer očekivanog vremena posluživanja i očekivanog međudolaznog vremena:

$$A = \frac{\text{očekivano vrijeme posluživanja}}{\text{očekivano međudolazno vrijeme}} = \lambda \cdot \overline{x}.$$

Ovdje, očekivano vrijeme posluživanja $\overline{x}$ predstavlja prosječno vrijeme posluživanja za promatrani sustav. Zahtjevi koji pripadaju toku prometnog intenziteta A dolaze u poslužitelj i tamo se poslužuju. U sustavu posluživanja može biti jedan ili više poslužitelja koji rade paralelno. Promatramo samo sustave u kojima su svi poslužitelji koji rade paralelno jednaki. Kada zahtjeva dođe na red za posluživanje, ona će s jednakom vjerojatnošću otići bilo kojem od s poslužitelja u paraleli. Budući da svaki od poslužitelja ima jednaku brzinu posluživanja, prosječno vrijeme posluživanja u poslužitelju će ostati nepromijenjeno bez obzira na broj poslužitelja $s$. Jedina razlika koju postiže arhitektura sustava s više paralelnih poslužitelja je mogućnost obrade više zahtjeva u jedinici vremena nego u slučaju jednog poslužitelja.

Svaki od $s$ poslužitelja poslužuje $1/s$ dio ukupnog dolaznog toka zahtjeva. Promatrajući jedan od tih $s$ poslužitelja, možemo zaključiti da će on određeno vrijeme provesti poslužujući zahtjeve, a jedno vrijeme ne radeći ništa. Ako promatramo dovoljno dug vremenski interval $T$, zbrojimo ukupno vrijeme koje je promatrani poslužitelj proveo poslužujući zahtjeve i podijelimo ga s $T$, dobivamo mjeru koju zovemo faktor opterećenja poslužitelja. Faktor opterećenja se preciznije definira relacijom:

$$\rho = \min\left\{\frac{\text{intenzitet dolaska zahtjeva u poslužiteljski dio}}{\text{intenzitet posluživanja zahtjeva u promatranom poslužiteljskom dijelu}}, 1\right\}$$

Opterećenje je ograničeno s gornje strane vrijednosti 1, jer sustav ne može obaviti više posla od svog kapaciteta. Prometni intenzitet $A$ i opterećenje $\rho$ mjere su koje se izražavaju istom jedinicom (Erlang), ali imaju bitno različita značenja. U slučaju sustava s ograničenim kapacitetom spremnika, ukupni dolazni tok intenziteta $\lambda$ neće biti jednak intenzitetu toka koji dopire do poslužitelja. To je zbog toga što dio dolaznog toka zahtjeva ne može biti prihvaćen u sustav posluživanja zbog prepunjenosti spremnika. Vjerojatnost da je zahtjeva odbijena jednaka je vjerojatnosti da je sustav potpuno ispunjen zahtjevama. Promatramo sustav s jednim poslužiteljem i neograničenim kapacitetom, faktor opterećenja tog sustava definiran je kao:

$$\rho = \frac{\lambda}{\mu}$$

gdje smo iskoristili jednakost $\overline{x} = \frac{1}{\mu}$. U slučaju sustava posluživanja s $s$ jednakih poslužitelja u paraleli, izračunavamo prometni intenzitet $A$ i opterećenje sustava $\rho$. Intenzitet dolazaka zahtjeva je $\lambda$. Potrebno je saznati kapacitet poslužiteljskog dijela sustava, koji je jednak $s\mu$, gdje je $\mu$ intenzitet posluživanja jednog poslužitelja. Opterećenje svakog poslužitelja jednako je:

$$\rho_{\text{poslužitelja}} = \min\left\{\frac{\lambda}{s\mu}, 1\right\}.$$

To je jasno jer se ukupni dolazni tok dijeli na $s$ dijelova, pa je prometni intenzitet svakog poslužitelja $\frac{\lambda}{s\mu}$. Ukupni prometni intenzitet je:

$$A = \frac{\lambda}{\mu}.$$

Također vrijedi jednakost:

$$A = s\rho.$$

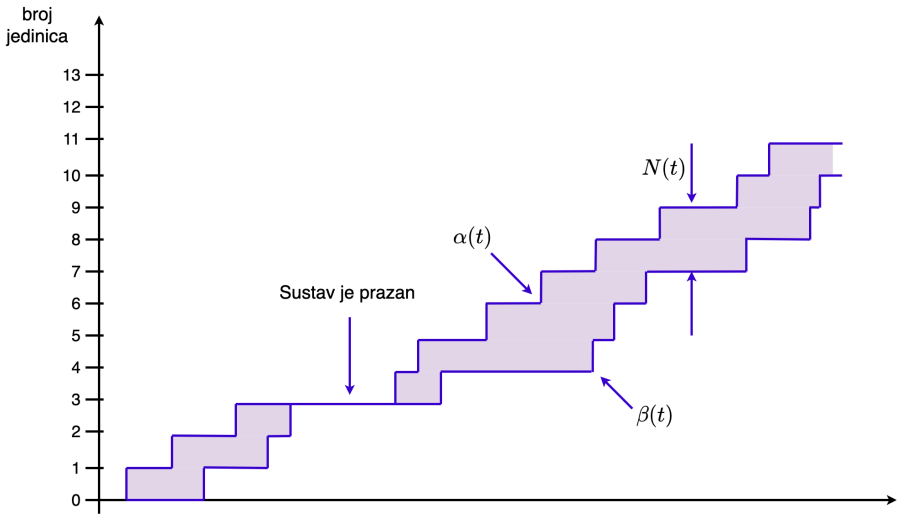
Prometni intenzitet je $s$ puta veći od opterećenja jednog poslužitelja u ovom poslužiteljskom sustavu. Opterećenje cijelog sustava posluživanja jednako je opterećenju jednog poslužitelja u sustavu i definirano je kao:

$$\rho_{\text{sustava}} = \min\left\{\frac{\lambda}{s\mu}; 1\right\} = \rho_{\text{poslužitelja}}.$$

Primijetimo da prometni intenzitet može biti veći od 1, ali opterećenje sustava ili opterećenje pojedinog poslužitelja ne mogu biti veći od 1. Opterećenje svakog poslužitelja u sustavu ne mora nužno biti jednako opterećenju cijelog sustava. Npr., u sustavu s dva paralelna poslužitelja koji poslužuju zahtjevi s različitim intenzitetima, opterećenje može biti različito za svaki poslužitelj. Opterećenje je očekivani udio vremena kada je sustav zauzet, odnosno kada obrađuje zahtjeve. Ako je intenzitet dolaska zahtjeva veći od intenziteta posluživanja, sustav će biti potpuno zauzet. Veliki broj zahtjeva će morati čekati, a red čekanja će neprestano rasti. Sustav je stabilan samo ako je intenzitet dolaznog toka manji od intenziteta posluživanja.

## 2.4. Littleova formula

Izraz za broj zahtjeva u sustavu posluživanja, $N(t)$, definiran je kao razlika između broja zahtjeva koje su ušle u sustav do trenutka $t$ ($\alpha(t)$) i broja zahtjeva koje su bile poslužene do trenutka $t$ ($\beta(t)$). Grafički prikaz odnosa stohastičkih procesa $N(t)$, $\alpha(t)$ i $\beta(t)$ može se prikazati pripadnim trajektorijama na slici.




**Slika 2.2:** Grafički prikaz stohastičkih procesa $N(t)$, $\alpha(t)$ i $\beta(t)$

Promatramo površinu između trajektorija do nekog trenutka $t$ koja predstavlja ukupno vrijeme koje su svi zahtjevi proveli u sustavu do tog trenutka. Tu površinu označavamo s $\gamma(t)$. Ako je $t$ jako veliko, tada je prosječan broj zahtjeva u sustavu:

$$N = \lim_{t\to\infty} \frac{\gamma(t)}{t}.$$

Također, intenzitet dolaska zahtjeva $\lambda$ može se izračunati kao ukupan broj zahtjeva koje su do trenutka $t$ ušle u sustav podijeljen s $t$, ako je $t$ jako veliko:

$$\lambda = \lim_{t\to\infty} \frac{\alpha(t)}{t}.$$

Ukupno vrijeme koje su zahtjevi proveli u sustavu posluživanja do trenutka $t$ podijeljeno s ukupnim brojem zahtjeva koje su ušle u sustav do trenutka $t$ jednak je prosječnom vremenu kojg su zahtjevi proveli u sustavu. Ako je $t$ jako veliko, dobivamo:

$$T = \lim_{t\to\infty} \frac{\gamma(t)}{\alpha(t)} = \lim_{t\to\infty} \frac{\frac{\gamma(t)}{t}}{\frac{\alpha(t)}{t}} = \frac{\overline{N}}{\lambda}.$$

Ovaj rezultat naziva se Littleova formula, a može se zapisati na sljedeće načine:

$$\overline{N} = \lambda T, \quad \overline{N_q} = \lambda W, \quad \overline{N_s} = \lambda \overline{x}.$$

Ovdje $N_q$ označava prosječan broj zahtjeva u spremniku, a $N_s$ prosječan broj zahtjeva u poslužiteljskom dijelu. Također, vrijedi odnos:

$$T = \overline{x} + W.$$

Važno je primijetiti da smo prilikom izvođenja Littleove formule pretpostavili da svi dolazni zahtjevi ulaze u sustav posluživanja. Međutim, to je opravdano samo za sustave s beskonačno velikim spremnikom. Ako je spremnik konačnog kapaciteta, neki zahtjevi koji pokušaju ući u sustav bit će odbačene. To znači da je efektivni intenzitet dolaska zahtjeva za sustav posluživanja manji od stvarnog intenziteta dolaska zahtjeva. Taj efektivni intenzitet označava se s $\lambda_{ef}$ i računa se kao umnožak stvarnog intenziteta dolaska zahtjeva i vjerojatnosti prihvaćanja zahtjeva u sustav posluživanja.

$$\lambda_{ef} = \lambda P(\text{prihvaćanja}) = \lambda(1 - P(\text{odbacivanja})).$$

Zbog prisutnosti konačnih spremnika, Littleove relacije za takve sustave imaju sljedeći oblik:

$$\overline{N} = \lambda_{ef} T, \qquad \overline{N_q} = \lambda_{ef} W, \qquad \overline{N_s} = \lambda_{ef} \overline{x}.$$

Ovdje, $\lambda_{ef}$ predstavlja efektivni intenzitet dolaska zahtjeva, koji je manji od stvarnog intenziteta dolaska zahtjeva zbog odbijanja zahtjeva uslijed ograničenog kapaciteta spremnika.

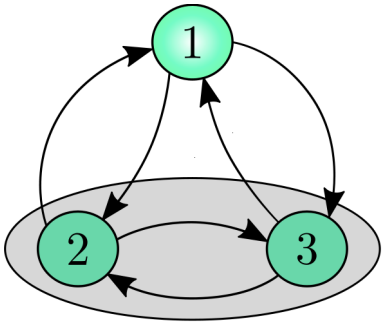
## 2.5. Markovljevi procesi

Markovljevi procesi su temeljni koncept u teoriji vjerojatnosti, s mnogo primjena u različitim područjima, uključujući teoriju posluživanja. Glavna karakteristika Markovljevih procesa je njihovo svojstvo bez memorije, odnosno činjenica da buduće stanje procesa ovisi samo o trenutnom stanju, a ne o prethodnim stanjima.

Definiramo Markovljev proces s neprekinutim vremenom kao skup slučajnih varijabli $\{X(t), t \geq 0\}$ koji za svaki izbor trenutaka $0 \leq t_1 < t_2 < \ldots < t_n < t$ i stanja $x, x_1, \ldots, x_n$ zadovoljava jednadžbu vjerojatnosti:

$$P(X(t) = x \mid X(t_n) = x_n, \ldots, X(t_1) = x_1) = P(X(t) = x \mid X(t_n) = x_n).$$

Markovljev lanac predstavlja specifičnu vrstu Markovljevog procesa. To je niz nezavisnih slučajnih varijabli $X_1$, $X_2$, $X_3$, ...s Markovljevim svojstvom. Moguće vrijednosti $X_i$ formiraju prebrojivi skup $S$, koji nazivamo skup svih stanja lanca. Markovljevi lanci često se opisuju usmjerenim grafom gdje su bridovi označeni vjerojatnostima koje predstavljaju prelazak iz jednog stanja u drugo.




**Slika 2.3:** Primjer Markovljevog lanaca s 3 stanja.

Matrica prijelaza $\pi$ je kvadratna matrica koja definira prijelaze između različitih stanja u Markovljevom lancu. Elementi matrice prijelaza predstavljaju vjerojatnosti prijelaza iz jednog stanja u drugo.

$$\pi = \begin{bmatrix} \pi_{11} & \pi_{12} & \cdots & \pi_{1N} \\ \pi_{21} & \pi_{22} & \cdots & \pi_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \pi_{N1} & \pi_{N2} & \cdots & \pi_{NN} \end{bmatrix}.$$

Za Markovljev proces s neprekinutim vremenom, matrica prijelaza je funkcija vremena, označena s $P(t)$, gdje $t$ označava vremenski interval. Element $P_{ij}(t)$ ove matrice predstavlja vremenski uvjetovanu vjerojatnost da će se proces prebaciti iz stanja $i$ u stanje $j$ u vremenu $t$.

$$P(t) = \begin{bmatrix} P_{11}(t) & P_{12}(t) & \cdots & P_{1N}(t) \\ P_{21}(t) & P_{22}(t) & \cdots & P_{2N}(t) \\ \vdots & \vdots & \ddots & \vdots \\ P_{N1}(t) & P_{N2}(t) & \cdots & P_{NN}(t) \end{bmatrix}.$$

Intenzitet prijelaza između stanja u Markovljevom procesu s neprekinutim vremenom opisujemo matricom $Q$, koja se definira kao derivacija matrice prijelaznih vrjerojatnosti u nuli. Budući da je $P(t)$ stohastička matrica, tj. da su sume poretcima jednake 1, slijedi da su sume po retcima elemenata matrice $Q$ jednake nuli. Ovo dovodi do definicije matrice intenziteta prijelaza, ili matrice gustoće prijelaza:

$$Q = \begin{bmatrix} -Q_{11} & Q_{12} & \cdots & Q_{1N} \\ Q_{21} & -Q_{22} & \cdots & Q_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ Q_{N1} & Q_{N2} & \cdots & -Q_{NN} \end{bmatrix}.$$

Za Markovljev proces vrijede Kolmogorovljeve jednadžbe koje opisuju dinamiku vjerojatnosti prijelaza između stanja sustava, a Kolmogorovljeva jednadžba unaprijed ima oblik:

$$P'(t) = Q \cdot P(t) \tag{2.1}$$

gdje je $P(t)$ matrica prijelaza koja predstavlja vjerojatnosti prijelaza u trenutku $t$, $Q$ je matrica gustoće prijelaza koja definira prijelaze između stanja, a $P'(t)$ je derivacija matrice $P(t)$ po vremenu $t$. Stacionarna razdioba, označena s $\pi = (\pi_1, \pi_2, \pi_3, \ldots)$, definira se kao razdioba za koju vrijedi:

$$\pi P(t) \approx \pi, \quad \forall t \geq 0.$$

Ovo znači da ako trenutno stanje procesa slijedi stacionarnu razdiobu, ona će ostati nepromijenjena nakon što proces evoluira kroz vremenski interval $t$. Želimo da vjerojatnosti prijelaza nakon vremenskog intervala $t$ ne mijenjaju stacionarnu razdiobu, odnosno želimo da vrijedi:

$$\pi P(t) = \pi.$$

Deriviranjem ove jednadžbe po $t$ i koristeći Kolmogorovljevu jednadžbu, dobivamo:

$$\begin{aligned} \frac{d}{dt}(\pi P(t)) &= \frac{d}{dt}(\pi) \\ \pi P'(t) &= 0 \\ \pi Q &= 0. \end{aligned} \tag{2.2}$$

Ova jednadžba nam pruža način da izračunamo stacionarnu razdiobu, što je posebno korisno u praksi jer omogućava analizu dugoročnog ponašanja procesa. Za procese koje ćemo promatrati u nastavku vrijedi da postoji limes koji ne ovisi o $i$

$$\pi_j = \lim_{t \to \infty} P_{ij}(t),$$

gdje $\pi_j$ predstavlja stacionarnu vjerojatnost a također i vjerojatnost da se proces nađe u stanju $j$ nakon dovoljno dugo vremena.

## 2.6. Sustavi posluživanja opisani procesima rađanja i umiranja

Najopćenitiji sustavi posluživanja su oni u kojima vrijedi da su međudolazna vremena i vremena obrade proizvoljno distribuirana. Sustavi posluživanja poput $M/M/1$,

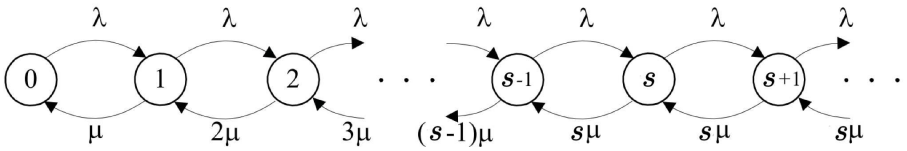
$M/M/1/m$, $M/M/s/m$ su njihovi podskupovi, u kojim su međudolazna vremena zahtjeva eksponencijalno distribuirana s parametrom $\lambda$, koji predstavlja intenzitet dolazaka zahtjeva, te u kojem su vremena obrade u poslužitelju eksponencijalno distribuirane s parametrom $\mu$, koji predstavlja intenzitet posluživanja. Primjeri navedenih sustava su sljedeći:

$M/M/1/\infty$: Sustav s jednim poslužiteljem i neograničenim spremnikom, međudolazna vremena zahtjeva i vremena obrade su eksponencijalno distribuirana.

$M/M/1/m$: Sustav s jednim poslužiteljem i spremnikom kapaciteta $m$, međudolazna vremena zahtjeva i vremena obrade su eksponencijalno distribuirana.

$M/M/s/m$: Sustav sa $s$ paralelnih poslužitelja i spremnikom kapaciteta $m$, međudolazna vremena zahtjeva i vremena obrade su eksponencijalno distribuirana.

Bilo koji sustav posluživanja opisan procesom rađanja i umiranja možemo predstaviti dijagramom stanja na sljedeći način:




**Slika 2.4:** Dijagram stanja za $M/M/s/\infty$ sustav posluživanja.

U procesu rađanja i umiranja, broj zahtjeva u sustavu posluživanja predstavlja stanje procesa u Markovljevom lancu. Koeficijenti $\lambda_i$ i $\mu_i$ su elementi matrice gustoće prijelaza $Q$ koja opisuje stope prijelaza između stanja sustava posluživanja. Ova matrica je često nazvana $Q$-matrica te za procese rađanja i umiranja izgleda ovako:

$$Q = \begin{pmatrix} -\lambda_0 & \lambda_0 & 0 & 0 & \cdots \\ \mu_1 & -(\lambda_1+\mu_1) & \lambda_1 & 0 & \cdots \\ 0 & \mu_2 & -(\lambda_2+\mu_2) & \lambda_2 & \cdots \\ 0 & 0 & \mu_3 & -(\lambda_3+\mu_3) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

Svaki sustav posluživanja karakterizira se matricom gustoća prijelaza $Q$ te specifičnim početnim uvjetima. Proučavamo kako se sustav ponaša nakon što dostigne stacionarno stanje, odnosno, stanje ravnoteže. Do takvog stanja dolazimo promatrajući sustav u dalekoj budućnosti, to jest, kada $t \to \infty$. U kontekstu procesa rađanja i umiranja,

stacionarne vjerojatnosti dobivamo rješavanjem sustava (2.2)

$$\begin{aligned}\lambda_0\pi_0 &= \mu_1\pi_1\\ \lambda_1\pi_1 &= \mu_2\pi_2\\ &\dots\\ \lambda_{m-1}\pi_{m-1} &= \mu_m\pi_m.\end{aligned}$$

S obzirom na prirodu vjerojatnosti, bitno je zadovoljiti uvjet:

$$1 = \sum_{i=0}^{m} \pi_i.$$

U slučaju da sustav ima neograničen spremnički kapacitet, sustav jednadžbi postaje:

$$\begin{aligned}\lambda_i\pi_i &= \mu_{i+1}\pi_{i+1};\ i = 0, 1, \dots\\ 1 &= \sum_{i=0}^{\infty} \pi_i.\end{aligned} \tag{2.3}$$

Riješimo dobiveni sustav jednadžbi za slučaj sustava posluživanja s konačnim brojem stanja $K = s + m$. Izrazimo preostale stacionarne vjerojatnosti $\pi_i$ u ovisnosti o $\pi_0$:

$$\begin{aligned}\pi_1 &= \frac{\lambda_0}{\mu_1}\pi_0\\ \pi_2 &= \frac{\lambda_1}{\mu_2}\pi_1 = \frac{\lambda_1\lambda_0}{\mu_2\mu_1}\pi_0\\ &\dots\\ \pi_K &= \frac{\lambda_{K-1}}{\mu_K}\pi_{K-1} = \frac{\lambda_{K-1}\lambda_{K-2}\cdots\lambda_2\lambda_1\lambda_0}{\mu_K\mu_{K-1}\cdots\mu_3\mu_2\mu_1}\pi_0\end{aligned}$$

a zatim iz jednadžbe (2.3) izračunavamo $\pi_0$:

$$\pi_0 = \left[1 + \sum_{j=1}^{K}\prod_{i=1}^{j}\frac{\lambda_{i-1}}{\mu_i}\right]^{-1}. \tag{2.4}$$

U slučaju kada imamo sustav s beskonačnim brojem stanja, koristimo isto rješenje ali puštamo da $K$ teži u beskonačnost. Poznajući stacionarnu razdiobu stanja sustava posluživanja, možemo izračunati statističke parametre sustava. Izračunajmo prosječni broj zahtjeva u sustavu. Taj broj jednak očekivanju procesa rađanja i umiranja u stacionarnom stanju. Proces rađanja i umiranja je diskretan po vrijednostima i kontinuiran po vremenskom parametru. Ako fiksiramo neki dovoljno velik trenutak $t$ i u njemu promatramo slučajnu varijablu $N(t)$, onda dobivamo diskretnu slučajnu varijablu s

vjerojatnošću stanja $n$ jednakom $\pi_n$. Prosječni broj zahtjeva u sustavu posluživanja jednak je očekivanju te varijable. Očekivanje računamo prema izrazu:

$$\overline{N} = \sum_n n\pi_n.$$

Slično možemo izračunati i varijancu broja zahtjeva u sustavu:

$$\sigma_N^2 = \sum_n (n - \overline{N})^2 \pi_n.$$

U procesu izračuna prosječnog broja zahtjeva unutar sustava, Littleovu formulu koristimo za jednostavnu procjenu prosječnog vremena zadržavanja zahtjeva u sustavu:

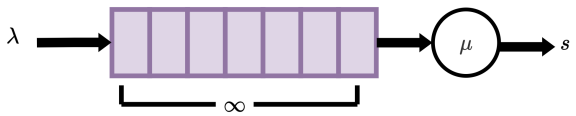
$$T = \alpha\overline{N} = \alpha \sum_n n\pi_n,$$

gdje je $\alpha$ efektivni intenzitet dolaska zahtjeva u sustav posluživanja. Efektivni intenzitet računamo prema izrazu:

$$\alpha = \sum_n \lambda_n \pi_n.$$

### 2.6.1. Sustav posluživanja $M/M/1/\infty$

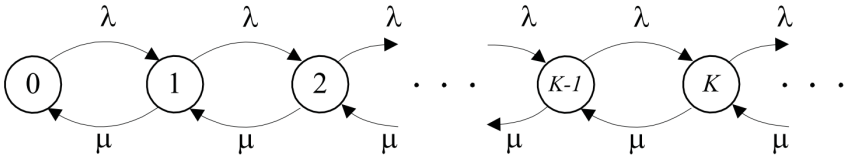
U sustavu posluživanja $M/M/1/\infty$, poslužiteljski dio se sastoji od jednog poslužitelja, pripadajući spremnik ima neograničen kapacitet.




**Slika 2.5:** Sustav posluživanja $M/M/1/\infty$.

Zahtjevi dolaze u sustav s međudolaznim vremenima raspodijeljenim po eksponencijalnoj razdiobi s parametrom $\lambda$. Vrijeme obrade zahtjeva je eksponencijalno distribuirano s parametrom $\mu$. Intenziteti prijelaza u viša i niža stanja su:

$$\lambda_n = \lambda, \quad n = 0, 1, 2, \ldots$$
$$\mu_n = \mu, \quad n = 1, 2, 3, \ldots$$




**Slika 2.6:** Dijagram stanja sustava posluživanja $M/M/1/\infty$ s neograničenim spremnikom za $K$ zahtjeva u sustavu.

Postavimo jednadžbe lokalne ravnoteže:

$$\pi_n = \frac{\lambda}{\mu}\pi_{n-1} = \pi_0 \prod_{i=0}^{n-1} \frac{\lambda}{\mu} = \pi_0 \left(\frac{\lambda}{\mu}\right)^n = \pi_0 \rho^n.$$

$\pi_0$ izračunavamo prema (2.4) uz uvjet da je $\frac{\lambda}{\mu} < 1$.

$$\pi_0 = \frac{1}{\sum_{n=0}^{\infty} \left(\frac{\lambda}{\mu}\right)^n} = \frac{1}{\frac{1}{1-\frac{\lambda}{\mu}}} = 1 - \frac{\lambda}{\mu}.$$

Koristeći formulu $\rho = \frac{\lambda}{\mu}$, slijedi:

$$\pi_0 = 1 - \rho.$$

Sada pomoću $\pi_0$ izračunavamo $\pi_n$:

$$\pi_n = (1-\rho)\rho^n.$$

Prosječan broj zahtjeva u sustavu posluživanja $N$ računamo kao očekivanje slučajne varijable koja predstavlja broj zahtjeva u sustavu:

$$\begin{aligned}
\overline{N} &= \sum_{n=0}^{\infty} n\pi_n = (1-\rho)\sum_{n=0}^{\infty} n\rho^n \\
&= \rho(1-\rho)\sum_{n=0}^{\infty} n\rho^{n-1} = \rho(1-\rho)\frac{\partial}{\partial\rho}\sum_{n=0}^{\infty}\rho^n \\
&= \rho(1-\rho)\frac{\partial}{\partial\rho}\left(\frac{1}{1-\rho}\right) = \rho(1-\rho)\frac{1}{(1-\rho)^2} \\
&= \frac{\rho}{1-\rho}.
\end{aligned}$$

Pomoću Littleove formule računamo prosječno vrijeme koje zahtjev provode u sustavu:

$$T = \frac{\overline{N}}{\lambda} = \left(\frac{\rho}{1-\rho}\right)\left(\frac{1}{\lambda}\right) = \frac{1}{\mu(1-\rho)}.$$

Na ovaj način, dobivamo prosječan broj zahtjeva unutar sustava i prosječnu duljinu trajanja boravka zahtjeva u sustavu. Medutim, zanima nas kakvo je čekanje unutar spremničkog dijela i kolika je prosječna duljina čekanja u redu? Izračunavamo prosječno vrijeme obrade svakog zahtjeva unutar poslužiteljskog sustava i na temelju toga izračunavamo prosječno vrijeme provedeno u redu čekanja. Prosječno vrijeme posluživanja svakog zahtjeva određeno je parametrom eksponencijalne distribucije $\mu$ i jednako je $\frac{1}{\mu}$. Slijedi:

$$W = T - \frac{1}{\mu} = \frac{1}{\mu(1-\rho)} - \frac{1}{\mu} = \frac{\rho}{\mu(1-\rho)}.$$

Prosječan broj zahtjeva u spremniku je:

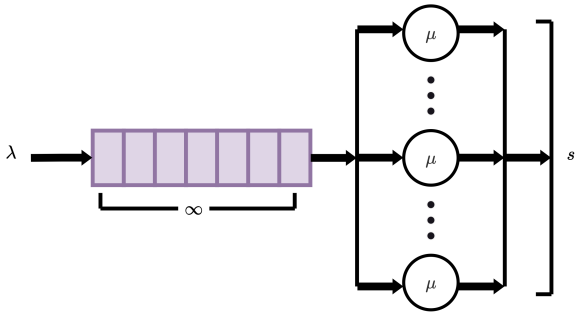
$$\overline{N_q} = \lambda W = \frac{\rho^2}{1-\rho}.$$

Prosječan broj zahtjeva u poslužitelju:

$$\overline{N_s} = \overline{N} - \overline{N_q} = \rho.$$

### 2.6.2. Sustav posluživanja $M/M/s/\infty$

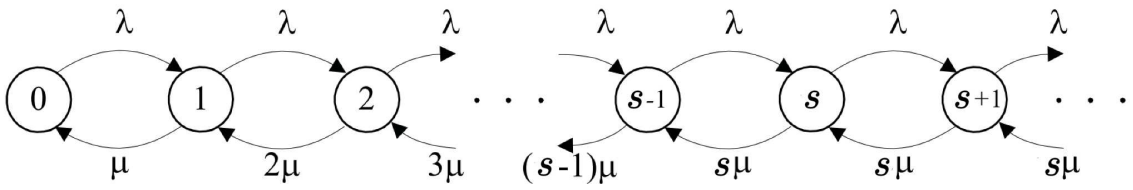
Razmatramo model s neograničenim kapacitetom i $s$ servera koji rade istovremeno, model i dijagram stanja prikazani su na donjim slikama.




**Slika 2.7:** Sustav posluživanja $M/M/s/\infty$.

Zahtjevi dolaze u sustav s međudolaznim vremenima raspodijeljenim po eksponencijalnoj razdiobi s parametrom $\lambda$. Vrijeme obrade zahtjeva je eksponencijalno distribuirano s parametrom $\mu$. Jedan od razloga zašto je intenzitet servera direktno proporcionalan broju zahtjeva u sustavu za $N(t) \leq s$ jeste taj što je broj aktivnih servera identičan broju zahtjeva unutar sustava u toj situaciji:

$$\lambda_n = \lambda, \quad n = 0, 1, 2, \ldots$$
$$\mu_n = min(n\mu, s\mu), \quad n = 1, 2, 3, \ldots$$




**Slika 2.8:** Dijagram stanja sustava posluživanja $M/M/s/\infty$ prije i poslije $s$ zahtjeva u sustavu.

Pronalaženje vjerojatnosti stanja ovog modela predstavlja izazov zbog potrebe za posebnim promatranjem slučaja kada je $n \leq s$ i kada je $n > s$. Iz jednadžbi lokalne ravnoteže proizlaze sljedeće jednakosti:

$$\text{za } n \leq s, \quad \pi_n = \pi_0 \prod_{i=0}^{n-1} \frac{\lambda}{(i+1)\mu} = \pi_0 \left(\frac{\lambda}{\mu}\right)^n \frac{1}{n!} = \pi_0 \left(\frac{\lambda}{s\mu}\right)^n \frac{s^n}{n!}$$

$$\text{za } n > s, \quad \pi_n = \pi_0 \left(\prod_{i=0}^{s-1} \frac{\lambda}{(i+1)\mu} \prod_{i=s}^{n-1} \frac{\lambda}{s\mu}\right) = \pi_0 \left(\frac{\lambda}{\mu}\right)^n \left(\frac{s^s}{s!s^n}\right) = \pi_0 \left(\frac{\lambda}{m\mu}\right)^n \left(\frac{s^s}{s!}\right)$$

Također, trebali bismo zabilježiti funkciju vjerojatnosti slučajne varijable koja meri broj zahtjeva procesa u stacionarnom stanju:

$$\pi_n = \begin{cases} \pi_0 \frac{(s\rho)^n}{n!}, & n \leq s, \\ \pi_0 \frac{\rho^n s^s}{s!}, & n > s. \end{cases}$$

Ipak, za potpuno razumijevanje, moramo pronaći izraz za vjerojatnost $\pi_0$. Pronalazimo ga iz izraza (5.31).

$$\pi_0 = \left(\sum_{n=0}^{s-1} \frac{(s\rho)^n}{n!} + \sum_{n=s}^{\infty} \frac{(\rho)^n s^s}{s!}\right)^{-1}.$$

Prvi zbroj u nazivniku nije moguće eksplicitno izračunati, ali drugi je i to na sljedeći način:

$$\begin{aligned} \frac{s^s}{s!} \sum_{n=s}^{\infty} \rho^n &= s^s s! \left(\sum_{n=0}^{\infty} \rho^n - \sum_{n=0}^{s-1} \rho^n\right) \\ &= s^s s! \left(\frac{1}{1-\rho} - \frac{(1-\rho)^s}{1-\rho}\right) \\ &= \frac{(s\rho)^s}{(1-\rho)s!}. \end{aligned}$$

Pojednostavljenjem izraza za $\pi_0$ dobivamo:

$$\pi_0 = \left(\sum_{n=0}^{s-1} \frac{(s\rho)^n}{n!} + \frac{(s\rho)^s}{(1-\rho)s!}\right)^{-1}.$$

Srednji broj zahtjeva u sustavu posluživanja pronalazimo kao očekivanje slučajne varijable s funkcijom razdiobe.

$$\overline{N} = \sum_{n=0}^{s-1} n\pi_0 \frac{(s\rho)^n}{n!} + \sum_{n=s}^{\infty} n\pi_0 \frac{\rho^n s^s}{s!}.$$

Nažalost, gornji izraz nije moguće jednostavno eksplicitno izračunati. Stoga statističke veličine računamo posrednim putem. Izračunat ćemo prosječnu dužinu čekanja.

$$\begin{aligned}
\overline{N_q} &= \sum_{n=0}^{\infty} nP(N(t) = s+n) = \sum_{n=1}^{\infty} nP(N(t) = s+n) \\
&= \sum_{n=1}^{\infty} n\pi_0 \frac{s^s \rho^{s+n}}{s!} = \pi_0 \frac{(s\rho)^s}{s!} \sum_{n=1}^{\infty} n\rho^n \\
&= \pi_0 \rho \frac{(s\rho)^s}{s!} \frac{\partial}{\partial \rho} \left( \sum_{n=1}^{\infty} \rho^n \right) = \pi_0 \rho \frac{(s\rho)^s}{s!} \frac{\partial}{\partial \rho} \left( \sum_{n=0}^{\infty} \rho^n - 1 \right) \\
&= \pi_0 \rho \frac{(s\rho)^s}{s!} \frac{\partial}{\partial \rho} \left( \frac{1}{1-\rho} - 1 \right) = \pi_0 \frac{(s\rho)^s \rho}{s!} \frac{\rho}{(1-\rho)^2}.
\end{aligned}$$

Sjetimo se iz prethodnog primjera da je opterećenje jednog procesora jednako prosječnom broju zahtjeva koje borave u njemu (koje se obrađuju). Budući da imamo $s$ procesora, zaključujemo da je $\overline{N_s} = s\lambda$.

$$\overline{N} = \overline{N_s} + \overline{N_q} = s\rho + \pi_0 \frac{(s\rho)^s}{s!} \frac{\rho}{(1-\rho)^2}.$$

Prosječno vrijeme zadržavanja u sustavu i čekanja u repu računamo pomoću Littleovih formula. Prosječno vrijeme obrade je $1/\mu$. Vjerojatnost da će zahtjev koji je stigao u sustav biti prisiljena čekati svoj red na posluživanje je jednaka vjerojatnosti da su svi poslužitelji zauzeti i da je u spremničkom dijelu sustava posluživanja $0$, $1$ ili više zahtjeva. Tu vjerojatnost nalazimo jednostavno kao:

$$\begin{aligned}
P(\text{čekanja}) &= \sum_{n=s}^{\infty} \pi_n = \sum_{n=s}^{\infty} \pi_0 \frac{\rho^n s^s}{s!} \\
&= \pi_0 \frac{s^s}{s!} \sum_{n=s}^{\infty} \rho^{n-s+s} = \pi_0 \frac{(\rho s)^s}{s!} \sum_{n=s}^{\infty} \rho^{n-s} \\
&= \pi_0 \frac{(s\rho)^s}{s!} \sum_{i=0}^{\infty} \rho^i = \pi_0 \frac{(s\rho)^s}{s!} \frac{1}{1-\rho}.
\end{aligned}$$

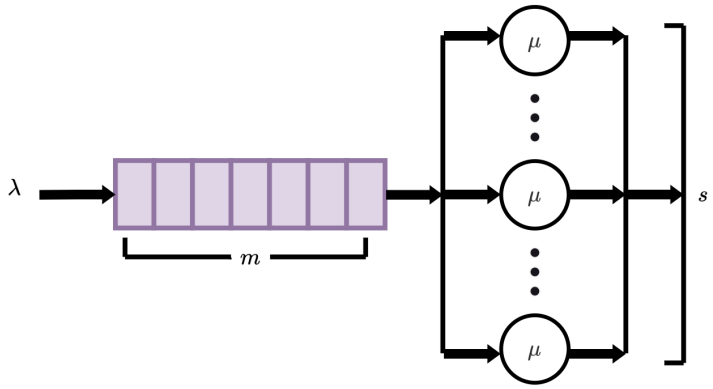
Donji izraz $P_s$ nazivamo Erlangova C Formula koja nam daje vjerojatnost da niti jedan server nije raspoloživ i da dolazni zahtjev mora čekati. Dakako, zahtjev nije odbijen, nego se stavlja na čekanje.

$$P_s = \frac{\dfrac{(s\rho)^s}{s!(1-\rho)}}{\left(\sum_{n=0}^{s-1} \frac{(s\rho)^n}{n!} + \dfrac{(s\rho)^s}{(1-\rho)s!}\right)}.$$

Erlangovu formulu obično koristimo u obrnutom smjeru. Naime, uobičajeni problem koji se javlja pri dimenzioniranju sustava posluživanja je određivanje dovoljnog broja poslužitelja ako se vjerojatnost čekanja želi svesti na neku vrijednost ispod $p_{\max}$.

### 2.6.3. Sustav posluživanja $M/M/s/m$

Ovaj sustav posluživanja predstavlja poopćenje sustava posluživanja $M/M/s/m$. Poslužiteljski dio se sastoji od $s$ poslužitelja, te ima kapacitet za $m$ zahtjeva u spremniku. Ako je spremnik pun, zahtjevi koji stižu u sustav se odbijaju.




**Slika 2.9:** Shema $M/M/s/m$ sustav posluživanja.

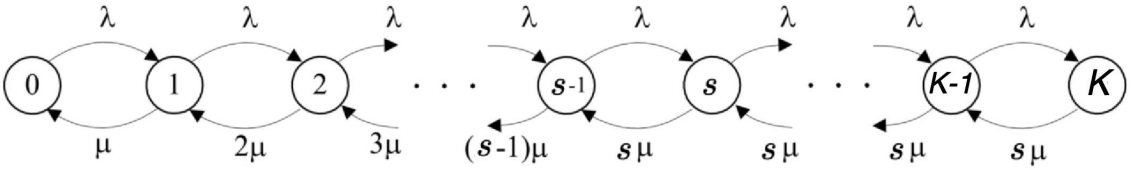
Dijagram stanja za sustav posluživanja $M/M/s/m$ s $K = s + m$ stanja prikazan je na slici ispod, u $K$-tom stanju se svi nadolazeći zahtjevi odbijaju jer je spremnika pun.

Razmatramo jednadžbe lokalne ravnoteže: za raspon $0 \leq n \leq s$

$$\pi_n = \pi_0 \prod_{i=0}^{n-1} \frac{\lambda_i}{(i+1)\mu} = \pi_0 \left(\frac{\lambda}{\mu}\right)^n \frac{1}{n!} = \pi_0 \left(\frac{\lambda}{s\mu}\right)^n \frac{s^n}{n!},$$

za uvjet $s \leq n \leq s + m$,

$$\pi_n = \pi_0 \prod_{i=0}^{s-1} \frac{\lambda_i}{(i+1)\mu} \prod_{i=s}^{n-1} \frac{\lambda}{s\mu} = \pi_0 \left(\frac{\lambda}{\mu}\right)^n \frac{s^s}{s!s^n} = \pi_0 \left(\frac{\lambda}{s\mu}\right)^n \frac{s^s}{s!}.$$




**Slika 2.10:** Dijagram stanja $M/M/s/m$ sustava posluživanja za $s$ zahtjeva u sustavu s $m$ mjesta u spremniku gdje je $K = s + m$.

Vjerojatnost $\pi_0$ možemo izračunati iz izraza:

$$\pi_0 = \frac{1}{\sum_{n=0}^{s-1} \left(\frac{\lambda}{s\mu}\right)^n \frac{s^n}{n!} + \sum_{n=s}^{s+m} \left(\frac{\lambda}{s\mu}\right)^n \frac{s^s}{s!}}.$$

Prvi dio zbroja u nazivniku nije moguće izračunati eksplicitno, ali drugi je:

$$\begin{aligned}
\frac{s^s}{s!} \sum_{n=s}^{m} \left(\frac{\lambda}{s\mu}\right)^n &= \frac{s^s}{s!} \left[ \sum_{n=0}^{m} \left(\frac{\lambda}{s\mu}\right)^n - \sum_{n=0}^{s-1} \left(\frac{\lambda}{s\mu}\right)^n \right] \\
&= \frac{s^s}{s!} \left[ \frac{1 - \left(\frac{\lambda}{s\mu}\right)^{m+1}}{1 - \frac{\lambda}{s\mu}} - \frac{1 - \left(\frac{\lambda}{s\mu}\right)^{s}}{1 - \frac{\lambda}{s\mu}} \right] \\
&= \frac{s^s}{s!} \left[ \frac{\left(\frac{\lambda}{s\mu}\right)^{s} - \left(\frac{\lambda}{s\mu}\right)^{m+1}}{1 - \frac{\lambda}{s\mu}} \right].
\end{aligned}$$

Slijedi pojednostavljeni izraz:

$$\pi_0 = \frac{1}{\sum_{n=0}^{s-1} \left(\frac{\lambda}{s\mu}\right)^n \frac{s^n}{n!} + \frac{s^s s!}{1 - \frac{\lambda}{s\mu}} \left[ \frac{1 - \left(\frac{\lambda}{s\mu}\right)^{m+1}}{1 - \frac{\lambda}{s\mu}} - \frac{1 - \left(\frac{\lambda}{s\mu}\right)^{s}}{1 - \frac{\lambda}{s\mu}} \right]}.$$

Sada želimo izračunati prosječan broj zahtjeva u cijelom sustavu. Ovaj izraz ne možemo izravno izračunati, pa ćemo ovdje dati samo konačan rezultat. Zbog jednostavnosti uvedimo supstitucij $B = \frac{\lambda}{s\mu}$:

$$\begin{aligned}
N &= \sum_{n=0}^{s-1} n\pi_0 B^n \frac{s^n}{n!} + \sum_{n=s}^{m} n\pi_0 B^n \frac{s^s}{s!} \\
&= \sum_{n=0}^{s-1} n\pi_0 B^n \frac{s^n}{n!} + \pi_0 \frac{s^s}{s!} \frac{mB^s + (1-s)B^{s+1} - (m+1)B^{m+1} + kB^{m+2}}{(1-B)^2}.
\end{aligned}$$

Efektivni $\lambda_{\text{ef}}$ se računa na isti način kao i prije:

$$\lambda_{\text{ef}} = \lambda \left[1 - \pi_k\right] = \lambda \left[1 - \pi_0 \left(\frac{\lambda}{s\mu}\right)^m \frac{s^s}{s!}\right].$$

Svaki od $s$ poslužitelja s intenzitetom $\mu$ poslužuje dolazni intenzitet $\lambda_{\text{ef}}/\mu$. S obzirom na to da je prosječan broj zahtjeva u bilo kojem od poslužitelja $\frac{\lambda_{\text{ef}}}{s\mu}$, slijedi da je:

$$\overline{N_s} = \frac{s\lambda_{\text{ef}}}{s\mu} = \frac{\lambda_{\text{ef}}}{\mu}.$$

Iz ovog rezultata slijedi da je:

$$\overline{N_q} = \overline{N} - \frac{\lambda_{\text{ef}}}{\mu}.$$

Jasno je da za bilo kakvu praktičnu primjenu izvedenih jednadžbi potrebno raspolagati s određenim podacima o sustavu posluživanja (intenzitetima dolazaka i posluživanja te kapacitetom sustava).

### 2.6.4. Sustav posluživanja $G/G/s/m$

U sustavu $G/G/s/m$ dolazak zahtjeva za posluživanje i vremena posluživanja su općenito raspodijeljeni. Ta široka klasa raspodjela omogućuje modeliranje mnogo različitijih realnih situacija, ali dolazi s dodatnom složenošću. Za razliku od slučaja $M/M/s/m$, rijetko je moguće dobiti zatvorene oblike za performanse sustava. Za općenitu razdiobu vremena dolazaka i posluživanja, vjerojatnosti $\pi_n$ da se u sustavu nalazi n klijenata i dalje su definirane jednadžbama lokalne ravnoteže, ali one nisu toliko jednostavne za izračunati kao u slučaju $M/M/s/m$. Srednji broj klijenata u sustavu, $\overline{N}$, i dalje je moguće izračunati kao težinsku sumu vjerojatnosti $\pi_n$, ali ove se vjerojatnosti moraju pronaći rješavanjem sustava linearnih jednadžbi umjesto eksplicitne formule. Također, efektivni intenzitet dolazaka $\lambda_{\text{ef}}$ može se razlikovati od nominalnog intenziteta dolazaka $\lambda$ ovisno o raspodjeli vremena dolazaka. Slično tome, srednji broj klijenata koji čekaju, $\overline{N_q}$, može se izračunati kao razlika između ukupnog broja klijenata u sustavu, $\overline{N}$, i srednjeg broja klijenata koje poslužuju poslužitelji, $\overline{N_s}$. Međutim, ove količine općenito nisu jednostavne za izračunavanje i često se moraju odrediti putem simulacija ili aproksimativnih metoda. Unatoč dodatnoj složenosti, $G/G/s/m$ model je izuzetno koristan u situacijama gdje su vremena dolazaka i posluživanja suviše složena da bi se modelirala Markovljevim procesima. S obzirom na to, njegova sposobnost opisivanja široke palete stvarnih sustava čini ga neophodnim alatom za bilo kojeg teoretičara ili praktičara u polju teorije posluživanja. Treba imati na umu da u ovom slučaju nisu izvedene eksplicitne formule za parametre kao što su $\pi_n$, $\overline{N}$ i $\lambda_{\text{ef}}$ jer ovise o specifičnim razdiobama vremena dolazaka i posluživanja koje nisu date. U općenitom slučaju, moglo bi biti potrebno koristiti numeričke metode ili simulacije za određivanje tih parametara.

## 2.7. Primjeri: Sustav posluživanja u zdravstvu

### Sustav posluživanja za zakazivanje sastanaka u bolnici

Ovaj scenarij se može modelirati kao $M/M/s/m$ sustav posluživanja gdje su klijenti pacijenti koji zovu, poslužitelji su operateri koji primaju pozive, a 'sustav' je telefonska služba za zakazivanje sastanaka. Pretpostavlja se da su vremena dolazaka poziva (pacijenata) modelirana eksponencijalnom distribucijom (što ukazuje na proces Poissonovog dolaska), kao i vremena posluživanja (vrijeme koje je potrebno da operater obradi poziv). Koristeći teoriju posluživanja, moguće je napraviti procjene o optimalnom broju operatera, vjerojatnosti da pacijent mora čekati na liniji prije nego što bude poslužen, ili ukupnom broju poziva koji se mogu obraditi u određenom vremenskom periodu. Na primjer, ako bolnička služba za zakazivanje sastanaka zna da će primati povećani broj poziva tijekom određenih vremenskih razdoblja (npr. prvi dan u mjesecu), moguće je koristiti model sustava posluživanja kako bi se bolje rasporedili operateri i drugi resursi kako bi se mogli nositi s povećanom potražnjom. Ovaj pristup može pomoći u smanjenju vremena čekanja pacijenata na liniji, poboljšanju iskustva pacijenata i povećanju učinkovitosti bolničkog osoblja.

### Sustav posluživanja za bolnički odjel hitne pomoći

U ovom kontekstu, pacijenti dolaze s različitim intenzitetima tijekom dana, a vremena posluživanja (vrijeme koje pacijent provede u hitnoj pomoći) također mogu biti vrlo različita, ovisno o specifičnim medicinskim potrebama pacijenta. Tako, ovo se može modelirati kao $G/G/s/m$ sustav posluživanja, gdje su 'klijenti' pacijenti, poslužitelji su liječnici ili medicinske sestre, a 'sustav' je hitna služba. Korištenjem teorije posluživanja, moguće je napraviti predikcije o potrebnoj veličini osoblja, vjerojatnosti da pacijent mora čekati na liječenje, ili ukupnom broju pacijenata koji mogu biti u hitnoj službi u određeno vrijeme. Ovi podaci mogu pomoći bolničkom menadžmentu u planiranju resursa i poboljšanju kvalitete skrbi. Primjerice, ako bolnica zna da ima povećani intenzitet dolazaka pacijenata tijekom određenih vremenskih razdoblja (npr. vikendom ili tijekom sezone gripe), moguće je koristiti model sustava posluživanja kako bi se bolje rasporedilo osoblje i resursi kako bi se moglo nositi s povećanom potražnjom. Unatoč svojoj složenosti, ovaj pristup može pružiti vrlo korisne uvide za bolničko osoblje i pomoći u poboljšanju kvalitete pacijentske skrbi.

# 3. Programska implementacija

## 3.1. Korištene tehnologije

Ovaj projekt, usmjeren na simulaciju teorije posluživanja, koristi Flutter, suvremeni okvir za razvoj, poznat po mogućnosti izrade naprednih korisničkih sučelja na različitim platformama. Programski jezik koji Flutter koristi je Dart, istaknut svojom efikasnošću, jednostavnošću koda i robusnošću tipova. Odabir Fluttera utemeljen je na njegovoj široko prihvaćenoj upotrebi, velikoj zajednici korisnika i obuhvatnoj dokumentaciji koja podržava razvoj simulacija i vizualizacija. Flutter nudi pristup razvoju aplikacija putem sastavnih dijelova, poznatijih kao widgeti, koji se mogu slobodno kombinirati u svrhu izrade složenih korisničkih sučelja. Tijekom izrade ovog projekta, iskorišten je niz preddefiniranih widgeta, kao i nekoliko prilagođenih, u svrhu postizanja specifičnih estetskih i funkcionalnih zahtjeva. Simulacija teorije posluživanja oslanja se na nekoliko Dart-ovih biblioteka, koje su specificirane unutar datoteke pubspec.yaml:

- `charts_flutter`: za prikaz grafičkih podataka.
- `syncfusion_flutter_charts`: za naprednije grafičke prikaze i analizu podataka.
- `rxdart`: za asinkrono programiranje i upravljanje tokovima podataka.
- `flutter_spinkit`: za dodavanje animacija tijekom učitavanja podataka.
- `awesome_dialog`: za prikazivanje prilagodljivih dijaloških okvira.
- `math_expressions`: za evaluaciju i manipulaciju matematičkim izrazima.
- `eval_ex`: za dodatne mogućnosti evaluacije izraza.
- `flutter_tex`: za prikaz matematičkih izraza u LaTeX formatu.

## 3.2. Prikaz programskog koda

Ovaj dio teksta se bavi programskim rješenjem za simulaciju koje upravlja s procesima prijave i odjave korisnika u sustavu koji uključuje čekanje u redovima. Glavna funkcija, koju smo nazvali *simulate()*, odgovorna je za praćenje vremena dolaska i odlaska korisnika te nadgleda sekvencu događaja.

```
void simulate() {
_serviceTime = _randomServiceTime();
  if (_numCustomersInQueue > 0) {
        _timeOfNextArrival += _serviceTime;
  } else {
    _timeOfNextArrival += double.infinity;
  }
  double nextEventTime;
  if (_timeOfNextArrival < _timeOfNextDeparture) {
    nextEventTime = _timeOfNextArrival;
  } else {
    nextEventTime = _timeOfNextDeparture;
  }
  _currentTime = nextEventTime;
  if (_timeOfNextArrival < _timeOfNextDeparture) {
    handleArrival();
  } else {
    handleDeparture();
  }
}
```

Funkcija pod nazivom *handleArrival()* upravlja procesima pristupa korisnika, te ažurira broj korisnika na čekanju. U situaciji kada su servisi dostupni, korisnik će biti poslužen, a vremenski interval do sljedećeg izlaska će biti ažuriran. S druge strane, ako servisi nisu dostupni, korisnik će biti stavljen na listu čekanja.

```
void handleArrival() {
      if (_numCustomersInQueue < queueCapacity) {
    if (numServers > 0) {
      numServers -= 1;
      _timeOfNextDeparture = _currentTime + _serviceTime;
    } else {
      _numCustomersInQueue += 1;
```

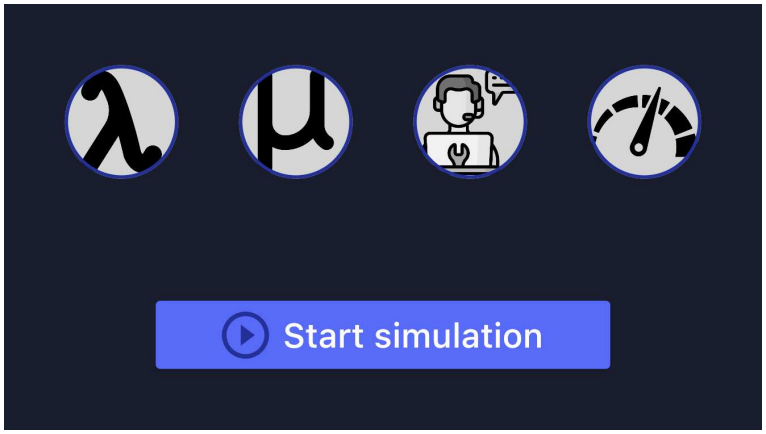
```
      }
    } else {
      _numCustomersServed += 1;
    }
    _timeOfNextArrival = _currentTime + _randomArivalTime();
  }
```

Zauzvrat, funkcija po imenu *handleDeparture()* kontrolira procese odlazaka korisnika, ažurirajući broj posluženih korisnika, kao i broj korisnika koji čekaju. Ako postoji lista čekanja, jedan od korisnika će biti poslužen, a vremenski okvir do sljedećeg odlaska će biti ažuriran.

```
  void handleDeparture() {
        _numCustomersServed += 1;
      if (_numCustomersInQueue > 0) {
        _numCustomersInQueue -= 1;
        _timeOfNextDeparture = _currentTime + _serviceTime;
      } else {
        numServers += 1;
        _timeOfNextDeparture = double.infinity;
      }
  }
```

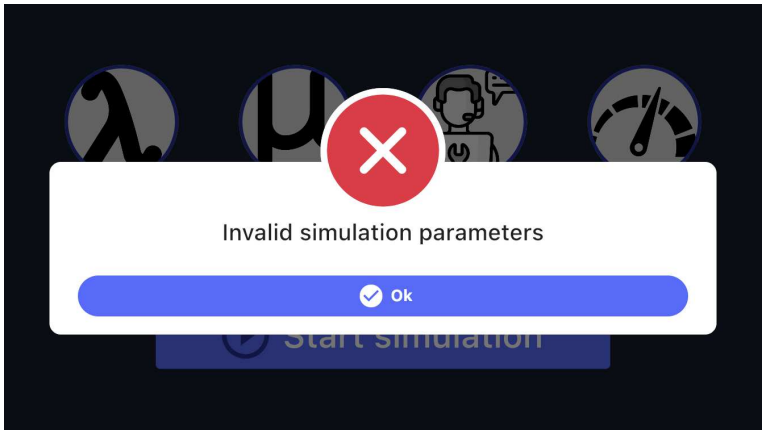
## 3.3. Korisničko sučelje

### 3.3.1. Početni zaslon




**Slika 3.1:** Početni zaslon

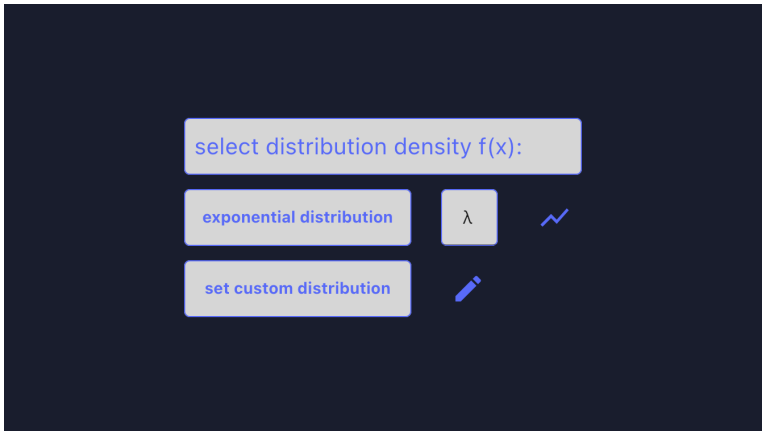
S početne zaslonske stranice sučelja, čiji je prikaz dostupan na slici, može se uočiti četiri ikone zajedno s gumbom *Start simulation*. Ikone simboliziraju parametre sustava na sljedeći način: dvije od njih označavaju funkcije gustoće parametara $\lambda$ i $\mu$, dok preostale dvije služe kao reprezentacije broja poslužitelja $s$ i kapaciteta spremnika $m$. Gumb za pokretanje simulacije je zaslužan za simuliranje sustava uz pretpostavku da su svi parametri poznati. U suprotnom, ako korisnik pokuša započeti simulaciju bez prvotno postavljanja svih parametara, program ga na to upozorava te korisniku signalizira grešku pomoću iskočnog prozora. Korisnik za postavljanje parametra klikne na odgovarajuću ikonu te mu se otvara prozor za definiranje parametra koji je odabrao. Nakon što je parametar definiran, korisnika se vraća na početni zaslon i ikona odabranog parametra poprima boju.




**Slika 3.2:** korisnik pokreće simulaciju bez definiranja svih parametara

### 3.3.2. Definiranje parametara

Na zaslonu se mogu uočiti dvije opcije za unos funkcije gustoće: izravni unos parametra $\lambda$ ili zadavanje općenite funkcije gustoće.

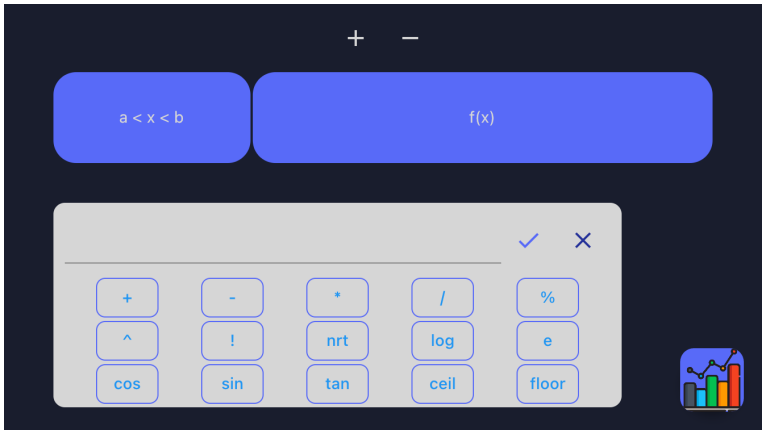



**Slika 3.3:** Slika zaslona s ponudenim opcijama za definiranje parametara

Ako se odlučimo za prvu opciju, trebamo unijeti numeričku vrijednost u polje za parametar $\lambda$. Potvrda unosa vrši se pritiskom na ikonu u obliku emotikon slike. Ako pak želimo odabrati općenitu funkciju gustoće, potrebno je pritisnuti ikonu u obliku emotikon olovke. Time će nas sustav preusmjeriti na novi zaslon koji je posebno osmišljen za unos proizvoljne funkcije."

### 3.3.3. Definiranje funkcije gustoće

Na vrhu ekrana, nalaze se '+' i '-' gumbi, a u donjem desnom kutu vidljiva je ikona grafa. Detaljnije informacije o funkcionalnosti ovih elemenata nalaze se u nastavku. Pritiskom na gumb '+' stvara se *slot* koji se sastoji od intervala "$a < x < b$" i funkcije $f(x)$. Klikom na bilo koji dio *slot-a* otvara se priručnik za definiranje odgovarajućih funkcija. Klikom na lijevu stranu *slot-a* otvara se prozor za definiranje intervala pomoću integrirane tipkovnice. Interval definiramo tako da zadamo numeričke vrijednosti $a$ i $b$ i relaciju s varijablom $x$, interval bi trebao imati oblik $a < x < b$, gdje *parser* za relaciju prihvaća znakove $<$, $\leq$. Klikom na desnu stranu *slot-a* otvara se priručnik za definiranje funkcije $f(x)$. Kao i u prethodnom slučaju, funkciju definiramo pomoću tipkovnice, ali pri definiranju funkcije korisniku je na raspolaganju veći broj tipki. Naime, korisniku je omogućeno zadavanje funkcija koje se često korištene matematičke funkcije. Najbolja ilustracija bi bila na primjeru u kojem korisnik želi koristiti logaritamsku funkciju. No kako će korisnik znati koja je sintaksa za logaritamsku funkciju? Upravo u tu svrhu služi priručnik. Naime, priručnik sadrži popis svih podržanih funkcija koje se mogu zadavati na ovaj način. Pritiskom na odgovarajuću funkciju, korisniku se u obliku *pop-up* prozora prikazuje opis s kratkim primjerom.



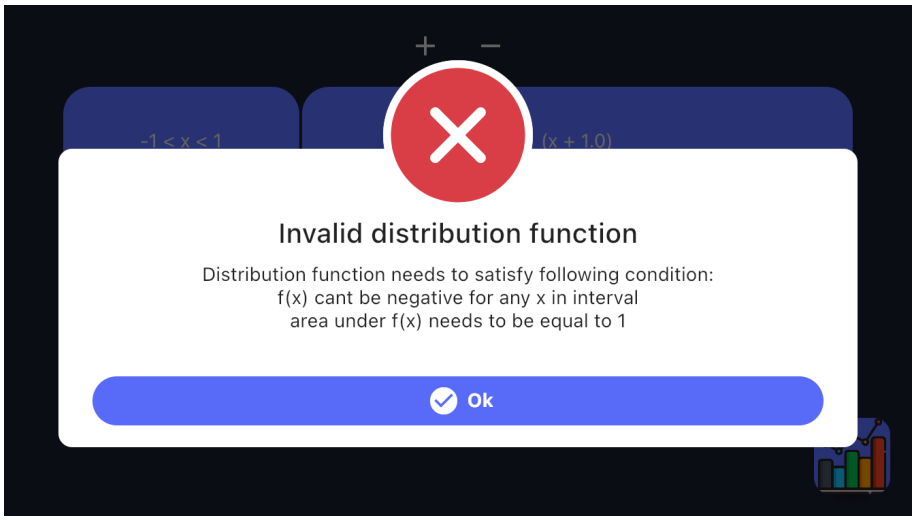

**Slika 3.4:** Definiranje funkcije gustoće

Sada kada znamo kako definirati interval i funkciju koja će vrijediti na tom intervalu, željeli bismo našim korisnicima omogućiti da koriste različite funkcije na različitim intervalima, to jest omogućiti im da naprave kompozitnu funkciju $f(x)$ od proizvoljnog broja funkcija $f_1(x), f_2(x), ..., f_n(x)$ na proizvoljnom broju intervala $I_1, I_2, ..., I_n$. Na primjer, želimo da na intervalu $x_1 < x < x_2$ funkcija $f(x)$ bude jednaka $x + 1$, a na intervalu $x_2 < x < x_3$ funkcija $f(x)$ bude jednaka $x^2$. Kako bi to postigli našim korisnicima omogućili smo da mogu dodati proizvoljan broj slotova. Na primjer, u našem slučaju korisnik bi dodao dva slot-a. Prvi slot bi definirao interval $x_1 < x < x_2$ i funkciju $x + 1$, a drugi slot bi definirao interval $x_2 < x < x_3$ i funkciju $x^2$.


**Slika 3.5:** Primjer kompozitne funkcije gustoće

Jednom kad je korisnik definirao funkciju gustoće, željeli bismo mu pokazati kako izgleda funkcija koju je definirao. Ovaj zadatak obavlja ikona grafa koja se nalazi u donjem desnom kutu. Pritiskom na ikonu grafa, korisniku se prikazuje graf funkcije gustoće koju je definirao. No primijetimo da postoji mali problem s tim, naime skup funkcija koje korisnik može unijeti beskonačan i nitko nam ne garantira da će korisnik unijeti funkciju koja zadovoljava kriterije funkcije gustoće. Stoga smo se odlučili na sljedeće rješenje. Ako je korisnikova funkcija gustoće ispravna, tada će korisnik omogućiti daljini rad s aplikacijom. U suprotnom korisniku će se prikazati poruka o pogrešci i blokirati daljini rad s aplikacijom sve dok ne unese ispravnu funkciju. Prisjetimo se ispravna funkcija gustoće mora zadovoljavati sljedeće kriterije:

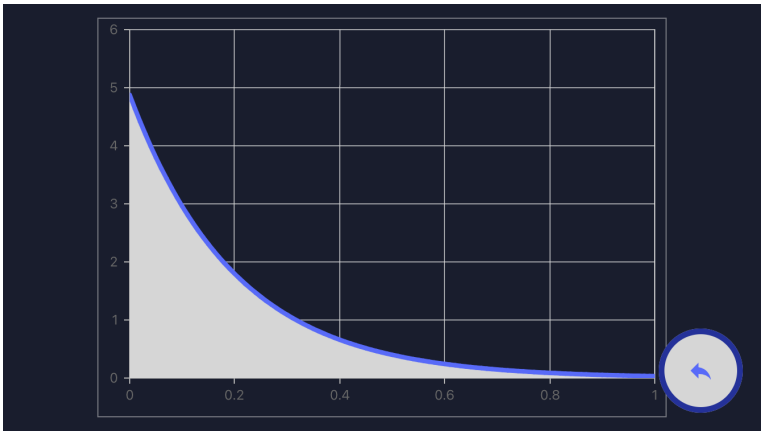
- $f(x) \geq 0$ za sve $x$ u definicijskom području funkcije,
- $\int_{-\infty}^{+\infty} f(x)dx = 1$.




**Slika 3.6:** Neispravna (kompozitna) funkcija gustoće

### 3.3.4. Prikaz grafa

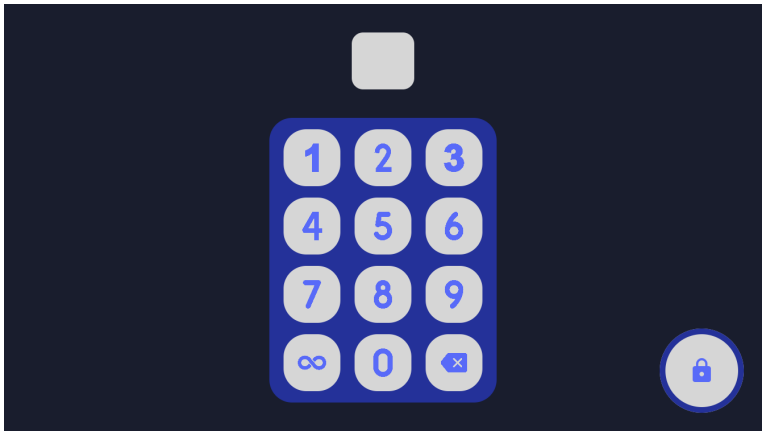
Nakon ispravnog unosa parametara na neki od prethodnih načina, korisniku se otvara novi zaslon s prikazom grafa. Graf iscrtava funkciju koju je korisnik definirao, bilo putem parametra $\lambda$ ili proizvoljne funkcije. Funkcija je prikazana ljubičastom linijom na tamnoj pozadini, dok je površina ispod grafa obojena bijelom bojom radi boljeg kontrasta za korisnika. Korisnik može pomicati po grafu lijevo i desno pomicanjem prsta po ekranu. Pritiskom na gumb u donjem desnom kutu korisnik se vraća na početni zaslon, gdje bi odabrana ikona trebala biti ispunjena bojom.




**Slika 3.7:** Prikaz grafa funkcije gustoće $f(x) = \lambda e^{-\lambda x}$ za $\lambda = 5$

### 3.3.5. Numerički unos

Numerički unos je zaslon koji se otvara korisniku kada želi zadati broj poslužitelja ili kapacitet servera. Na zaslonu se nalazi jednostavan kalkulator u kojem korisnik može unijeti brojeve od 1 do 9999, dok se sve veće vrijednosti tretiraju kao $\infty$ (primijetite da korisnik ne može unijeti broj 0 za prvu znamenku, budući da takva implementacija nema smisla). Kalkulatorov display s bijelom pozadinom automatski se širi kako korisnik unosi brojeve, a unosom prevelikog broja korisnikov unos se tretira kao beskonačnost. Također, korisnik može unijeti vrijednost beskonačno usred upisivanja broja, što rezultira cjelokupnim rezultatom kao beskonačnim brojem. Potvrda unosa korisnika odvija se u dva dijela. Prvi dio je klik na gumb u donjem desnom dijelu zaslona (*lock*) kako bi se zaključala vrijednost rezultata. Vrijednost rezultata ne može biti zaključana dok korisnik ne unese vrijednost veću od 0. U stanju zaključavanja, donji desni gumb mijenja ikonu u povratnu strelicu. U tom stanju, pritiskom na bilo koju tipku osim gumba na dnu zaslona, korisnik ulazi u stanje unosa i omogućuje se ponovni unos, pri čemu je ponovno potrebno zaključati unos. Pritiskom na donji gumb korisnik se vraća na početni zaslon.




**Slika 3.8:** Kalkulator za numerički unos parametara broja poslužitelja i kapaciteta sustava

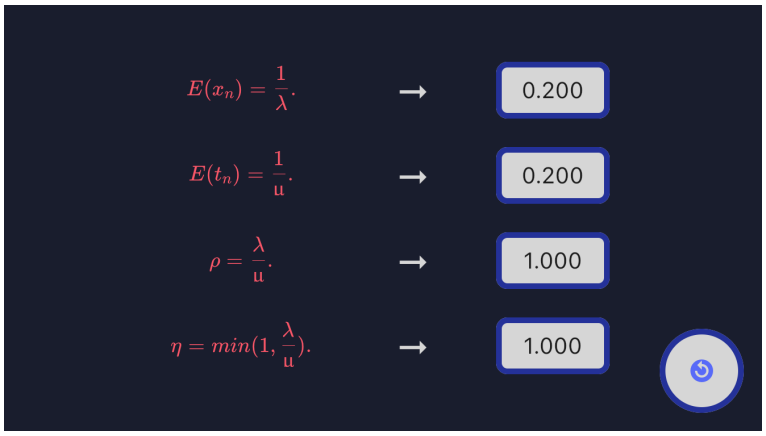
### 3.3.6. Simulacija u stvarnom vremenu

Klikom na gumb *Start simulation*, korisniku se prikazuje zaslon s grafom koji ima vrijeme $t$ na $x$-osi i broj jedinica $N(t)$ na $y$-osi, što predstavlja broj jedinica u sustavu za zadane početne parametre. Na tom zaslonu koristi se glavni algoritam aplikacije opisan ranije. U donjem dijelu zaslona nalaze se dva gumba: *Play* gumb je odgovoran za pokretanje same simulacije. Nakon pritiska na taj gumb, on se mijenja u gumb za zaustavljanje, što znači da korisnik može zaustaviti simulaciju u bilo kojem trenutku. Simulacija se može nastaviti nakon što je zaustavljena. Jednom kada je simulacija pokrenuta, korisnik treba obratiti pažnju na ljubičastu vertikalnu liniju koja predstavlja broj servera i bijelu liniju koja predstavlja broj jedinica u sustavu. Važno je napomenuti da je broj jedinica u sustavu ograničen zbrojem broja servera i kapaciteta, jer se sve dodatne pridošle jedinice odbijaju. Pritiskom na tipku *Results* otvara se novi zaslon koji prikazuje izračunate rezultate simulacije koji pratite ponašanje grafa u beskonačnost.


**Slika 3.9:** Primjer prikaz simulacije

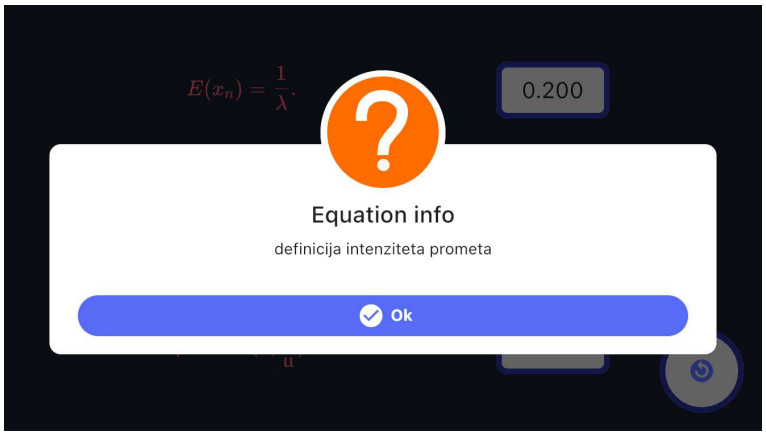
### 3.3.7. Rezultati simulacije

Ovaj zaslon prikazuje općenite parametre odabranog sustava, a rezultati koji se prikazuju uključuju slijedeće: prosječno vrijeme između dolazaka korisnika u sustav, označeno s $E(\tilde{t})$, koje je jednako $\frac{1}{\lambda}$, prosječno vrijeme obrade korisnika u sustavu, označeno s $E(\tilde{x})$, koje je jednako $\frac{1}{\mu}$, intenzitet prometa, označen s $\rho$, koji je jednako $\frac{\lambda}{\mu}$, te efektivna intenzitet prometa, označen s $\eta$, koji je jednako $\min\{1, \frac{\lambda}{\mu}\}$.




**Slika 3.10:** Primjer prikaza rezultata simulacije

Klikom na strelicu iza svakog rezultata, korisnik dobije *pop-up* ekran s definicojom varijable.




**Slika 3.11:** *Pop-up* ekran s definicijom varijable $\rho$

Korisniku se nudi mogućnost ponovnog pokretanja ovog procesa s novim zadanim parametrima pritiskom na donju desnu ikonu za ponovno pokretanje (*restart*).

# Zaključak

U ovom završnom radu razvijena je aplikacija za simulaciju sustava posluživanja pruža korisnicima intuitivno sučelje za definiranje parametara sustava i prikaz rezultata simulacije. Korisnicima se omogućuje odabir funkcije gustoće, definiranje intervala i funkcije, te numerički unos parametara kao što su broj poslužitelja i kapacitet servera. Grafički prikaz rezultata simulacije pruža korisnicima uvid u promjene broja jedinica u sustavu tijekom vremena, uz mogućnost pokretanja i zaustavljanja simulacije. Osim toga, korisnici mogu pregledati rezultate simulacije na zasebnom zaslonu. Uz intuitivno sučelje i mogućnost ponovnog pokretanja simulacije s novim parametrima, aplikacija omogućuje korisnicima bolje razumijevanje i analizu sustava posluživanja. Također, primjena kriterija za pravilnu funkciju gustoće osigurava da korisnici unose valjane parametre. Sveukupno, ova aplikacija pruža korisnicima alat za simulaciju sustava posluživanja te olakšava istraživanje, analizu i donošenje informiranih odluka u kontekstu teorije posluživanja.

# LITERATURA

N. Elezović and L. Mihoković. *Stohastički procesi*. FER, 2021. Skripta.

Linda Green. Queueing theory and modeling. 2023. URL `https://www0.gsb.columbia.edu/mygsb/faculty/research/pubfiles/5474/queueing%20theory%20and%20modeling.pdf`.

Carlos M. Hernandez-Suarez, Carlos Castillo-Chavez, Osval Montesinos Lopez, and Karla Hernandez-Cuevas. An application of queueing theory to sis and seis epidemic models. *Mathematical Biosciences and Engineering*, 7(4):809–823, 2010.

Peter J. Kolesar, Kenneth L. Rider, Thomas B. Crabill, and Warren E. Walker. A queueing-linear programming approach to scheduling police patrol cars. *Operations Research*, 23:1045–1062, 1975.

Houda Mehri, Djemel Taoufik, and Hichem Kammoun. Solving of waiting lines models in the airport using queuing theory model and linear programming: The practice case. Technical report, A.I.M.H.B., 2006. Preprint.

Flutter packages. *Zadnji pristup: 7.6.2023*, 2023. URL `https://pub.dev/`.

## Simulacija i primjena modela teorije posluživanja


### Sažetak

U sklopu ovog završnog rada razvijena je aplikacija za simulaciju sustava posluživanja. Aplikacija korisnicima pruža intuitivno sučelje za definiranje parametara sustava i prikaz rezultata simulacije. Korisnicima se omogućuje odabir funkcije gustoće, definiranje intervala i funkcije, te numerički unos parametara kao što su broj poslužitelja i kapacitet servera. Grafički prikaz rezultata simulacije pruža korisnicima uvid u promjene broja jedinica u sustavu tijekom vremena, uz mogućnost pokretanja i zaustavljanja simulacije. Osim toga, korisnici mogu pregledati rezultate simulacije na zasebnom zaslonu. Uz intuitivno sučelje i mogućnost ponovnog pokretanja simulacije s novim parametrima, aplikacija omogućuje korisnicima bolje razumijevanje i analizu sustava posluživanja. Također, primjena kriterija za pravilnu funkciju gustoće osigurava da korisnici unose valjane parametre.



**Ključne riječi:** Teorija posluživanja, Flutter, mobilna aplikacija, simulacija


## Mobile application for simulating queuing systems


### Abstract

As part of this thesis, an application for simulating a queuing system was developed. The application provides users with an intuitive interface for defining system parameters and displaying simulation results. It enables users to choose the probability density function, define intervals and functions, and input numerical parameters such as the number of servers and server capacity. The graphical representation of simulation results gives users insight into changes in the number of units in the system over time, with the possibility of starting and stopping the simulation. In addition, users can view simulation results on a separate screen. With its intuitive interface and the ability to restart the simulation with new parameters, the application enables users to better understand and analyze the queuing system. Also, the application of criteria for the correct probability density function ensures that users enter valid parameters.



**Keywords:** Queuing theory, Flutter, mobile application, simulation