

I фаза пројекта игре Покер

Предмет Архитектура и пројектовање софтвера

Јана Здравковић 15482

Никола Илић 15119

1. Функционални захтеви

Основна функционалност овог система је могућност да више играча игра игру покер. Игра се игра на столовима са најмање 2 играча а највише 5 играча. Корисник у сваком тренутку може да промени сто за којим игра при чему у једном тренутку може играти само за једним столом. Жетони освојени на столу улазе у укупну своту жетона које корисник поседује и сума жетона се може видети у сваком тренутку.

2. Нефункционални захтеви

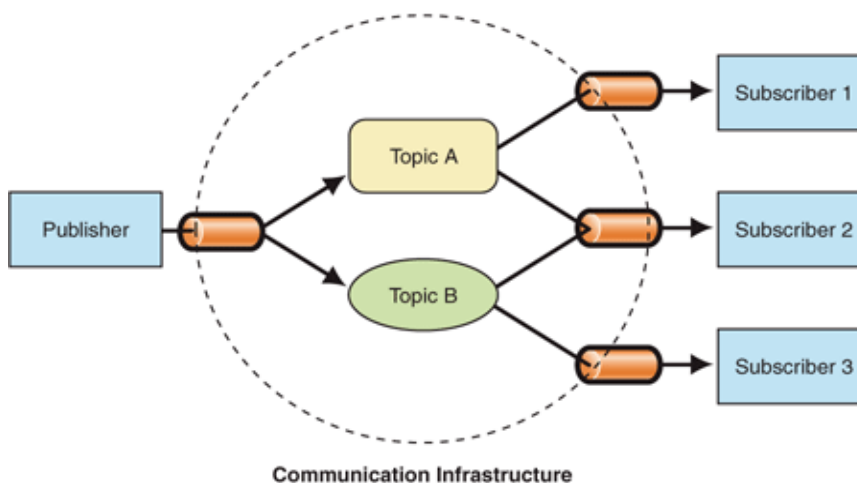
Систем треба да подржи велики број играча – Скалабилност.

Модификабилност, лакоћа промена – систем треба да може да одговори на промене корисничких захтева у вези функционалности (другачија правила играња покера, друге карте, другачији кориснички интерфејс, итд.).

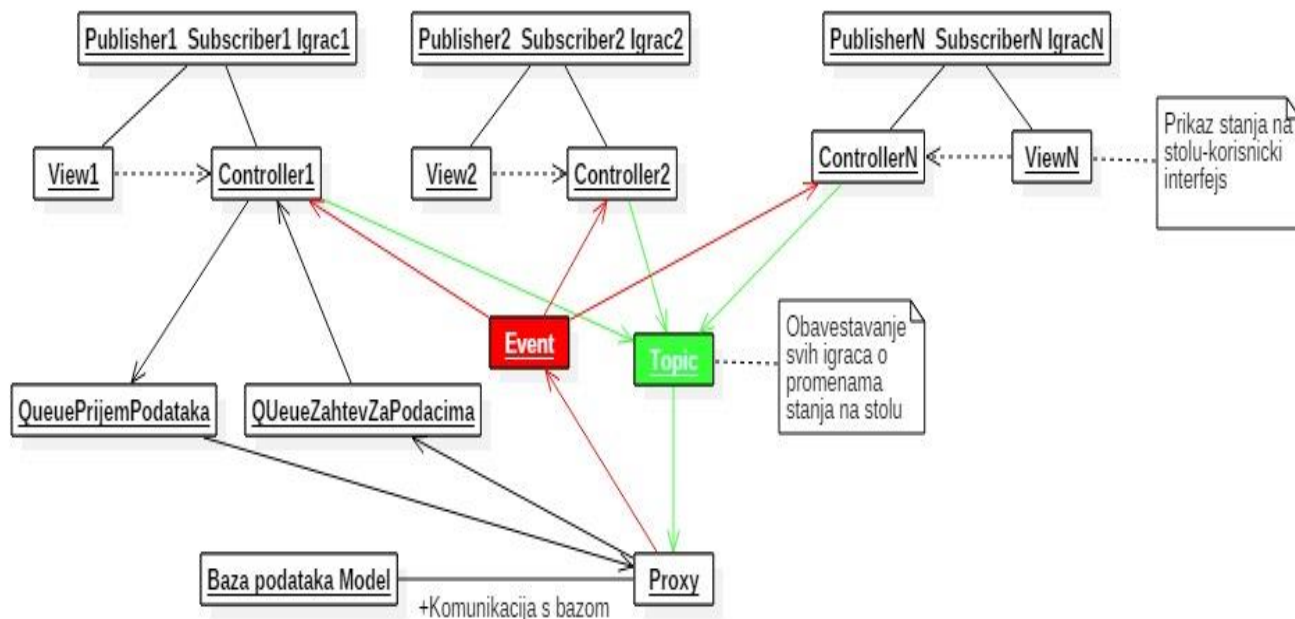
Перформансе – време одзива треба да буде мало обзиром да се ради о интерактивној апликацији.

3. Архитектурни патерни

Клијентска апликација биће организована уз помоћ MVC обрасца – Model-View-Controller. Комуникација између клијената реализоваће се Publish-Subscribe обрасцем. Дистрибуција комуникације између клијената биће реализована уз помоћ Proxy-ја.



Слика 1



Слика 2

Апликација ће бити организована као MVC , View ће да зависи од контролера, а преко контролера ће се извршавати логика апликације. Контролер ће стога бити компонента која учествује у комуникацији са Proxy-јем. Proxy врши дистрибуцију комуникације између свих играча. Сваки играч је и publisher i subscriber. Када играч изврши неку акцију у току игре (повећа улог, одустане или прати) обавештавају се сви остали играчи за столом о тој акцији. Модел представља база података која је централизована и повезана је са проху-јем. Уколико су неком играчу потребни подаци из базе контролер тог играча шаље захтев проху-ју за тим подацима који извлачи тражене податке и враћа их контролеру одређеног играча.

4. Спецификација библиотека и програмских оквира

Апликација ће бити писана у C# у framework-u .NET. Комуникација ће бити реализована преко RabbitMQ. База ће бити направљена у MySQL-у , а комуникација са њом преко ORM mapper-a FluentNHibernate.