**6. Arhitektura i dizajn sustava**

*6.1. Svrha, opći prioriteti i skica sustava*

Izbor odgovarajuće arhitekture programske potpore bitan je korak u samom oblikovanju sustava jer ona predstavlja most između zahtjeva koji su postavljeni na sustav i implementacije sustava. O izboru arhitekture programske potpore ovisi cijela programska potpora, jer dobra arhitektura znači fleksibilnost sustava, dobru prenosivost, laganu mogućnost nadogradnje i jeftino održavanje.

Budući da je zadatak ovog programskog sustava napraviti aplikaciju za turističku zajednicu, koju će koristiti korisnici radi lakšeg uvida u rezervaciju soba i apartmana, web aplikacija se čini kao optimalan izbor. Za razliku od desktop aplikacije, web aplikacija je puno lakša za korištenje, održavanje i nadogradnju (nije potreban sustav nadogradnje aplikacije). Isto tako, radit će na svakoj platformi i sustavu koji ima standardni web preglednik, čime smanjujemo troškove jer nije potreban razvoj za više platformi. Web aplikacija se brine za komunikaciju izmeñu zahtjeva korisnika i baze podataka koja se nalazi na poslužitelju.

Arhitektura našeg sustava sastoji se od sljedećih podsustava:

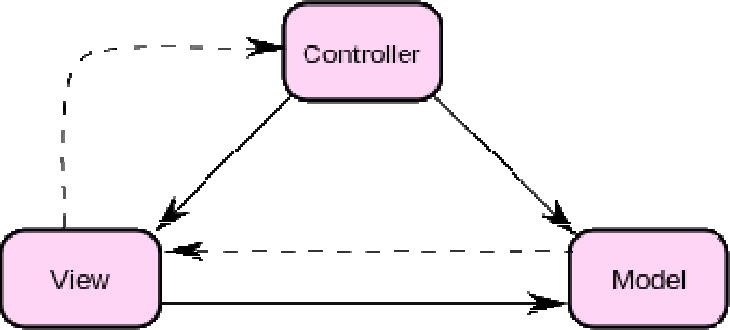
1. Web aplikacija

2. Baza podataka

***Web aplikacija***

Web aplikacija je prozor u naš programski sustav. Ona omogućava prikaz svih podataka iz baze na lijep, strukturiran i siguran način. Pomoću nje se vrši gotovo sva administracija sustava i razmjena podataka.

Za izradu aplikacije smo odabrali MVC (engl *Model – View - Controller*) arhitekturu zato jer omogućava nezavisan razvoj, ispitivanje i održavanje određene aplikacije. Kao jezik u kojem ćemo kodirat izabrali smo javu.



Slika MVC koncept – povezanost dijelova

MVC se sastoji od:

**Model** – predstavlja podatke neke aplikacije. U konkretnoj implementaciji modela(ASP.Net MVC) postoje dvije vrste modela. Jedan prema bazi – kojim se komunicira s bazom, a drugi prema pogledu (View) - koji prenosi podatke do html predloška na kojem se prikazuju

**View** - prikazuje prezentaciju prethodno modeliranih podataka. Konkretno htmlpredložak koji se puni pomoću Razor stroja (engl. *engine*)

**Controller** – služi za upravljanje korisničkim zahtjevima