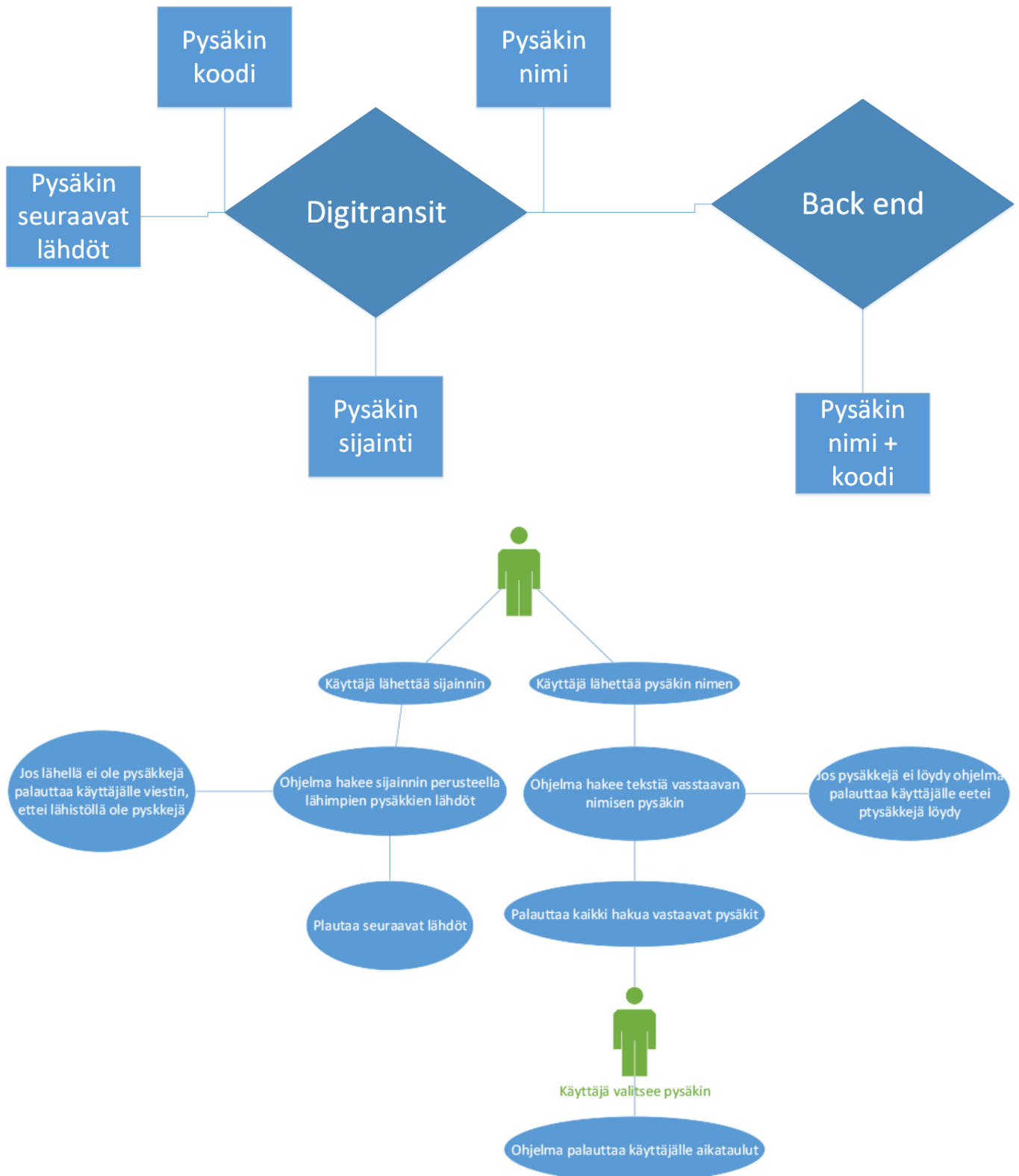


# **Pysäkkibot**

## Laskentalogiikka

## Laskentalogiikka

### Sovelluksen kaaviot



## Kuvaus

Sovellukseen on toteutettu pysäkin etsiminen ja tulosten tulostaminen. Pysäkkejä voi etsiä joko tekstillä tai lähettämällä sijainnin sovellukselle, jonka jälkeen sovellus tulostaa lähellä olevat lähdöt. Keskeneräisiä toimintoja ovat paikannuksen perusteella saatujen tietojen korjaus ja pysäkkihaulla saatujen pysäkkien lähtöjen saaminen.

Koodaamme myös suodattimen joka poistaa olemattomat lähdöt tuloksista.

Alla olevassa koodipätkässä sovellus prosessoi sijainnin perusteella saadusta tuloksista jokaisen eri lähdön kellonajan, numeron tai kirjaimen, määränpään ja pysäkin koodin jolta lähtö lähtee. Koodi on vielä keskeneräinen ja siinä on vielä korjattavaa, mutta esimerkki siitä mitä työistämme.

```
//Uus for looppi
    for (i = 0; i < stoptimes.length; i += 1) {
        var stoptimes1 = JSON.stringify(stoptimes[i])
        //Jos "haamuvuoro / linja"
        if (stoptimes1 == "[]") {
            console.log("Hypätty yli");
            //Älä tee mitään muuta
        } else {
            var locVastaus1 = realtimedep[i]
            console.log(locVastaus1)
            if(locVastaus1 == undefined) {
                //Do nothing
            } else {
                //Muuttaa sekunnit tunneiksi ja minuuteiksi
                var aika = TimeFormat.fromS(locVastaus1, 'hh:mm');
                var aika2 = limit(aika, 5);
                console.log(aika2)
            }
            //Yhdistää ajan ja määränpään
            var locVastaus2 = aika2+" "+bNumero[i]+" "+headsign[i]+" -
"+pCode[i]+"\\n";

            console.log("Yhdistää ajan ja määränp: " + locVastaus2)
            if(lahdot == null){
                lahdot = locVastaus2;
                console.log("Tyhjään lahtöön lisäys")
            }else{
                console.log("Lahtöön lisäys")
                lahdot = lahdot + locVastaus2;
                //console.log(lahdot)
            }
        }
    }
}
```