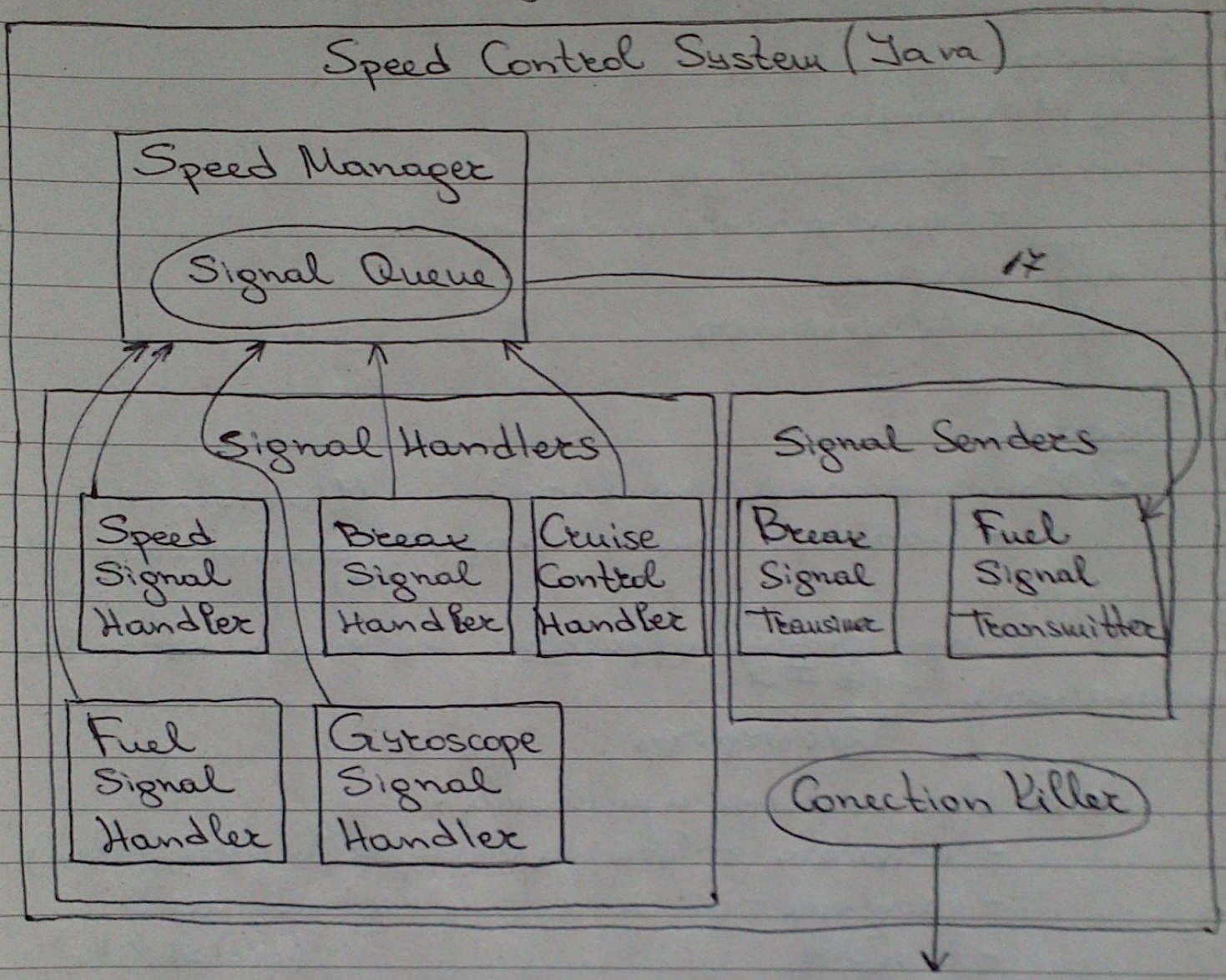
САРС – 29.06.2013г.

Структури
1) Модулни
   - декомпозиция
   - структура на употреба
2) Процесни
3) Разположението

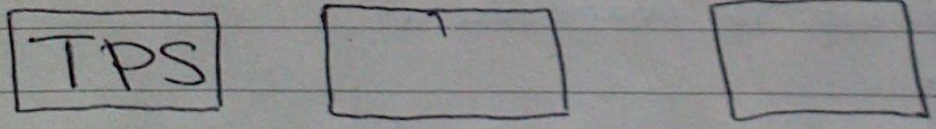
Задача:
- Функционални изисквания:
  - шофьора задава скорост → (1) автопилот / (2) алгоритъм
  - при спускане / качване системата отразява промяната → (1) I/O / (2) алгоритъм
- Нефункционални изисквания:
  - системата трябва да е бърза (бързодействие)
  - хардуера – да може да се подменя лесно (portability)
  - interface:
    - входове
      - *датчик за текуща скорост
      - *сензор за спирачката
    - изходи
      - *механизъм за задействане на спирачките
      - *клапан за подаване на гориво

! *датчик за текущо подаване на гориво
! *датчик за наклон

• Декомпозиция на модули:



може да е отделна



- T_1 ! Бързи алгоритми
  * C++; За embedded system C++ е по-добро
  ! Многонишково приложение – паралелни
  * Queue

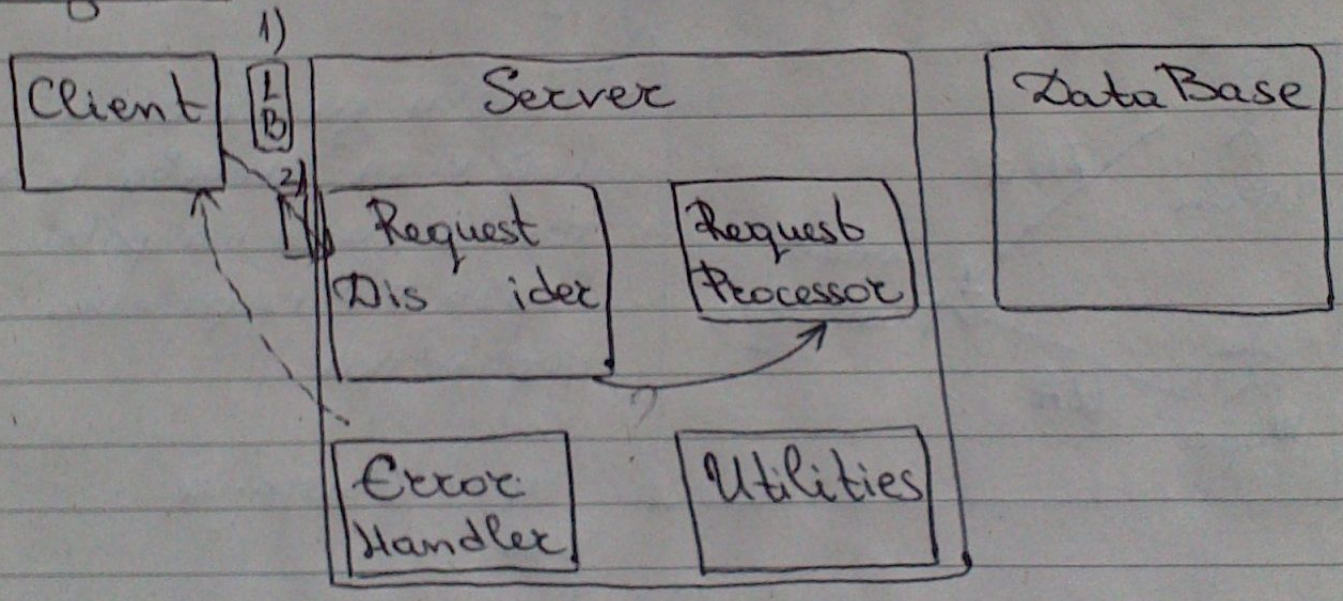
! NF Reliability

1. Speed Signal Handler извиква Speed Manager-а и му подава информация за текущата скорост

interface: currentspeed double;
Speed Signal Handlere{
:
speedManager. setCurrentSpeed(curSpeed:double)
:
}

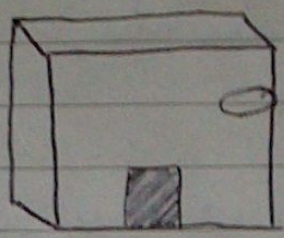
17. Speed Manager-а комуникира с Fuel и му подава следните аргументи:
- throublevel: double

Задача:

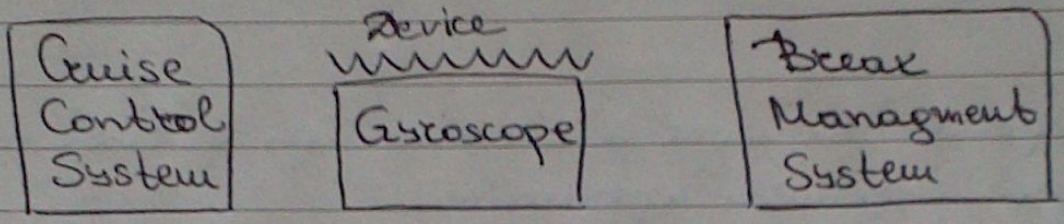


a) FT - fault tolerance
1. LB - load balancer
! 2. Queue (20 requests) - реже заявките

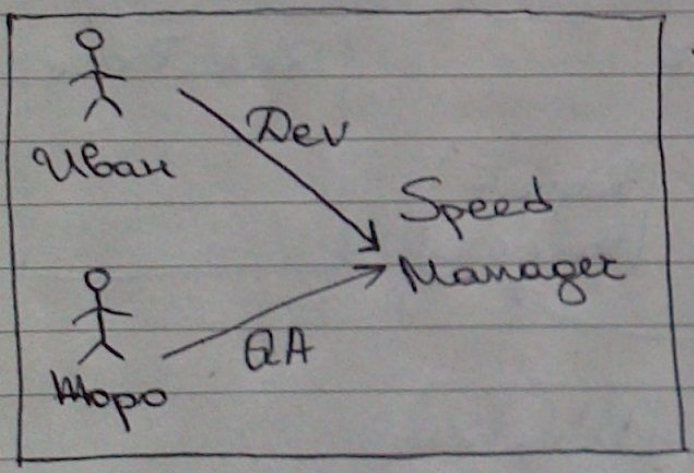
Deployment: Колите



Speed Control System

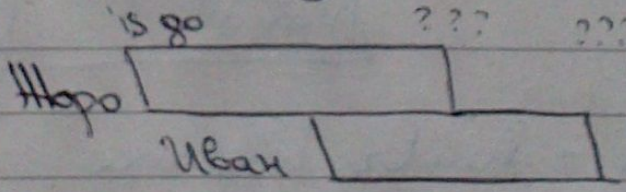


WAD - work assignment diagram



- Трябва да се опише кой кога започва и ако трябва да се направи бързо => паралелна обработка

Grant диаграма

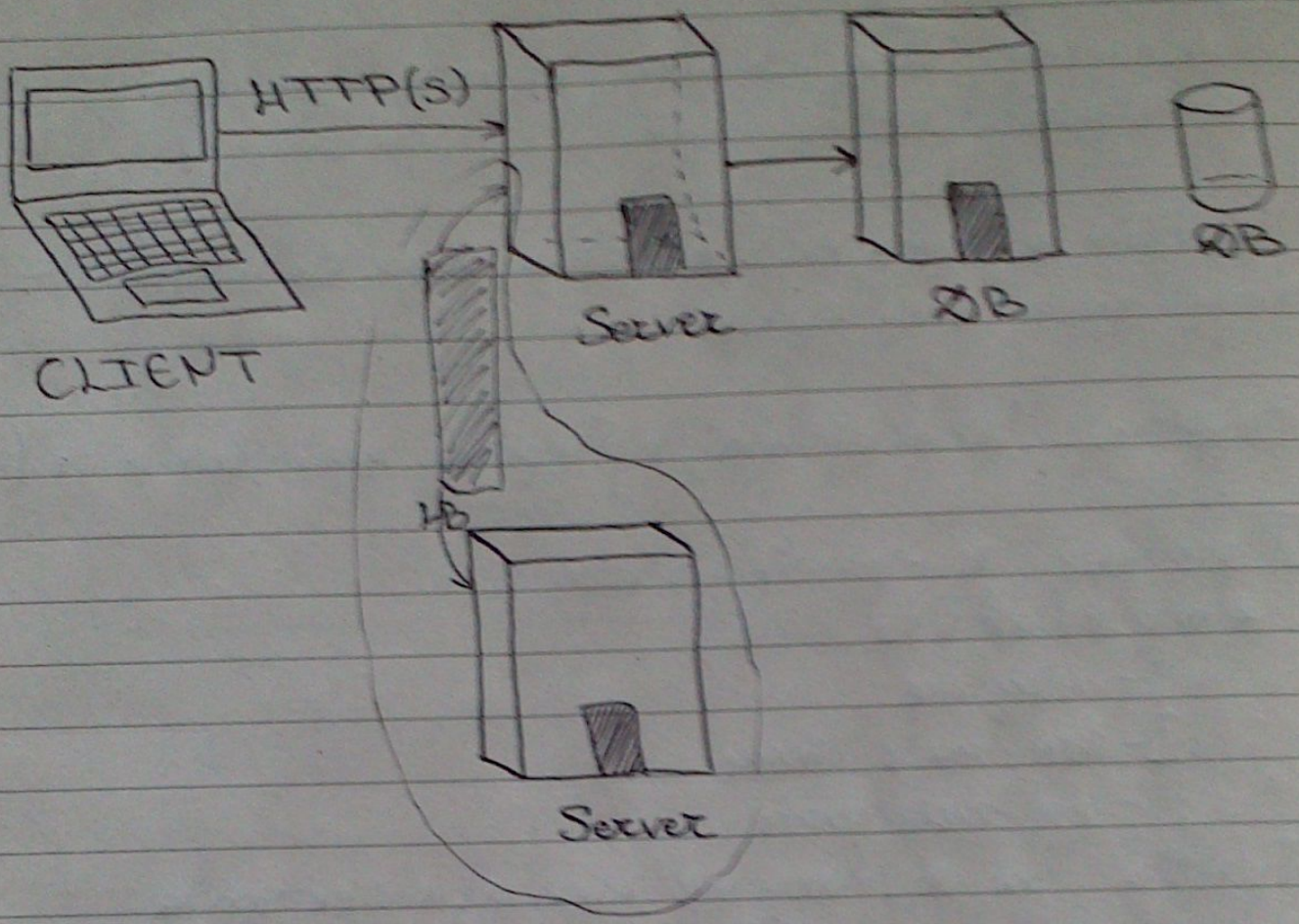


б) Да се състави диаграма на употреба

1. RB ... RP следния интерфейс:

```
request: Request;
Request {
    var username: String;
    var password: String;
}
```

б) Структура на внедряване



САРС – 03.07.2013г.

Тактиките – общ смисъл, няколко примера
availability – излишък

Какво е архитектурен стил и да ги изброим – поне един параграф

Задачи: Изисквания, избиране най-важните (нефън)

Пример: к-modifiability
* modifiability
  - плащане
  - външни системи
  - подкатегории

Трябва да се напише как архитектурата удовлетворява изискванията

* security

Изискването за сигурност се осъществява или чрез HTTPS, или с Encr/Decr модул

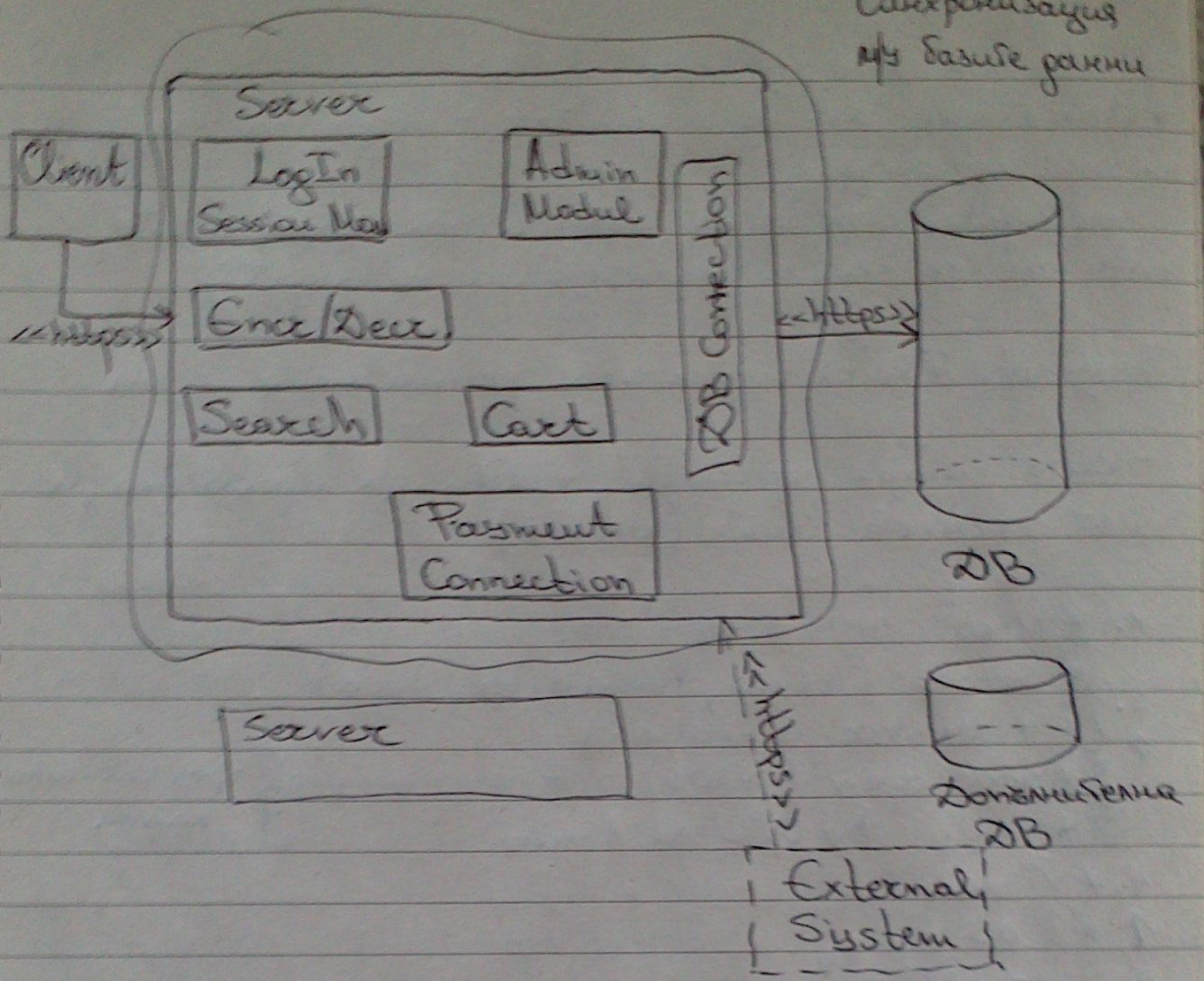
* availability (24/7 примерно)

* DB
* Browser
} => трислойна архитектура

- Продажба на стоки (Trade)

Добавяне на подкатегориите към базата данни
Стандартно решение да се използва HTTPS за сигурност или Encr/Decr

Синхронизация
на базите данни



* Availability - Cloud и се пуска на виртуална машина или друг сървър, active/passive redandancy
↳ по-добрия вариант

* Производителност - load-balanced (разпределяне м/у сървърите)

* Описание

* Статически данни - модул за изчисления
* Структурата на употреба - кой какво ползва (use)

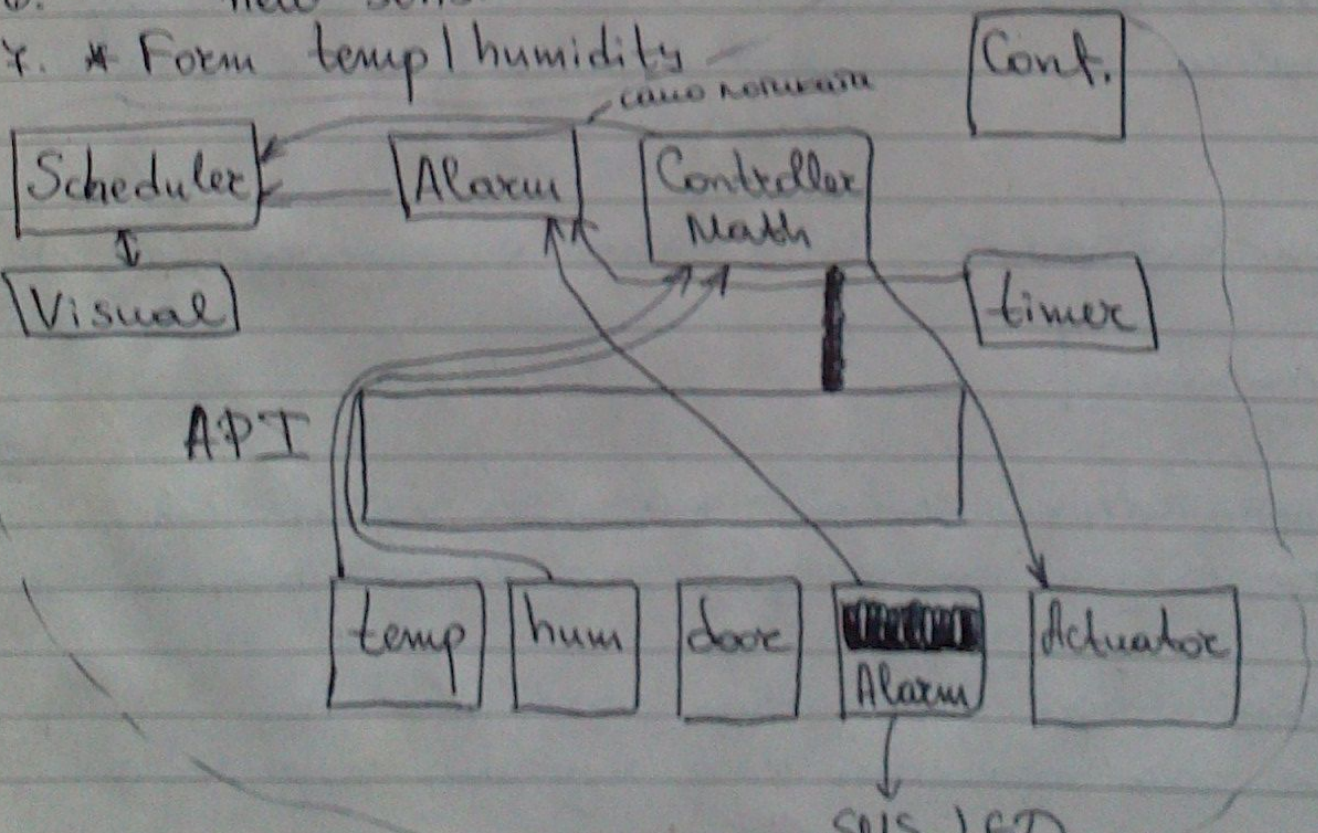
- Няма нужда да правим интерфейсите
- Ако има процеси - то ще има процесите

* Изисквания за бързодействие - модул за подобряване на алгоритъм,
паралелна обработка, арбитър на приоритетите,
повишаване на изчислителната мощност;

Задача: Да се проектира климатична с-ма

- Като имаме датчици, трябва да предвидим, че имат некакъв laser (driver)

1. * temp$^{25°}$, humidity$^{30\%}$
2. * sensors (temp, humidity, door)
   * actuator (клапан)
3. * door open (>20s) - F(функционално) } чрез сигнална лампа
   * temp ↑↓ 05°C за 1 min
4. * visual
   * Mod
5. - SMS - !
6. - new sens.
7. * Form temp / humidity



API - фасада на междинно ниво

[Conf.] - конфигурационни параметри
където се въвеждат тези 25°, 20s и т.н.

HAL - hardware abstraction layer

- Controller - пресмятане на изчисленията
- actuator - само за отворен/затворен
- Alarm - има възможност за промяна на състоянието (външна с-ма - SMS / подмодули - различни оператори)
- Scheduler - като настъпят промени се задейства Controller и Visual опера

Сценарий за качество - може и да има

пр. С-мата трябва да е достъпна 24/7 с изключение на 2 часа седмично за maintanance

САРС - 09.07.2013г.

За темата:
* деф. на нефункционално изискване
* примери за сценарии за качество

Структури на компоненти и конектори - sequence, activity диаграми.

Структурите на компоненти и конектори показват представяне на процесите в системата, какво, е тяхното съответствие с модулите и как тези процеси си взаимодействат помежду си. Връзките м/у процесите са: (избраяване гч).
Избраяване структурите и кая за какво е.

Сценарии за качество - нефункционални изисквания (6 компонента). Какво, къде се случва, кой го предизвиква и т.н.

ATAM - обект, въздействие, резултат, количествени параметри (от слайдовете)

Задачи:

• Сценарий за производителност
пр. Системата обработва при нормална среда до 10000 заявки и върша отговор за 1 секунда.

• Сценарий за надеждност (колко време да е достъпна)
пр. 1) С-мата трябва да е задължително достъпна в работно време (от 9^{00} до 18^{00}).

2) С-мата трябва да е достъпна 24/7, като се допуска 2ч. maintanance в рамките на 1 месец.
При възникнал отказ в с-мата трябва да се даде restart, който да продължи до 20s.

• Сценарий за тестване (testability)
пр. 1) По време на тестване на с-мата да се постигне 75% covetidge в рамките на 2 човеко-дена.
2) С-мата да изтърпи до 80% натоварване на хардуера.
3) С-мата да позволява тестване на 60% автоматизиране на потребителските сценарии.

• Сценарий за сложност
-брой параметри, методи на интерфейс и т.н.

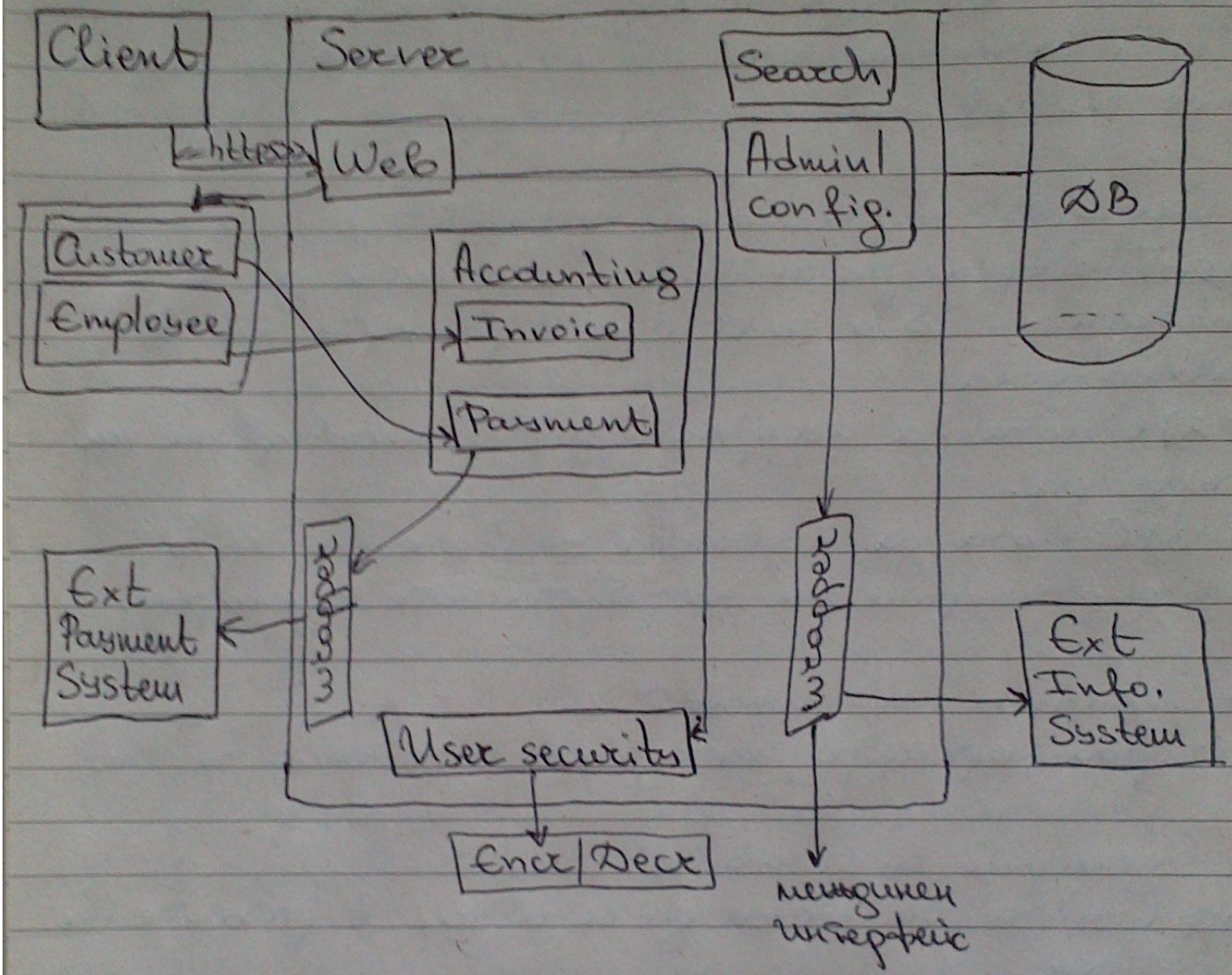
• Сценарий за usability
пр. С-мата да може да се изучи в рамките на 1 ден.

! Може да се даде задача → да се напише примерен сценарий.

* Тактики за производителност, изправност и т.н.

Задачата от листчето!

Декомпозиция на модулите



Взимаме от DB н-я за връзката с тази услуга. Имаме конфигурация и wrapper взима от тях и бор-ва за реализиране потребители.

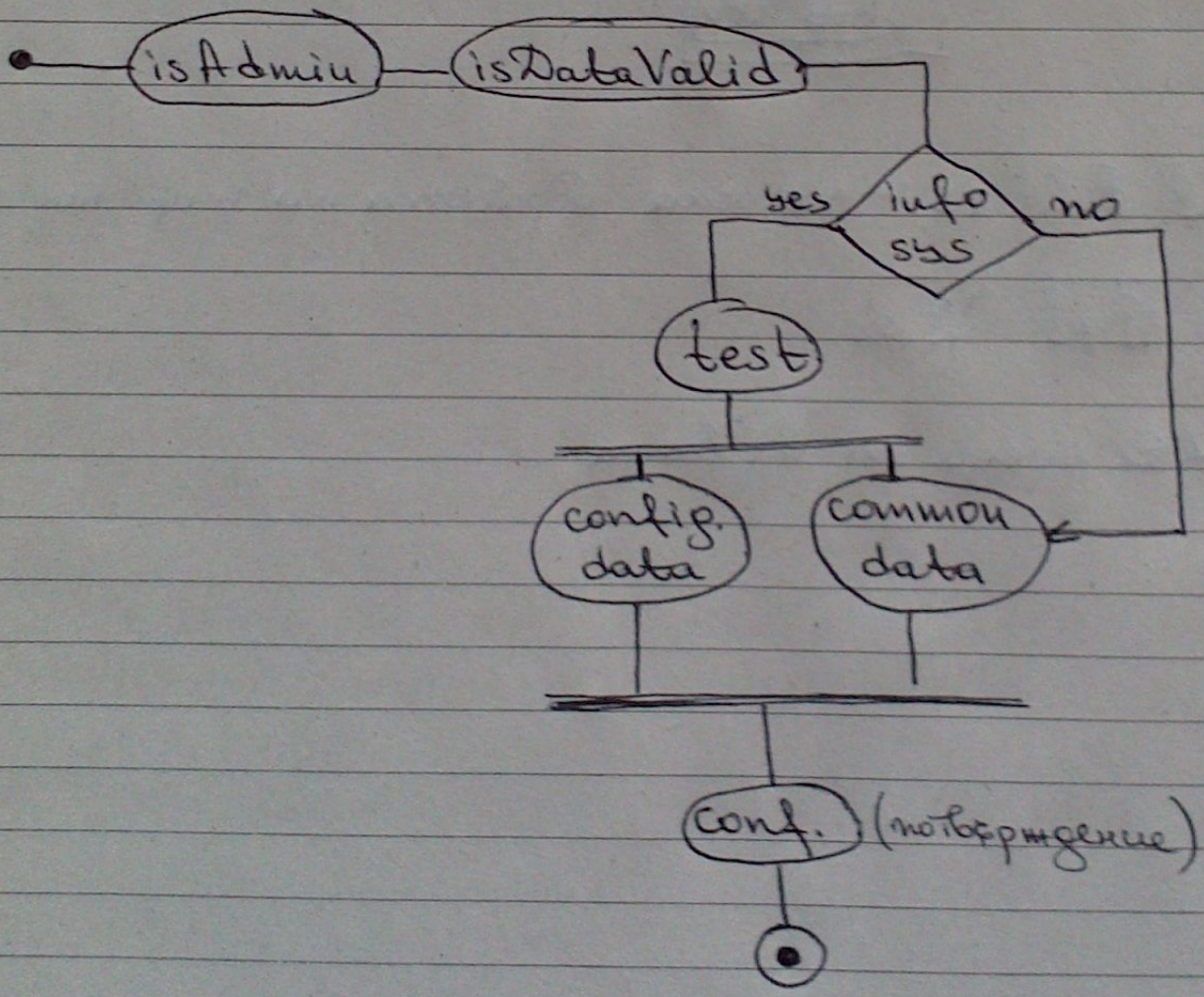
В зависимост от потребителя имаме различни изгледи (подмодули) – employee, customer.

Може да има резервен сървър, който се синхронизира с DB 2 пъти дневно (примерно).

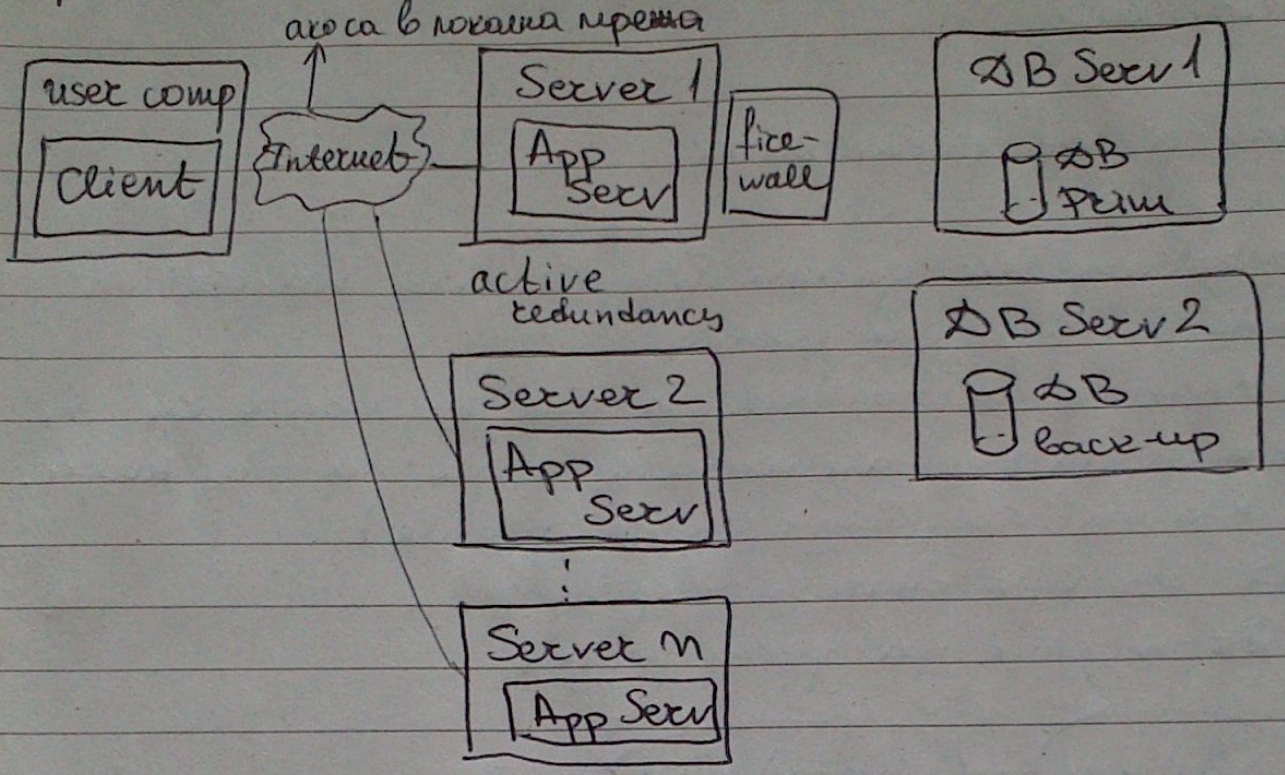
*Да се визуализира процеса на регистрация на партньор

Admin-а ще регистрира.

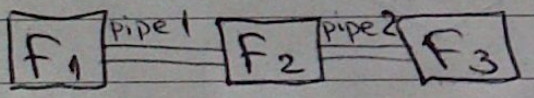
- попълване на данни → isAdmin → input data validation
- запис в базата данни
  - запис на конфиг. параметри, които са за връзка с информационната с-ма на партньора.
  - запис на общи данни (profile)
- тест за връзката с партньора
- потвърждение на регистрация



• Deployment (Разположението) – зад. от мин. път за плащането:



Pipes & Filters – поточна обработка на информация



Пр. IDE среда за обработка

Да се проектира архитектурата (и да се покаже чрез подходящи структури) на система за обработка на туристически услуги и създаване на интегрирани оферти според следните изисквания:

1. Системата се използва от главно от служители в туристическа агенция. Възможните потребителски роли са: туристически посредник, мениджър, финансист и администратор.
2. Дава възможност за търсене на услуги, предлагани от регистрирани партньори на използващия системата (хотелиери, транспортни фирми и авиокомпании и застрахователи).
3. Партньорите могат да се регистрират само от организацията, която използва системата, чрез административен акаунт.
4. Ако партньорите имат програмно достъпна собствена информационна система, то достъпът до актуални цени и наличност на услугите (има/няма билети, места, хотелски легла и т.н.) става през специфичен API.
5. Достъпът до системата от потребителите става през уеб-браузър.
6. Системата дава възможност за търсене на оферти през уеб сайт на туристическата агенция от крайни клиенти и без регистрация.
7. Плащането на услугите става с кредитна карта или в брой в офиса на агенцията.
8. Системата трябва да издава фактура при потвърдено от служител на туристическата агенция плащане и да съхранява всички данни за транзакцията за последваща обработка от финансистите и счетоводния отдел.
9. Всички данни трябва да са защитени от достъп на неоторизирани потребители.
10. Системата трябва да е задължително достъпна в работно време на туристическата агенция.

Да се визуализира процеса на регистрация на партньор. Да се покаже структурата на декомпозиция на модулите.