# Глава 2. Архитектура на облачна система за управление на поръчки от клиенти

Тази глава разглежда решение от високо ниво, което да съсредоточава върху всички основни потребителски, бизнес и ИТ изисквания. Важна част от глава 2 са градивните елементи и интерфейси, изграждащи системата, както и комуникационните модели, които да ръководят композицията. Сложността на операциите стреми да бъде сведена до минимум. Представени са всички случаи на употреба и бизнес сценарии, съвместно с които се моделират приложенията за обслужване на клиенти. Освен това дизайнът обхваща функционалност, използваемост, устойчивост, производителност, икономически, технологични ограничения, компромиси и естетически проблеми на бекенд частта.

## 2.1. Ключови бизнес процеси и дейности свързани със системата за управление на поръчките

Думата „**архитектура**“ често се използва в контекста от високо ниво, което е отделено от детайлите на по-ниско ниво (Martin et al., 2017). Софтуерният продукт, разглеждан в настоящия труд, се състои от 2 клиентски приложения, които се свързват към разпределена бекенд система, базиранa на микроуслуги, работеща върху множество процеси и сървъри (хостове). Всяка услуга се изпълнява в отделен процес като контейнер, разположен в клъстер от виртуални машини. Това разделение на подсистеми и отделни нива и компоненти цели да постигне разбираемост и лесна поддръжка. Работните рамки са съвместими на всяко ниво, без да се дублират функционалности.

На фигура 2.1 са показани приложенията, които изграждат системата за управление на поръчките от клиенти.

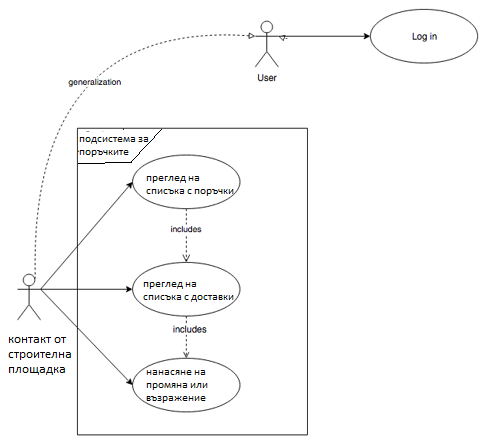


Фиг. 2.1. Диаграма от високо ниво на главните приложения. (разработка на автора)

Тази подточка представя важни случаи на употреба, които са критични за бизнеса и са част от основния домейн. Използвани са UML диаграми на бизнес сценариите. Те идентифицират действия, които очакваме потребителите да направят.

Най-подходящ за взаимодействие с крайните потребители са мобилните приложения. Важни техни характеристики са, че поддържат функции като местоположение, камера и работят с уеб API. Клиентите на фирмата, които се явяват крайните потребители, управляват и проследяват поръчките и доставките в реално време с мобилно приложение. Целта му е да помага с планирането и логистиката, да въздейства върху крайния резултат с информация и данни. Тази информация, на смартфона, трябва винаги да е актуална, тъй като текущото състояние на поръчка и местоположение на доставките се проследява на живо. Други възможности са преглед на история, създаване на нова, промяна или отказване на **не активна** съществуваща поръчка. Приложението може да се разпостранява безплатно чрез Google Play Store и Apple App Store.

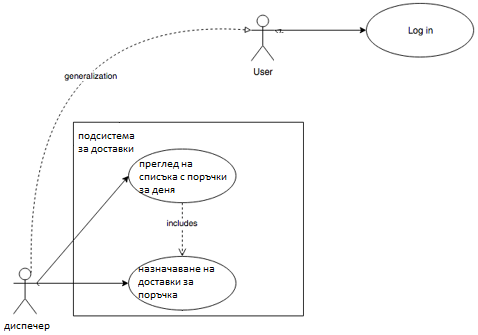
Обхватът на мобилното приложение, насочено към крайните клиенти, включва екран за вход, интерфейс за текущите поръчки и доставки към тях. Също така панел за създаване или промяна на поръчка. Фигура 2.2 представя процесите под формата на диаграма.



Фиг. 2.2. Диаграма на главен бизнес сценарий (разработка на автора)

Уеб порталът е софтуер насочен към диспечерите, е част от цялостната система за управление на транспорта (TMS). Чрез него могат да се създават поръчки и доставки, като същевременно се сравняват, за да се гарантира, че поръчките се доставят от най-подходящото превозно средство. Уеб порталът служи като инструмент за вземане на решения, с предварително зададени предложения, които могат да бъдат одобрени, или отхвърлени и променени, според гледната точка на диспечера на смяна. Вземайки под внимание текущите събития, подсистемите зад уеб портала насрочват за доставка това, което и когато клиентът е поръчал. Те разчитат на правилна информация за поръчка и актуализация на събития. Целта е да се минимизират разходите.

Обхват на уеб портала включва балансиране на работното натоварване на превозните средства, позволява проследяване и коригиране, както на поръчките, така и на доставките, осигурява предварително зададени решения, на база на които диспечерите могат да коригират и контролират броя на доставките. Диспечерите имат възможност да поправят грешни данни, като говорят с клиентите или шофьорите. Същевременно всички промени се отразяват в мобилното приложение.



Фиг. 2.2. Диаграма на главен бизнес сценарий (разработка на автора)

SyncroTESS прилага усъвършенствани техники за оптимизиране: ● Три вида оптимизиращи контексти, така наречените „режими“ ● Оптимизация в реално време (RT) ● Оптимизация за оценка на поръчки (OEV) ● Оптимизация на предварителното планиране (PP). RT/OEVPP Постоянно се извършва във фонов режим Извършва се на преден план Едно стартиране на оптимизатора следва следващото При поискване за игра на сценарии „какво ако“ Параметрите определят минимално време на цикъла Отделени от работещия бизнес

…..

Входни данни:●Завод/Паркинг/Клиентски местоположения●Камиони, техните свойства и статус●Поръчки●Параметри на оптимизатора Различни оптимизатори за различни бизнес линии:●Прехвърляне/FIFO за Бетон●Съжаление за Инертни материали●COFLO за Цимент (помпи и баржи само ръчно изпращане)

Бизнес линиите и специалните данни са разделени в групи за изпращане. Възможна е повече от една група за изпращане за бизнес линия:

## 2.2. Концептуален модел на системата

Концептуалните модели са абстрактни представяния за това как трябва да протича изпълнението на задачите. Те представят визуално концепция или операция. За визуализиране и конструиране на елементите е използван унифицираният език за моделиране (Unified Modeling Language).

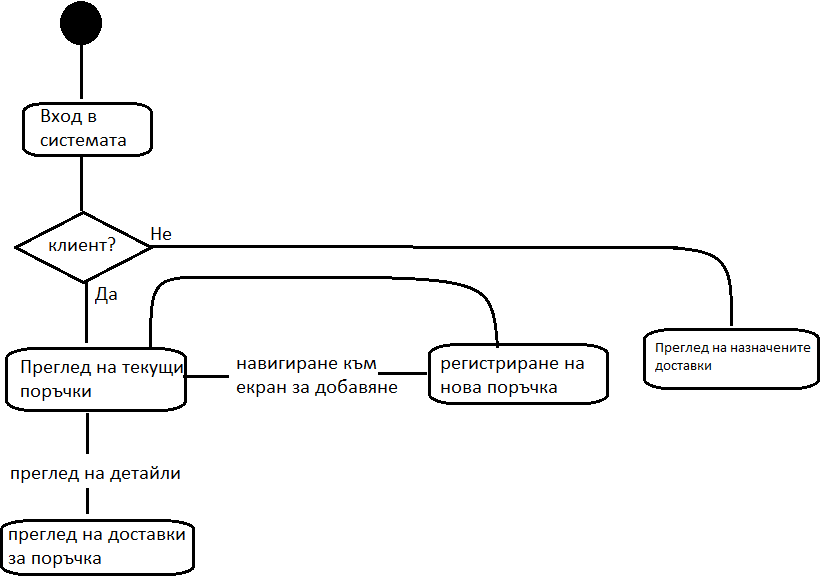
### 2.2.1. Поведенчески диаграми

Поведенческите диаграми идентифицират как различните елементи взаимодействат помежду си.

#### 2.2.1.1 Диаграми за активност UML

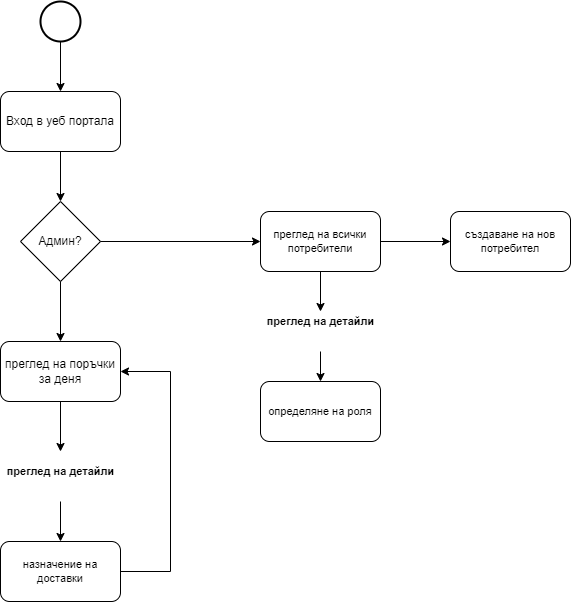
Диаграмите за активност изглеждат много подобни на блок-схемите. Наличието на тези прилики улеснява комуникацията между технически и не-технически лица (stakeholders).

Следната диаграма представя работни потоци и общи операции за мобилното приложение:



Фиг. 2.3. Диаграма на активноста за мобилно приложение. (разработка на автора)

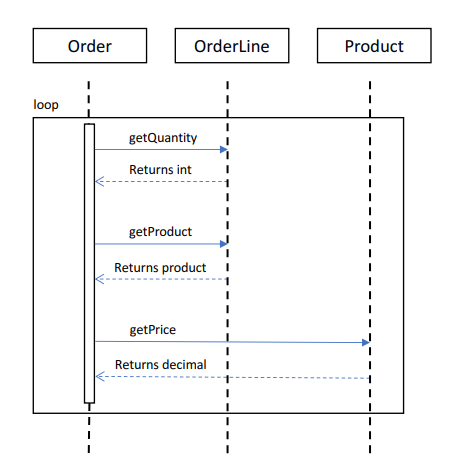
Следната диаграма изобразява потока от операции в уеб портала:

 Фиг. 2.4. Диаграма на активност за уеб портал. (разработка на автора)

#### 2.2.1.2. Диаграма на последователностите UML

Диаграмите на последователностите също са често използвани поведенчески диаграми в UML. Те идентифицират как обектите в система взаимодействат помежду си, за да реализират определена функционалност, като визуализират времевата линия и редът, в който се извършват операциите.

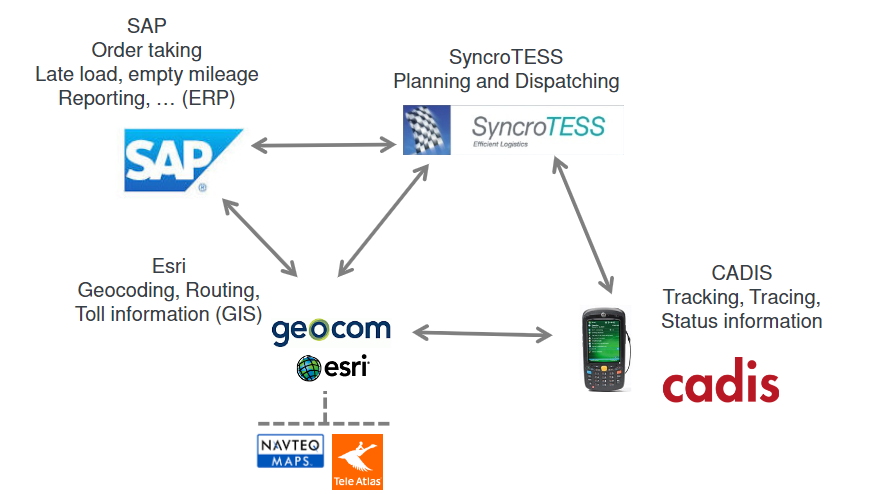
Пример може да разгледаме в контекста на подсистемата за поръчки:



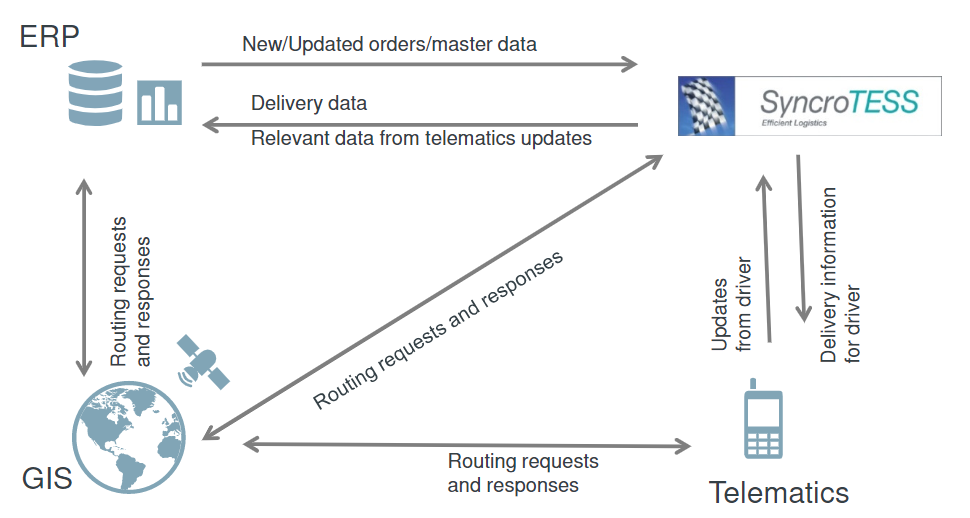
Фиг. 2.7. Диаграма на последователностите. (разработка на автора)

<https://blog.openreplay.com/jwt-authentication-best-practices>

#### 2.2.1.3. Общ преглед на системата и интерфейси



……………………..



…………………….

### 2.2.2. Структурни диаграми

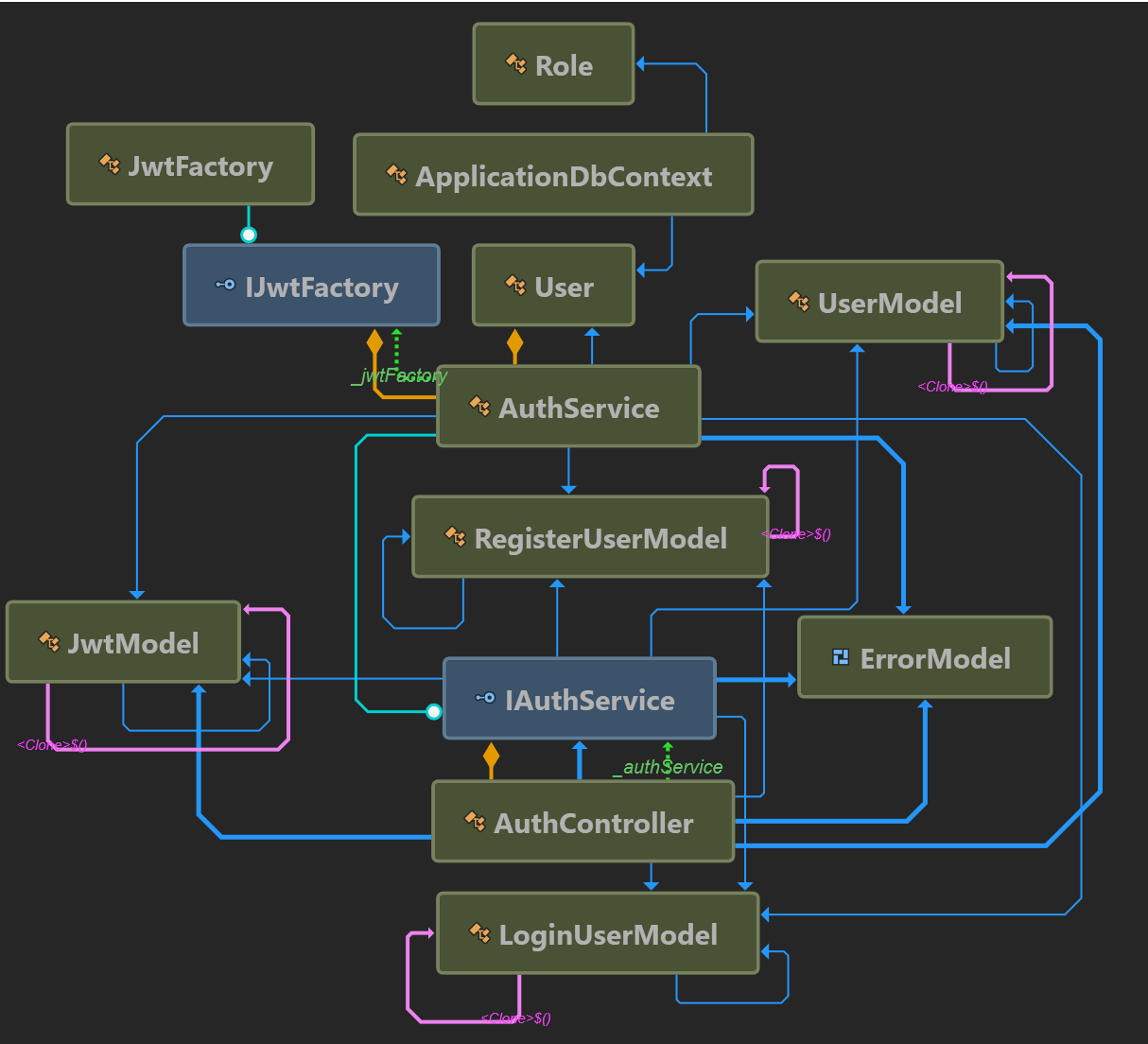
Структурните диаграми помагат за дефиниране цялостната структура системата, подобно на плана, който определя как изглежда една къща. Структурните диаграми моделират как изглежда системата в архитектурно отношение. Те ни помагат да дефинираме „речника“ на системата, гарантират съгласуваност от заинтересовани страни в проекта. Идентифицират различни връзки между различните частти.

Структурните UML диаграми изобразяват елементите на система, които са независими от времето и които предават концепциите и как те се свързват помежду си. Елементите в тези диаграми приличат на съществителните в естествения език.

#### 2.2.2.1. Диаграма на класовете UML

Диаграмите на класове са едни от най-често срещаните, когато става на въпрос за разработката на софтуер. Едно от основните неща, които тези диаграми правят е да идентифицира речника на системата. Например, те определят връзките между обектите, които съответстват на основните съществителни.

Следващата част представя диаграма на класовете, свързани с удостоверяване. Това е процесът на определяне кой има достъп до системата. Елементите от приложението и зависимости, които обслужват тази част са визуализирани на фиг. 2.2. **DbContext** и **ApplicationUser** представляват комбинация от класове, които оперират с базата от данни.  **AccountController** използва тези свойствa чрез **UsersService**, който капсулира логиката по безопасен за използване начин.

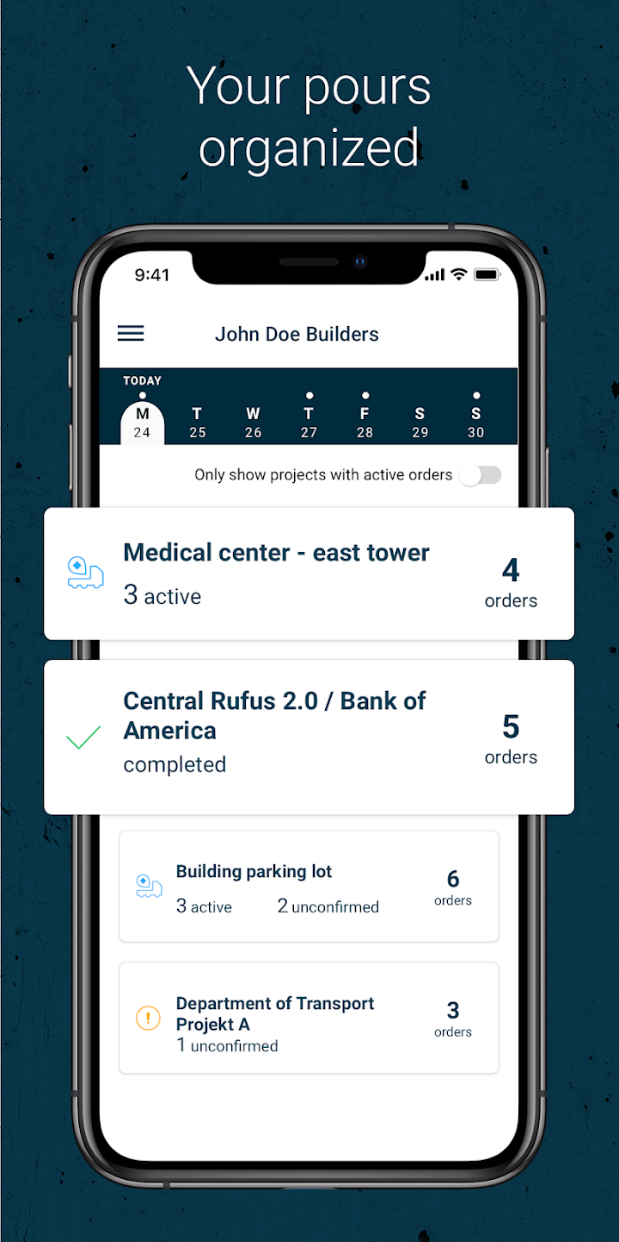


Фиг. 2.8. Интерактивна диаграма за БСВС „Разработка на продукт / Планиране на проект”. (разработка на автора)

## 2.3. Функционалност и потребителски интерфейс

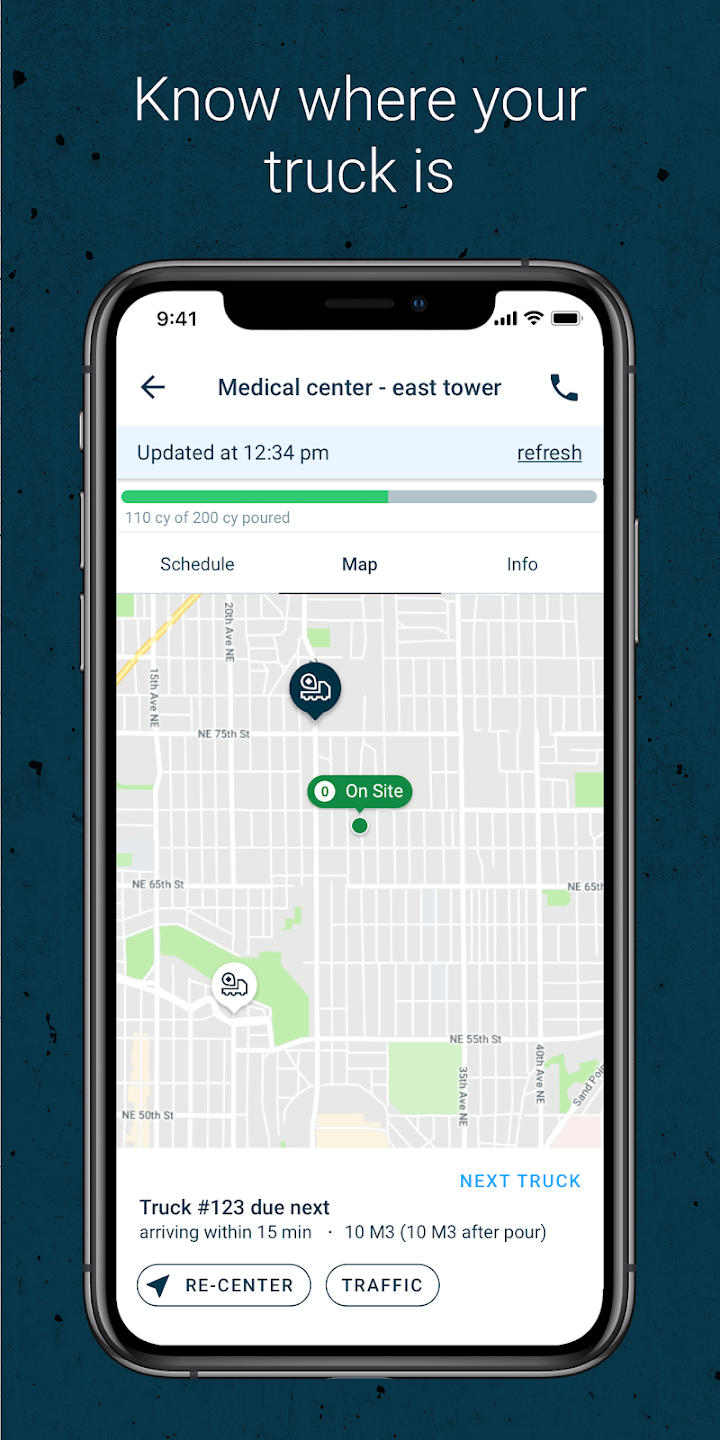
Ясно дефинираните изисквания са основата на успешен проект, тъй като включват набор от процеси като анализ, спецификация и валидиране. Функционалните изисквания са продуктови характеристики, които разработчиците трябва да внедрят, за да позволят на потребителите да изпълнят своите задачи. Като цяло функционалните изисквания описват поведението на системата при определени условия.

Нека започнем с преглед на характеристиките и изискванията на мобилното приложението. Както беше отбелязано, то представлява приложение за поръчки и проследяване на доставки онлайн. Примерна скица на интерфейса на началния екран, след вход, в на потребителското приложение е даден на фиг. 2.10. Представени са основни елементи на заглавна част – име на текущ потребител, инструмент за избор на дата и списък на текущите поръчки.



Фиг. 2.10. Скица на основен екран на приложението. (разработка на автора)

Фигура 2.11. представя детайлна информация за определена поръчка и доставките към нея.



Фиг. 2.11. Скица на екран за информация за поръчка. (разработка на автора)

Регистрирането на нови поръчки ще се осъществява чрез екран в главното меню.

Фигура 2.12 представя използвано на приложението от доставчика. Както беше споменато, мобилното приложение допринася за бързо изпълнение на процесите, сравнително лесно за използване и удобно за работа, чрез функционалностите за достъп до геолокация, навигация, съобщения, телефон. Това допринася за стандарт за сигурност на данни и връзки. Водача може да провери списъка, с предстоящи доставки, назначени към него. Приложението притежава функционалност за уведомяване, в случай че списъкът е експортиран, но няма насрочени задачи.



Фиг. 2.12. Скица на начален екран за доставчика. (разработка на автора)

Посоченият екран съдържа подробности за поръчка, включващ материал, количество, местоположение за товарене и разтоварване и планирани часове. Тъй като, доставката може да бъде анулирана или пренасочена към друга поръчка, остатъкът се докладва и след това бива върнат, използван повторно или отклонен. Също така шофьорът може да съобщи за повреда, като отписването е възможно след изпращане на лог файлове към диспечера.

За да подпомогне автоматизирането и рационализирането на документацията, приложението поддържа функционалност за електронно доказателство за доставка. Това е процес, който създава документацията, валидираща получаването на стоката от клиента. Традиционно POD се осъществява чрез подпис на клиента на физически документи. В случай на липса на подпис трябва да се посочи причина. Мобилното приложение „улавя“ съответните данни и снимки, като това бива последния етап от доставката. Следната фигура представя екрана за тази функционалност.



Фиг. 2.13. Скица на екран за доказателство за доставка (ePOD). (разработка на автора)

Нефункционалните изисквания често се наричат ​​„атрибути за качество“ на системата. Те са критериите за оценка на това как една софтуерна система трябва да работи.

Следващите точки отбелязват някои от основните изисквания:

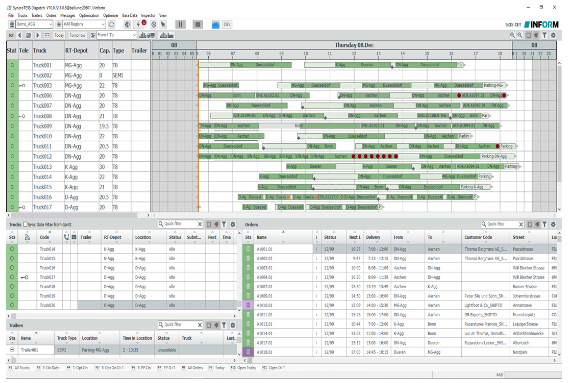
• Системата трябва да е високо-достъпна и да може автоматично да разширява мащаба, за да отговори на увеличаващия се трафик (също така да намалява мащаба, след като трафикът спадне);

• Трябва да осигурява лесен диагностични дневници, за да помогне при отстраняване на неизправности или други проблеми, които би могли да възникнат по време на работа;

• Трябва да поддържа гъвкав процес на развитие, включително подкрепа за непрекъсната интеграция и внедряване (Continuous integration / deployment);

• Трябва да поддържа междуплатформен хостинг и развитие;

TODO: ИЗИСКВАНИЯ ЗА бързодействие и изпълнение / натоварване / ИЗИСКВАНИЯ ЗА ОБЕМ ДАННИ / брой потребители /



Графичен потребителски интерфейс (GUI) Използван от разпределители и планиращи Информация за поръчките, които трябва да бъдат доставени Пълен контрол върху всички камиони

Изглед на приемащия поръчки (OTV) Дава на обслужването на клиенти представа как ще изглежда денят Важна цел: Постижим план до края на деня Определете допустимото ниво на свръхрезервиране (и кажете на поемащите поръчки)

Depot viewer (DV) – Уеб инструмент, използван в рамките на депата, за да видите какво се приближава Дайте актуализации на състоянието на пристигане/напускане на завода и натоварването за камиони без телематика и неинтегрирани инсталации Включвайте/изключвайте камиони без телематика

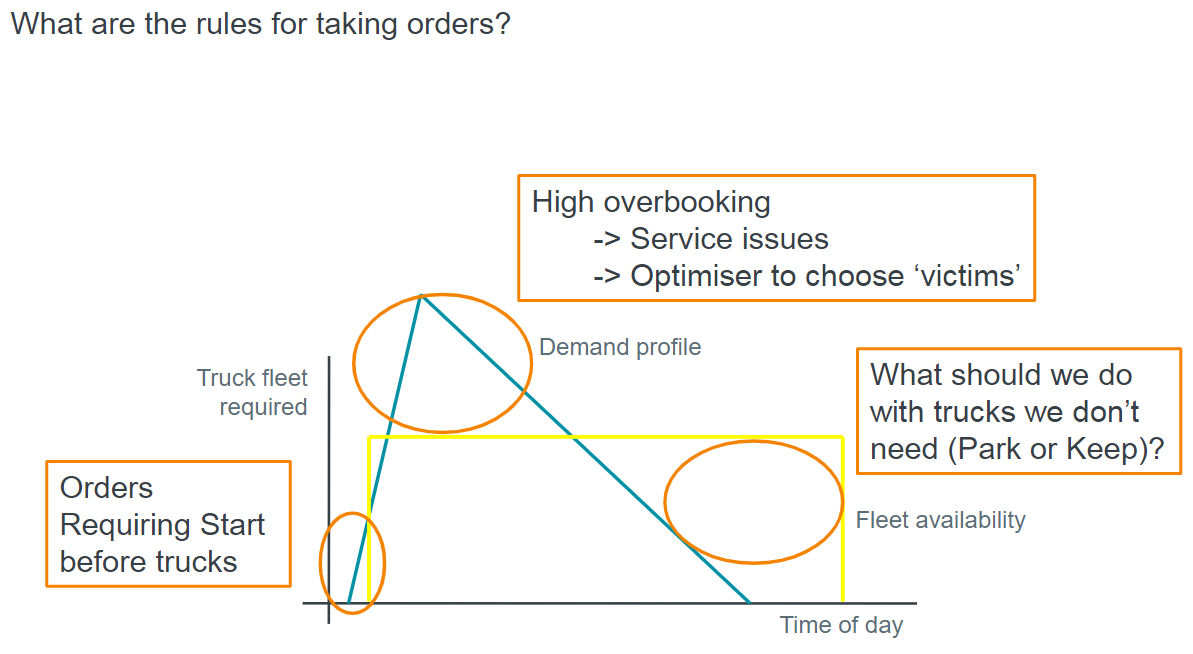
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

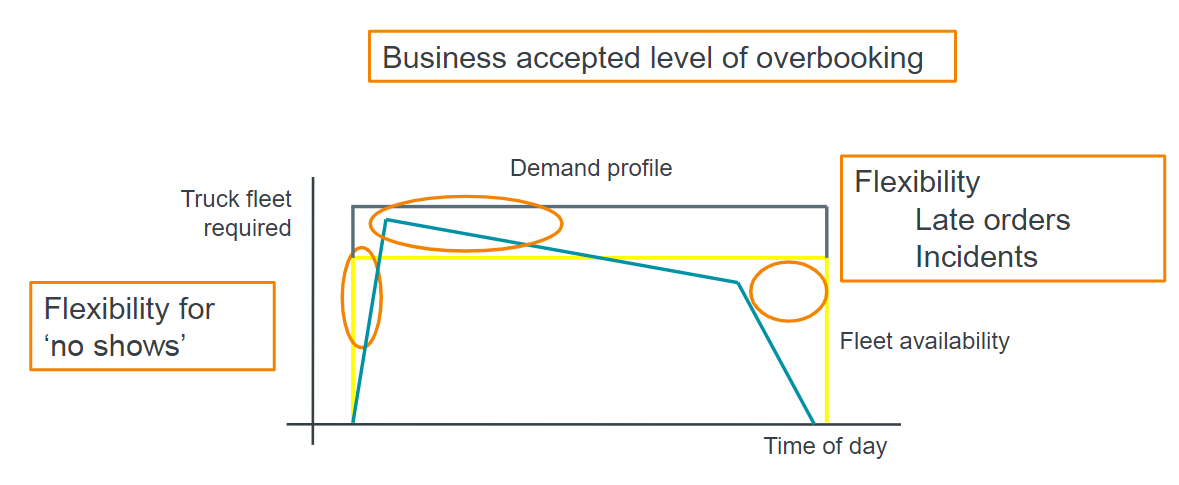
Планиране на камиони●Планиране от началото на деня до края на всяка смяна●Включва пътуване от паркинга до вкъщи през нощта●Може да премести камиони от едно депо в друго.Изпълнение на поръчката●Изцяло навреме, въз основа на информацията в поръчката●Ще се стреми към редовно доставка от клиента●Няма да търси свръхпредлагане на сайта●Ще трябва да се увери, че информацията за поръчката е правилна за времето и разпределението на товара

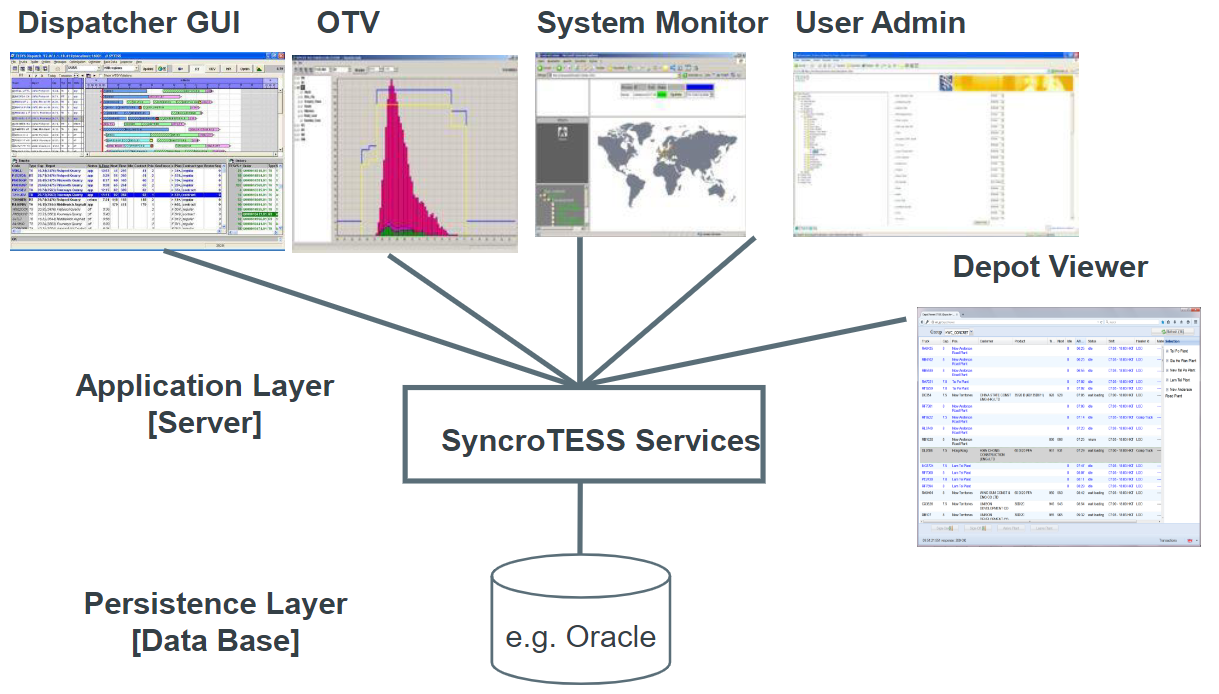
\_\_\_\_\_\_\_\_\_\_\_\_

Отговорности●Управление на профила на поръчката●Отговаряне на въпроси на клиенти относно доставките●Връзка с разпределителите за подпомагане на заявки за промяна на поръчката●Връзка с разпределителите●Промени на входящи поръчки●Закъснели заявки за поръчки●Поемане на „нормални“ поръчки●Известие за късно зареждане

правила за приемане на поръчки







Дава изглед на високо ниво какви са изискванията на камиона●Трябва да се използва като основа за Demand SmoothingView върху същите данни, които виждат всички останали●OTV изгледът се променя с промяната на поръчките и извършването на доставкиЕ неоптимизиран изглед на поръчките●Отразява времената от поръчките●По-малко изразителни за големи времеви прозорци за доставка и минимално разпределение на товара Също така позволява лесен достъп, за да видите кога са доставени поръчките●Възможно е да се отговори на въпроси на клиенти като●„Кога ще бъдете тук?“●„Колко вече сте доставено?'



Изглед на Gantt: Един ред на камион със съответните планирани/присвоени доставки ●Решетка на камиони: Съдържа всички камиони със съответните основни данни и данни за състоянието ● Решетка на поръчките: Съдържа всички получени поръчки и съответните доставки

33© 2019 -INFORM GmbH Диаграма на Гант – Преглед Таблица на камиони Времева скала Сегашна линия (червена) Превключване на камиони (бяло) Включване/изключване (триъгълници)

34© 2019 -INFORM GmbH Мрежите в общи линииОбща функционалност на всички мрежи●Ширината на колоната може да се променя чрез плъзгане●Подсказката показва съдържанието на колоната в случай, че ширината е твърде малка●Сортиране●Сортирането е възможно по всяка колона (възходящо/низходящо)●Щракнете върху заглавието на колоната за сортиране (стрелка)●Възможности за вложено сортиране (вижте по-късно)●Персонализиране на показаните колони и техния ред.Текущ критерий за сортиране (стрелка)





GUI анкетира сървъра: ●Актуализира на всеки 10 секунди и след всяка транзакция●GUI искания 'Модели' и 'Полета'●Различни потребители= различни GUI и всички GUI polldataindependent●GUI-сървърът трябва да отговори на всяка заявка за актуализиране на GUI поотделно●Сглобяване на всички по-нови данните изискват изчислителна мощност➢Броят на GUI на GUI сървър е ограничен.

…..

Резюме:●SyncroTESS съхранява (почти) пълната оперативна база данни в паметта●В оперативната база данни се поддържа само малък набор от данни.Историческите данни не са необходими за оптимизиране и изпращане:●Данните от последните дни се преместват в архивна база данни още на възможно ● След като данните бъдат прехвърлени, те могат да бъдат изтрити от оперативната база данни.

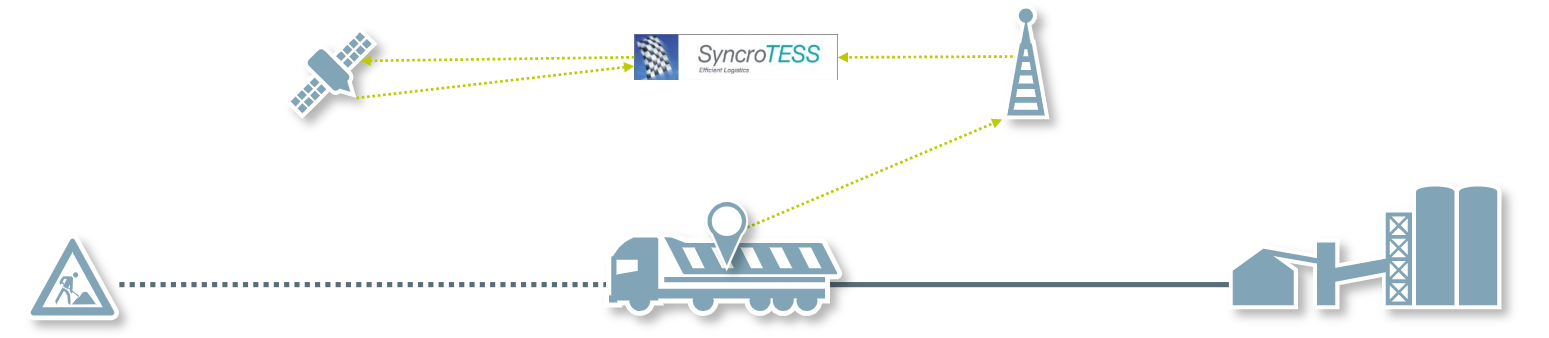
SyncroTESS трябва да знае колко време отнема да се стигне от А до Б:●Продължителността на пътуването от завода до мястото на клиента се намира в Idoc.НО:●Пътят обратно може да е различен (еднопосочни улици)●От камиона може да бъде поискано да върнете се някъде другаде=> SyncroTESS трябва да поиска инструмент за разстояния и продължителност на пътуването при поискване

………..

SyncroTESS поддържа вътрешна „Матрица за време и разстояние“ (често наричана „Разстояния и продължителности“, или „диду-матрица“), за да съхранява заявените времена и разстояния (една матрица на клиент):



Заявки за маршрутизиране :●За всеки входящ Plant Master Idoc●За входящ Idoc за поръчка●Всеки нов клиентски сайт генерира нов ред и колона в матрицата●Не, ако в поръчката е въведено фиксирано време и разстояние за пътуване или ръбът, който вече е наличен в матрицата●За квартала на сайта на клиента, определен от „Mapping Dist. Threshold’●Веднъж на нощ за cust/custrelations в помпени групи



SyncroTESSизползва избрани събития за местоположение, за да поиска прогнозната оставаща продължителност на пътуването от ESRI:●Необходима е телематична система, изпращаща геокоординати●След 25% и 50% от очакваната продължителност на пътуването на пътуването, SyncroTESS иска от ESRI актуализация на ETA за следващото отчетено събитие за местоположение ( след 50% на всеки 10 минути)

……………….

SyncroTESS трябва да комуникира с камиони и заводи: ● Да знае за текущите статуси (напр. Включване/изключване, почивки, отчети за състоянието) ● Да знае за началото и края на товарите ● Да информира за задания, анулации и т.н. => SyncroTESS се нуждае от телематична система който дава обратна връзка

………….

Докладът за късно зареждане (LLR) е обратна връзка от SyncroTESS към SAP:●Ако клиентските поръчки закъснеят, клиентите могат да бъдат извикани проактивно●Цел: Ако все пак закъснеем, по-добре е да кажем на клиента предварително●LLR е насипно данни (много трафик)●Нормален Idoctransfer е твърде бавен●Решение Horizon с връзка към база данни n/a за HeidelbergCement (одит)●Нова реализация с RFC●LLRFrontprogram потвърждава незабавно входящото съобщение●Ако връзката към SAP е прекъсната, съобщението се губи ( без опашка)

………

The Material Replenishment(MRP)

За производството на бетон са необходими суровини и SyncroTESSплан за група бетони определя търсенето на бетон:●SyncroTESSекспортира търсенето на готов бетон ●на завод●на продукт●на интервал от половин час●SAP използва тази информация, за да изчисли търсенето на суровини●SAP генерира/актуализира автоматично обобщените поръчки

……….

Mileage Interface

За внедряване на схемата за заплащане на превозвача, SAP изисква информация за натоварения и разтоварен пробег на всеки камион: ●Информацията може да бъде извлечена от базата данни SyncroTESS●INFORM предоставя ежедневно експортиране на пробег въз основа на отчет в статистическата база данни ●Автоматизиран интерфейс е проектиран и внедрен, за да замени ръчното извличане на данни ●SAP предоставя нов функционален модул за получаване на данни за пробега от SyncroTESS чрез RFC.

Веднага след като камион приключи смяната си, всички изминати натоварени и разтоварени маршрути се записват в таблица в операционната база данни на SyncroTESS●Веднъж на ден се стартира „Пробег за експортиране на пробег“, който събира всички тези данни и ги експортира с уникален RunIdin RFC към SAP (чрез extpqmiddleware)●Ако това експортиране е било успешно, събраните данни за пробега ще бъдат изчистени от оперативната база данни на SyncroTESS●Ако е било неуспешно, съобщение, съдържащо RunId, ще бъде създадено в полето за съобщения на SyncroTESS и данните остават в SyncroTESS

Epod

Едно допълнително подобрение е поддръжката на електронно доказателство за доставка (ePOD)●Когато материалът е доставен, от клиента се иска да потвърди получаването на материала чрез подпис на мобилното устройство●Подписът се предава заедно с допълнителни данни (напр. A списък с допълнителни услуги) от устройството чрез cadis към SyncroTESS●SyncroTESS препраща ePOD документа чрез RFC към SAP●SyncroTESS не съхранява или по друг начин обработва ePODdata вътрешно.

Не се използва криптиране за предаване на ePODdata●Подписът се предава като base64 кодиран низ чрез XML и се записва като TIFF-файл в /usr/syncrotess/inst/v9/var/work/epod/●Файловете с подписи са сортирани в поддиректории DEU\, POL\, ...● ePO не е законно доказателство

……

Резюме:●Всяка команда е транзакция●Всяка команда може да задейства други команди●Всяка команда може да има странични ефекти (напр. изходящ Idoc)➢Всяка команда се записва във файл; това може да се използва за възпроизвеждане на поведение. Една обща директория с регистрационни файлове: $INSTROOT/var/log/●Всички tservprocesses на сървър записват в един и същ набор от лог файлове ('cmd.out.\*')●Всички възли на клъстер SyncroTESS споделят едно и също набор от лог файлове ●Една директория $INSTROOT/var/logXXper възел XX