

Курсова работа по Софтуерни архитектури и разработка на софтуер

Тема: *AgroSmart*

Изготвили: Мария Марковска – 62568

Цветан Стръмски – 62586

Тина Томова – 62594

Съдържание

0. Въведение.....	3
0.1 Организация на документа.....	3
0.1.1 Предназначение.....	3
0.1.2 Обяснение на структурите.....	3
0.1.2.1 Декомпозиция на модулите	3
0.1.2.2 Структура употреба на модулите	4
0.1.2.3 Структура на процесите	4
0.1.2.4 Структура на внедряването.....	4
0.1.3 Структура на документа.....	4
0.2 Общи сведения за системата.....	4
0.3 Разширен терминологичен речник.....	5
0.3.1 Списък на софтуерните елементи.....	5
0.3.2 Терминология.....	5
1.Драйвери.....	7
2.Модулна декомпозиция.....	8
2.1.CLIENT.....	8
2.2.SERVER.....	9
2.3DEVICE MANAGER.....	10
3.Допълнителни структури.....	11
3.1.Употреба на модулите.....	11
3.1.1.Мотивация	11
3.1.2.Първично представяне.....	11
3.1.3.Описание на връзките.....	12
3.1.3.1.От Client до:.....	12
3.1.3.1.1.Client.....	12
3.1.3.1.2Подмодули в Server.....	13

3.1.3.1.3.Device Manager.....	13
3.1.3.2.От Server до:.....	13
3.1.3.2.1.Server.....	13
3.1.3.2.2.Подмодул в Client.....	13
3.1.3.3.От Device Manager до:.....	13
3.1.3.3.1Device Manager.....	13
3.2.Структура на процесите	14
3.2.1.Мотивация	14
3.2.2.Описание на елементите и връзките.....	16
3.3.Структура на внедряване.....	17
3.3.1.Мотивация за избор.....	17
3.3.2.Първично представяне.....	17
3.3.3Описание на елементите и връзките.....	18
4.Архитектурна обосновка.....	19

0. Въведение

0.1 Организация на документа

0.1.1 Предназначение

Предназначението на документа е да се представи софтуерната архитектура на системата AgroSmart, предназначена за отдалечено управление на дейности свързани със земеделието, под формата на различни команди.

0.1.2 Обяснение на структурите

0.1.2.1 Декомпозиция на модулите

Показва системата, разделена на отделни модули, които са: Клиент, Сървър, Управление на устройствата.

Клиент е приложението, чрез което потребителят взаимодейства със системата.

Сървър - осигурява бизнес логиката, административните функции и връзката между модулите.

Управление на устройствата осигурява достъп до сензорите, събиране и обработване на информацията от тях

0.1.2.2 Структура употреба на модулите

Тази структура визуализира различните статични връзки между отделните модули и подмодули на системата.

0.1.2.3 Структура на процесите

Изобразяват се нагледно някои от важните процеси на системата.

0.1.2.4 Структура на внедряването:

Показва начина на разпределение на софтуера върху хардуера.

0.1.3 Структура на документа

Документът се състои от 4 секции:

0. Въведение към документа.
1. Определяне на архитектурните драйвери.
2. Структура от тип “Декомпозиция на модулите”.
3. Допълнителните структури, които са от тип “Структура на употреба на модулите”, “Структура на процесите” и “Структура на внедряването”.

0.2 Общи сведения за системата

AgroSmart, предназначена за отдалечено управление на дейности свързани със земеделието, под формата на различни команди, например за стартиране/спиране на поливане, наблюдение на данни в реално време, справки и анализи и други.

0.3 Разширен терминологичен речник

0.3.1 Списък на софтуерните елементи

Номерът пред името на елемента показва секцията в която се намира.

2.1. CLIENT

- 2.1.1. GUI
 - 2.1.1.1. Registration Form
 - 2.1.1.2. Login Form
 - 2.1.1.3. Add/Remove
 - 2.1.1.4. Optimisation plans
 - 2.1.1.5. Statistics
- 2.1.2. Validator

2.2. SERVER

- 2.2.1. Request Handler
- 2.2.2. Forecasting Algorithm
- 2.2.3. GUI Maker
- 2.2.4. Device Controller
 - 2.2.4.1. Live Data Fetcher
 - 2.2.4.2. Moving Parts Controller
- 2.2.5. Admin Supervision

2.3. DEVICE MANAGER

- 2.3.1. Formatter
- 2.3.2. Synchronizer
- 2.3.3. Reader
- 2.3.4. DEVICE DB

0.3.2 Терминология

- Софтуер – набор от машинно-четими инструкции, които управляват действията на компютърния процесор
- Софтуерна система – комуникиращи помежду си компоненти, които са част от една
- Операционна система (ОС, operating system) – софтуер, който управлява ресурсите на хардуера и софтуера и осигурява общи услуги за компютърните програми
- Android, iOS – видове мобилни ОС
- Браузър – е приложен софтуер, предназначен за разглеждане на информационни ресурси в Уеб.

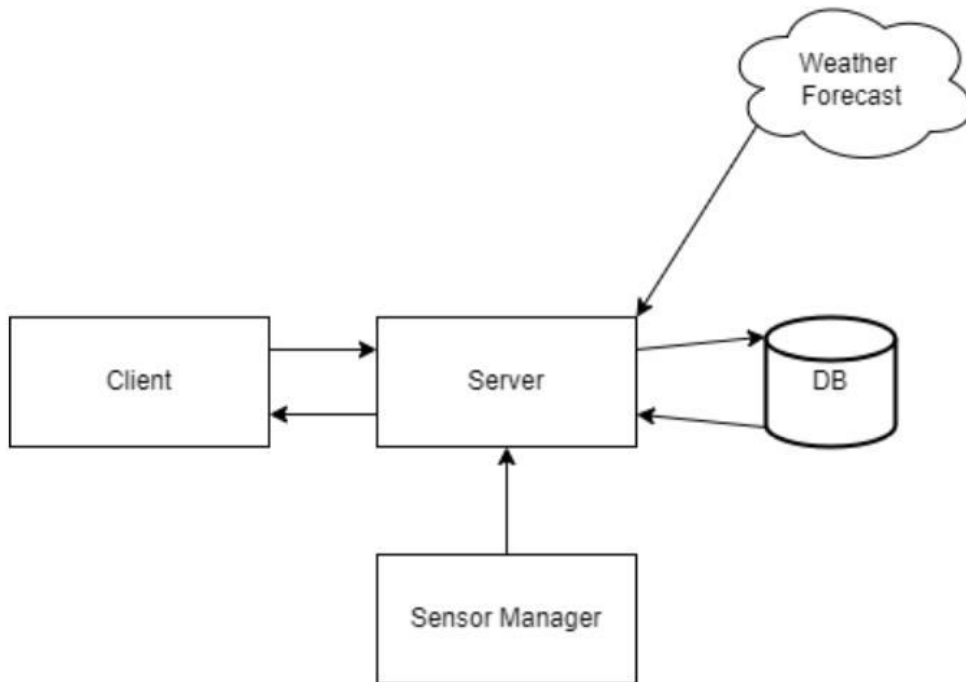
- База от данни – колекция от информация, която е така организирана, че да може лесно да се достъпва, управлява и актуализира
- Потребител – човек, който използва компютърна или мрежова услуга
- Сървър – стартирана инстанция на софтуерна система, която може да приема заявки от клиент и да връща подходящи отговори
- Клиент – част от компютърна или софтуерна система, която достъпва услуга, предоставена от сървър
- Приложение (application) – софтуер, предназначен да помогне на потребителя да извърши определена задача
- Интерфейс (interface) – споделена граница, между която два отделни компонента на компютърна система си обменят информация
- Модул – логически обособена софтуерна единица
- Процес – съвкупност от стъпки, която изгражда логическо действие и стига определена цел
- Декомпозиция – софтуерна структура, показваща как системата се разделя на отделни модули. Типовете елементи изграждащи тази структура са модули, а връзките между тях са от типа “X е подмодул на Y”.

1. Драйвери

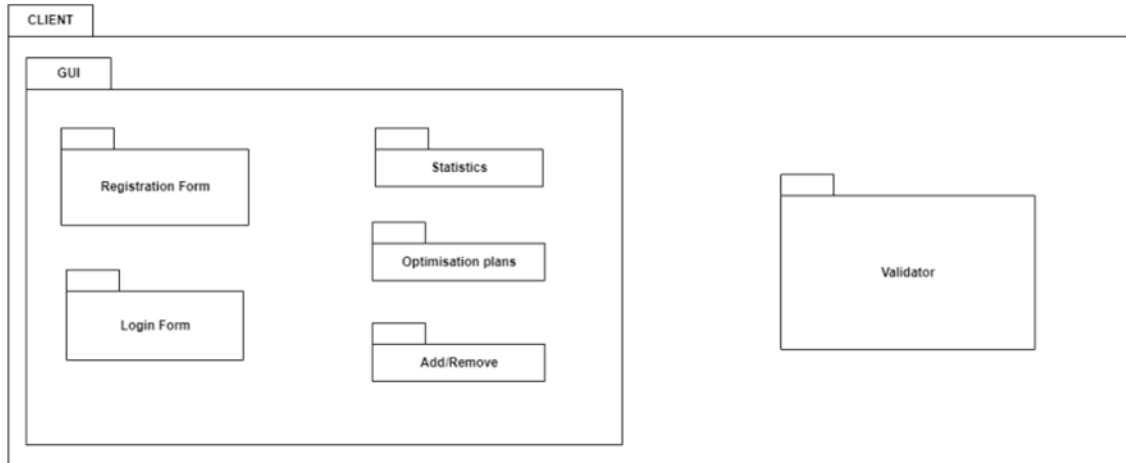
1. Определя ролите в системата и техните функции; изискване за основна функционалност и хардуер
2. Основно изискване по отношение на достъпността
3. Дава информация за наличието на профили на потребителите и управлението на фермите, пояснява ролята на потребителя и взаимодействието му със системата
5. Уникална регистрация на сензорите, основно изискване по отношение на функционалностите
7. Наблюдение в реално време; основно изискване по отношение на функционалностите; едно от изискванията, което определя, как ще се използват данните, получени от сензорите
8. Добавя функционалност за генериране на справки и анализи, основно изискване по отношение на функционалностите; едно от изискванията, което определя, как ще се използват данните, получени от сензорите
9. Дава информация за връзката с външни системи; изготвянето на метеорологична прогноза; основно изискване по отношение на функционалностите; едно от изискванията, което определя, как ще се използват данните, получени от сензорите
11. Дава възможност на потребителя да контролира поливането, основно изискване по отношение на функционалностите; едно от изискванията, което определя, как ще се използват данните, получени от сензорите
13. Основно изискване по отношение на сигурността на системата
17. Функционалност за добавяне на сензори

2. Модулна декомпозиция

Общ вид на декомпозиция на модулите на системата



1. CLIENT



2.1.1. GUI отговаря за представянето на съдържанието на системата при клиента

2.1.1.1. Registration Form - форма за регистриране на нов потребител в системата

2.1.1.2. Login Form - форма за вписване на съществуващ потребител в системата

2.1.1.3. Add/Remove - добавяне и премахване на сензорите, ферми и техните имена

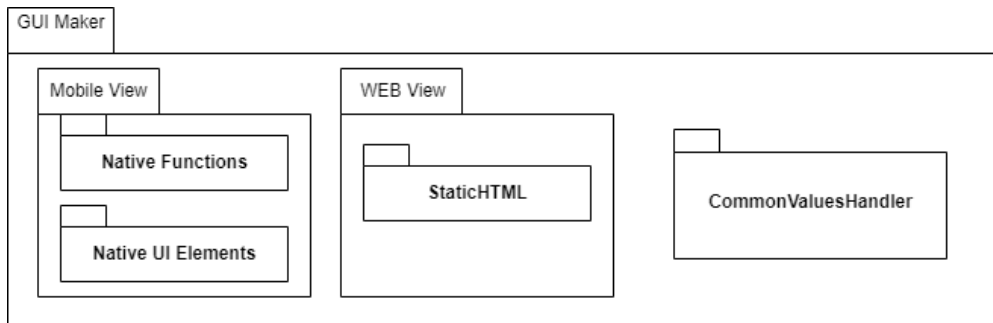
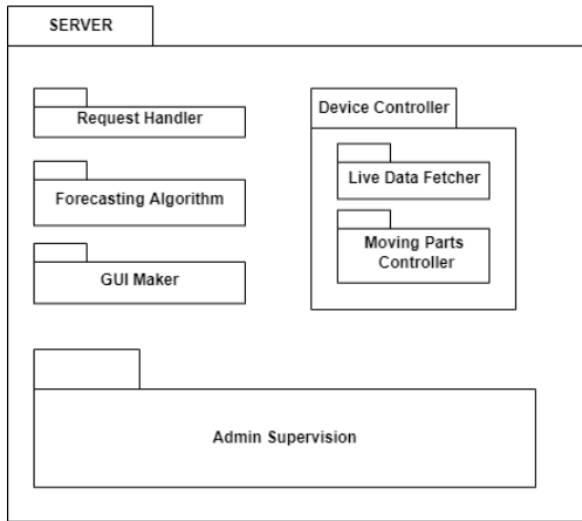
2.1.1.4. Optimisation plans – графично представяне на генерирания оптимален план за поливане

2.1.1.5. Statistics - според информацията събрана от измерванията на сензорите от всяка ферма, добавена от потребителя, се генерират справки и анализи в различни формати

2.1.2. Validator проверява различни входни данни подадени от клиента още преди да се изпратят към сървъра. Модула отговаря и за появяване на подходящо съобщение, ако входните данни не се валидират.

2.1.3. GUI Maker – отговаря за съставянето на различните интерфейси които предлага платформата като използва предварително направени шаблони със placeholders и попълващи се чрез CommonValuesHandler

2. SERVER



2.2.1. Request Handler - обработва заявките подадени от потребителя.

2.2.2. Forecasting Algorithm - предоставя прогноза за метеорологичните условия на района (вятър, атмосферно налягане, точни часове за изгрев и залез и др.) и спрямо нея оптимизиран план за поливане

2.2.3. GUI Maker - използвайки общи променливи позволява паралелното генериране и на двата формата на приложението (WEB/Mobile), спестявайки изчислителен ресурс и коректност на данните.

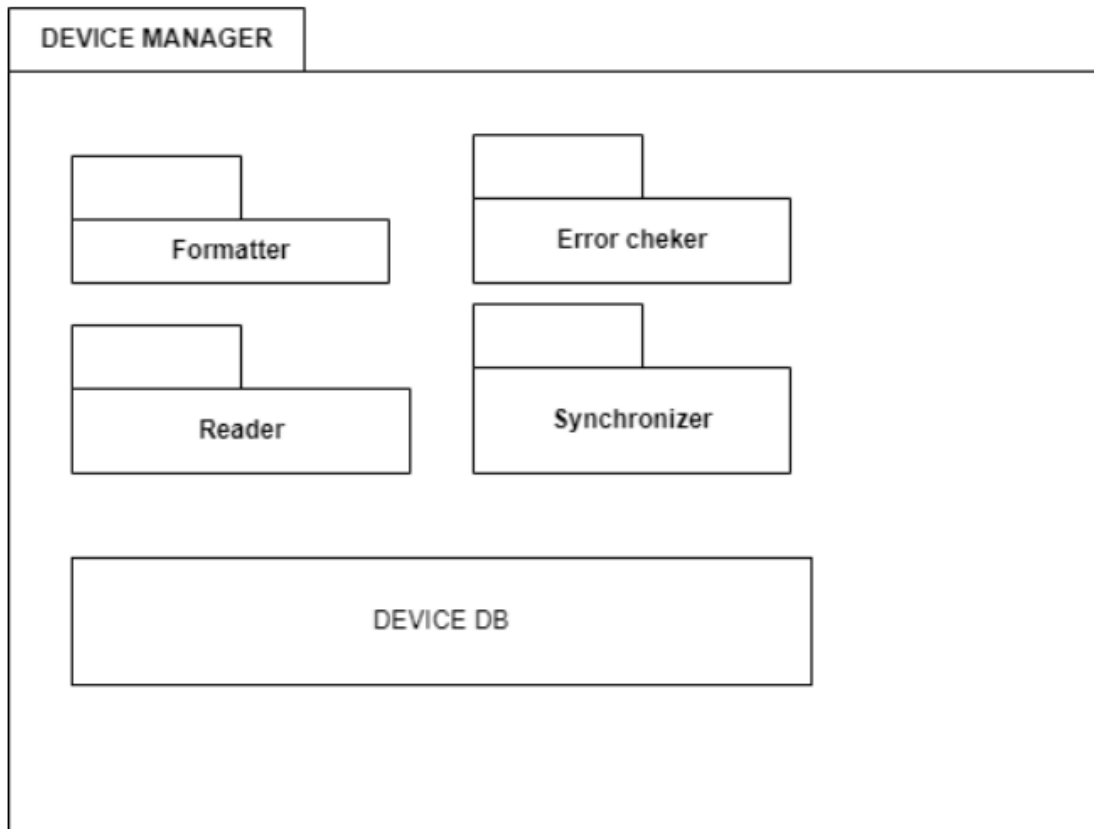
2.2.4. Device Controller – спрямо уеднаквеното представяне и сруктуриране на устройствата от SENSOR MANAGER извършва действия върху тях.

2.2.4.1. Live Data Fetcher – приема и обработва информация от сензорите (например: температура, влажност, химичен състав на почвата и др.).

2.2.4.2. Moving Parts Controller – контролира всички устройства, които имат движещи се елементи (кранове, бутала, шлюзове)

2.2.5. Admin Supervision - проверка за дубликати, опити за манипулация и пропуски в сигурността.

3. DEVICE MANAGER



2.3.1. Formatter – Форматира информацията от сензорите и ги привежда в общ вид

2.3.2. Synchronizer – Синхронизира времето за приемане и предаване от сензорите, с цел избягване на грешки

2.3.3. Reader – Използвайки утвърдени стандарти, позволява четенето на голямо разнообразие от сензори.

2.3.4. DEVICE DB – Собствена памет съхраняваща пътя за достъп за всеки от сензорите

3. Допълнителни структури

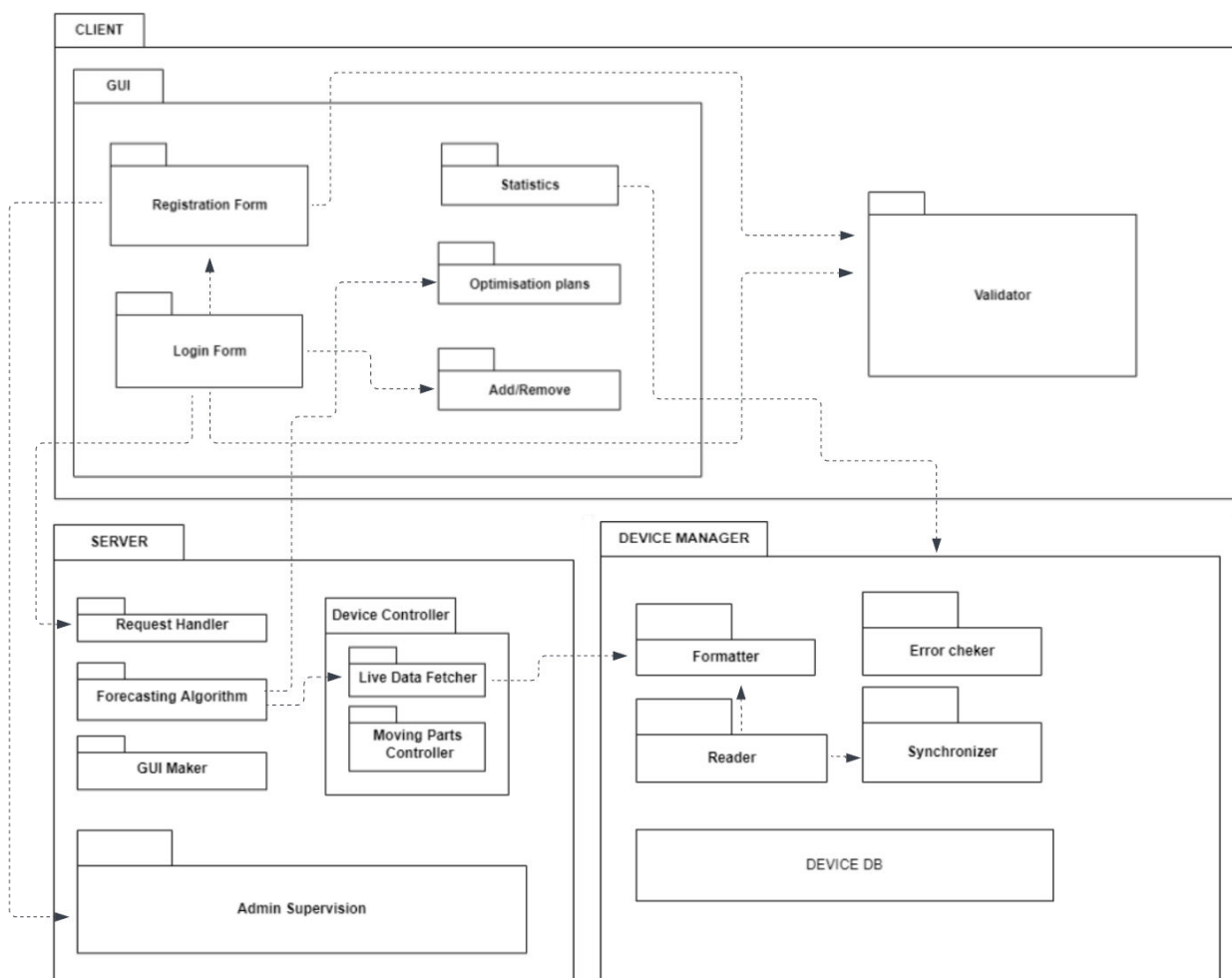
Употреба на модулите

Мотивация

Демонстрира нагледно статичните връзки между модулите и подмодулите от направената вече декомпозиция. Позволява по-нататъшна разработка и лесно добавяне на нови функционалности. Тази структура ще бъде използвана в работата на разработчиците, QA специалистите, софтуерните анализаторите.

Първично представяне

В структурата са изобразени основните елементи на системата (Client, Server, Device Manager), както и връзките между тях, описани по-долу. Стрелките обозначават взаимодействие между отделните модули(връзки), като посоката на стрелката пояснява кой от двата модула е източникът на заявката.



Описание на връзките

От Client до:

Client

- При регистрация на нов кандидат в подмодула Registration Form се проверява валидността на данните с модула Validation(например паролата да има поне 6 символа и с определено ниво на сложност, името да съдържа само разрешени символи и т.н.) изписвайки се подходящо съобщение, ако те не са коректни.
- След близане в профила потребителя може да извършва промени на сензорите, броя на фермите и техните имена

Подмодули в Server

- След влизане в системата потребителя подава заявки, които се обработват от Request Handler
- След извършена регистрация се осъществяват проверки за дубликати, опити за манипулация или пропуски в сигурността от Admin Supervision

Device Manager

- От събраната информация от модула Device Manager се съставят справки и анализи в различни формати

От Server до:

Server

- След уеднаквеното представяне и структуриране на устройствата от Device manager , Live Data Fetcher приема и обработва информация от сензорите (например: температура, влажност, химичен състав на почвата и др.), въз основа на които Forecasting Algorithm - предоставя прогноза за метеорологичните условия на района (вятър, атмосферно налягане, точни часове за изгрев и залез и др.)

Подмодул в Client

- и спрямо нея оптимизиран план за поливане

Подмодул в Device Manager

- Чрез Formatter, форматирайки информацията от сензорите, се съставя Forecasting Algorithm

От Device Manager до:

Device Manager

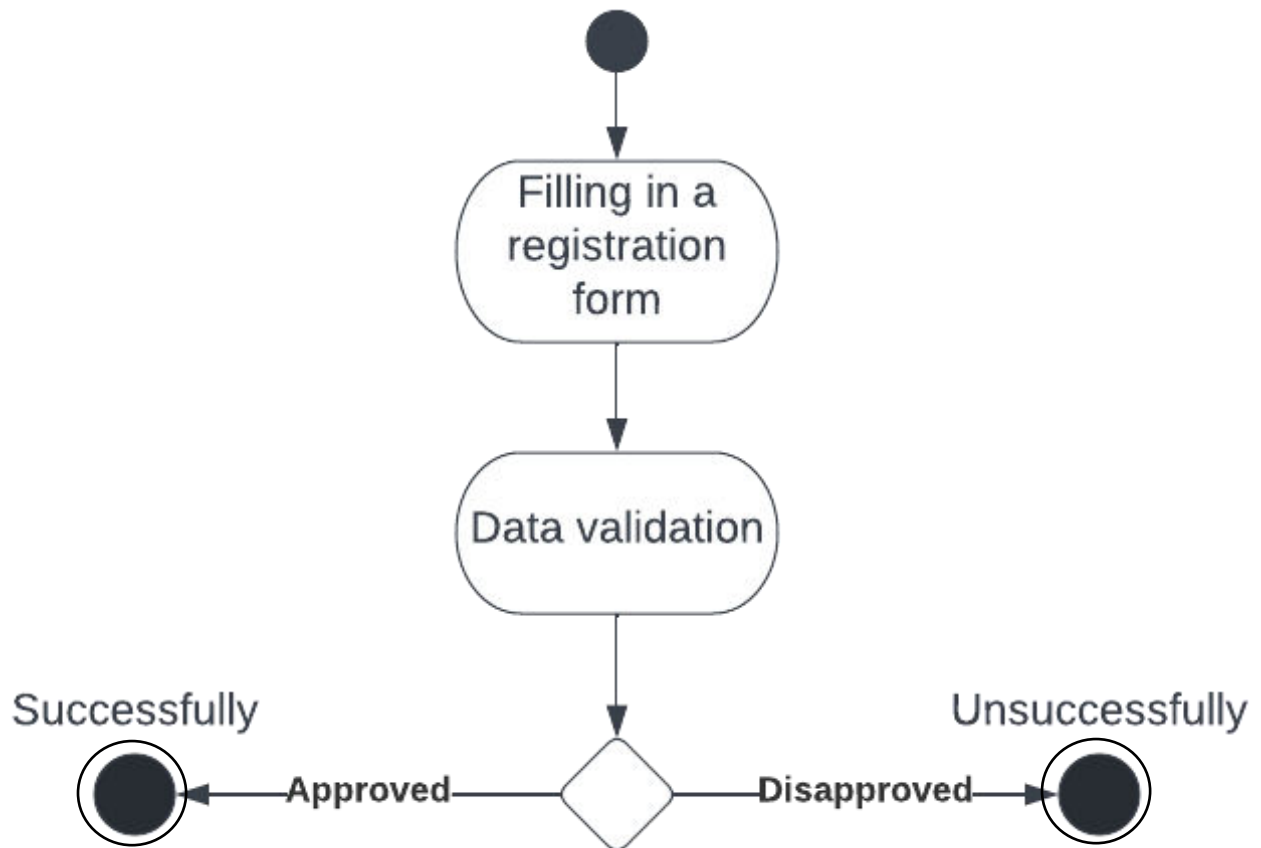
- Времето за приемане и предаване от сензорите се синхронизира с цел да се избегнат грешки, Formatter-а форматира информацията взета от Reader след синхронизацията

Структура на процесите

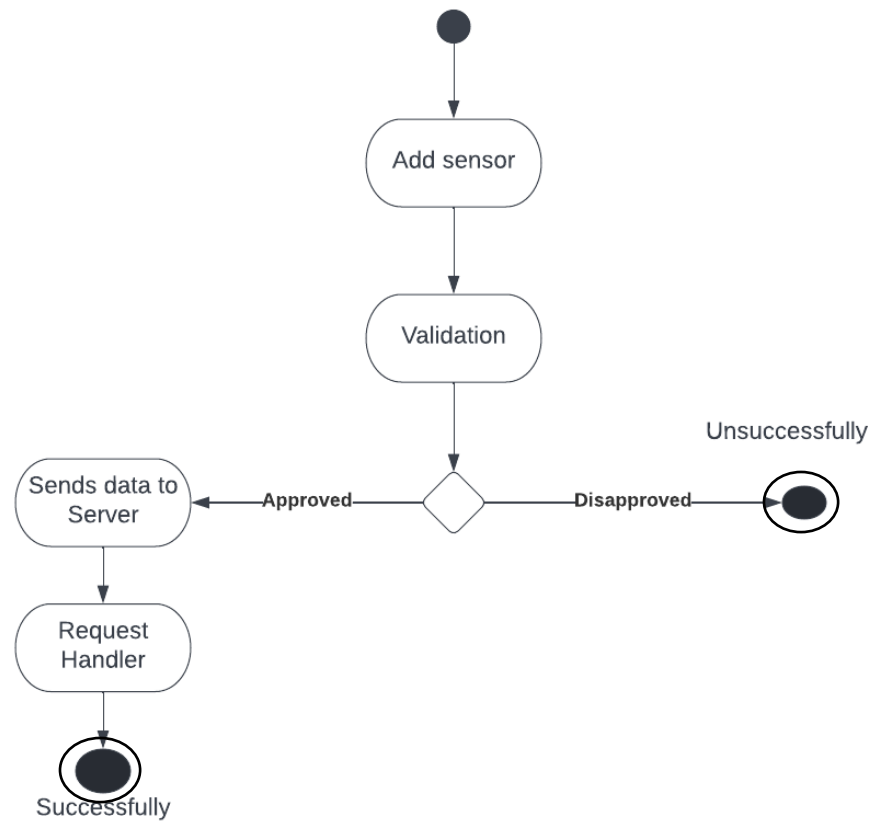
Мотивация

Структурата на процесите показва ясно последователността и връзката между процесите в системата. Структурата е полезна за нашата система поради подобряването на бързодействието по време на изпълнение и надеждността.

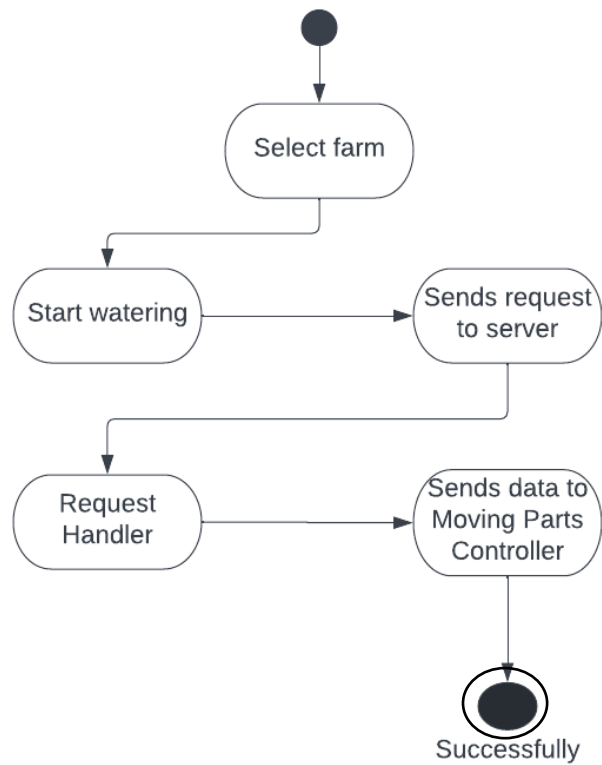
1. Регистрация



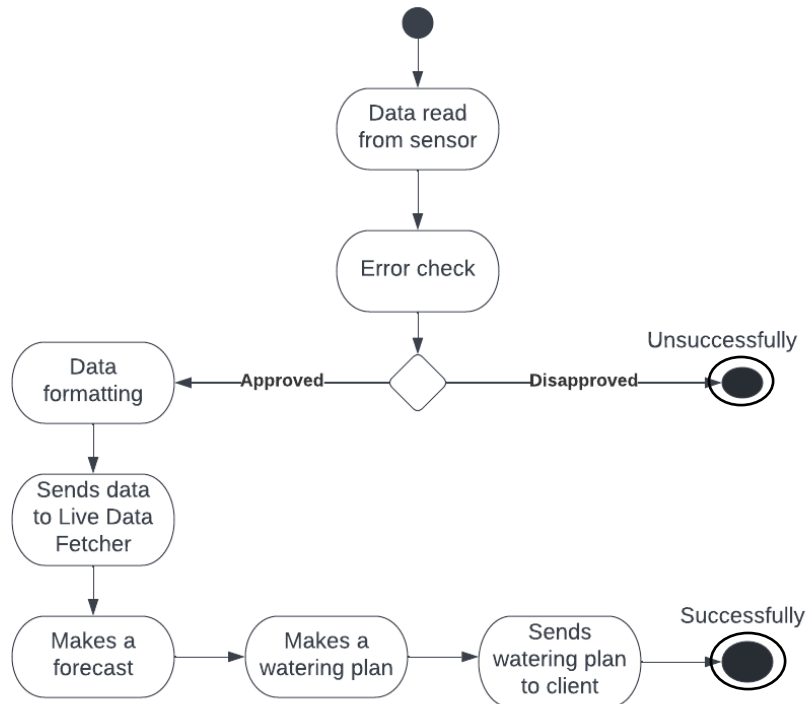
2. Добавяне на сензор



3. Стартиране на поливане



4. Планиране на поливане



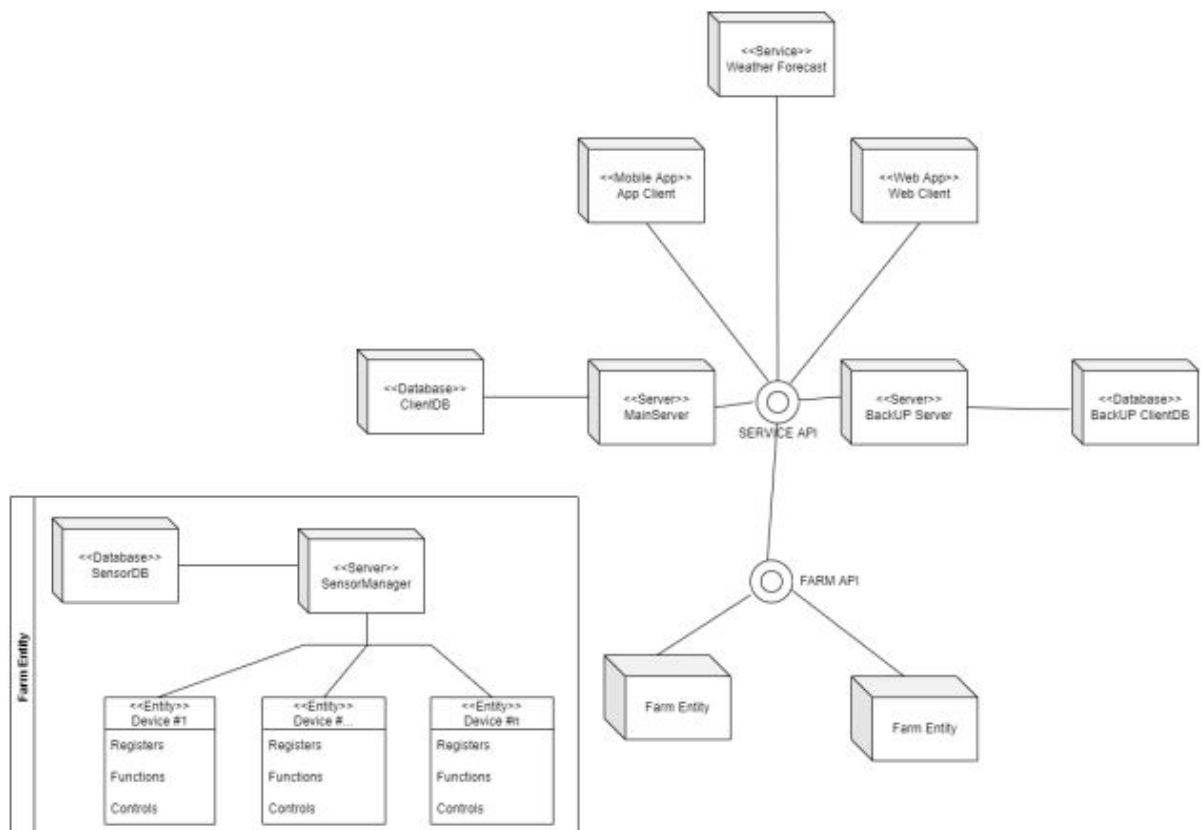
Описание на елементите и връзките

Започвайки от Client модула, при влизане в системата потребителят трябва да извърши регистрация през Registration Form модула, като валидаторът на входните данни проверява информацията и съответно я одобрява. След регистрацията се преминава през Login Form модула. В зависимост от избора, потребителя може да продължи към поливане(Optimisation plans), добавяне или премахване(Add/Remove) и справки и анализ(Statistics). При избор за добавяне или премахване, подадената информация първо се валидира и после се подава на Server модула, там тя се обработва от Request Handler. При изготвяне на план за поливане информацията от сензорите се събира и се проверява за грешки, след това се изпраща на сървър. Сървърът, след като е получил форматирана информация от сензорите и я е обработил, изготвя прогноза за времето, която използва за съставяне на плана за поливане и анализа. Когато потребителя реши да започне поливане той първо трябва да избере фермата, след това се изпраща заявка към сървър, която се обработва от Request Handler. На база плана за поливане сървърът управлява Moving Parts Controller.

Структура на внедряване

Мотивация за избор

Структурата на внедряване предоставя много значима информация, имайки предвид че нашата система е изградена от много хардуерни компонента, чрез структурата на внедряване предоставяме значима информация – комуникацията между отделните компоненти и тяхното позициониране.



Първично представяне

Основни елементи в структурата са сървъри, приложения и детайлно представените единици "Farm". Организирано е резервно копие на цялата система в случай, че главният сървър не е достъпен или при случай на непредвидени неблагоприятни обстоятелства. Приложенията биват десктоп и мобилни. Това е софтуер, който се инсталира на съответното устройство. Той бива използван от крайния потребител на системата. Приложенията може да се обръщат към сървърите чрез заявки за изпълняване на определено действие чрез SERVICE API. Използваме външна система за прогноза на времето.

Описание на елементите и връзките

MainServer – Основният сървър на системата който обработва, съхранява и модифицира съответно заявките на потребителите и информацията в системата. ClientDB – Основната база от данни съхраняваща информацията за потребителите и съответно ключовете към съответните „Farm“ единици обвързани със всеки клиент. BackUP Server – Изпълнява ролята на Main Server, в случай че не е достъпен BackUP ClientDB – Съхранява същата информация като ClientDB Service API – API чрез който се осъществява комуникацията между всички компоненти и чрез който се осъществява връзката между BackUP и Main сървърите. Mobile App – Мобилната версия на потребителското приложение Web App – WEB версията на потребителското приложение Weather Forecast – Външна услуга, която предоставя метеорологични прогнози. Farm Entity – Всяка отделна ферма ще я разглеждаме като отделна единица която комуникира със останалите елементи чрез FARM API, а компонентите в него са следните: SensorManager – Главния сървър на всяка ферма който изпраща и приема информация от главния сървър на системата SensorDB – Пази архив от стойности и измервания на сензорите, информация използвана в изчислителните алгоритми на MainServer N на брой Entity Sensor – Всеки сензор се разглежда като отделно Entity на под системата Farm

4. Архитектурна обосновка

Взимайки всяко от изискванията за проекта “AgroSmart” ще обосновем решенията взимани относно архитектурата на системата.

4.1. Системата трябва да поддържа следните потребители:

а. Обикновени потребители - фермери, които използват функционалностите на системата. Те получават необходимия хардуер (сензори, изпълнителни механизми и други) от доставчика на системата.

б. Администратори - представители на доставчика на системата, които се грижат за поддръжката.

Това изискване е изпълнено посредством Client-Server заявките от регистрационната форма описана в Client модула и Request Handler в Server модула за обикновените потребители, а отделен подмодул (Admin Supervision) е интегриран в Server, за да удовлетвори и втория вид потребител и функционалностите обвързани с него

4.2. Достъпът на потребителите до системата трябва да може да се осъществява или през браузър, или чрез мобилен клиент за iOS и Android.

Вкарвайки оптимизационен аспект в съставянето на различните изгледи се спряхме на решението да използваме native библиотеки елементи и функции за различните ОС на мобилното приложение. Също за уеднаквяване на информацията използваме placeholder-и, които динамично се доставят от сървъра чрез RequestHandler от страна на сървъра и CommonValuesHandler от страна на клиента. Същият подход е взет и при съставянето на web view.

4.3. Всеки потребител трябва да си направи профил в системата и може да управлява множество различни ферми (различни местоположения с различни инсталации) към профила си.

4.4. Потребителите да имат възможност да управляват следната информация в профила си:

а. личните данни и данни за контакт

б. детайли относно техните ферми, които включват имена на фермата, местоположение, “friendly” имена за устройствата, и други.

В Client модула има модул който се занимава с редакция на информацията в профила Add/Remove, а Registration Form/Login Form отговаря за създаването и достъпването на профила. Описани подробно в структурите на процесите.

4.5. Системата трябва да осигури възможност за уникална регистрация на съответните сензори така че да не се объркват данните със сензори от близки по местоположение други ферми.

4.6. Сензорите осигуряват измервания на атмосферната влажност и атмосферната температура за фермата, като едно централно устройство събира всички данни от тези сензори и ги изпраща към сървъра.

DEVICE MANAGER е модул, който изисква ситуиране на сървър за всяка ферма а модула Error Checker има освен функционалност да проверява коректно-работата на самите сензори, а и да сравнява идентификационната информация на сензора в базата данни на сензорите(DEVICE DB) възпрепятствайки прочитането на данни от сензор, който не е от съответната ферма

4.7. Сензорите осигуряват измервания на атмосферната влажност и атмосферната температура за фермата, като едно централно устройство събира всички данни от тези сензори и ги изпраща към сървъра.

Тук отново очакваме отговор в реално време, който ще бъде обработен от Request Handler-а комуникирайки с сървъра на фермата, чрез Farm API получавайки предварително подготвени, верифицирани и форматиращи данни от Farm Entity конструкцията.

4.8. Системата да дава възможност за генериране на справки и анализи в различни формати (например: pdf, xls, csv) според данните, подадени от сензорите, за определен период от време.

В Client подмодула Statistics ще извършва локално генериране на съответния анализ в желания от клиента формат премахвайки ненужен server load.

4.9. Системата да предоставя прогноза за метеорологичните условия на района (вятър, атмосферно налягане, точни часове за изгрев и залез и др.) за прогнозен период от няколко дни. Данните за прогнозата се взимат от външно приложение.

Използваме външно приложение за взимане на метеорологичните измервания, както и прогнозата знаейки, че тяхната реализация на тези проблемите ще е по-добра от другите алтернативи.

4.10. Потребителят да може да започва отдалечено и да спира поливането на площи във фермата (чрез браузър или чрез мобилното приложение).

Поради мерки за сигурност не позволяваме заявка от потребител да бъде изпълнена, освен ако не е преминала през MainServer защитните механизми и AdminSupervision, за това контрола на спирането и пускането се изпълнява отново чрез заявка към DeviceController който от своя страна комуникира с Farm Entity.

4.11. Потребителят може да създава план за автоматично поливане, което параметризира по: i) време и ii) минимални и максимални стойности на влажност и температура.

4.12. Системата да предоставя препоръки за оптимални параметри за поливането, спрямо анализ на данните, получени от сензорите и метеорологичната прогноза за следващите дни (от външното приложение).

Синергетично действие на модулите Client:OptimisationPlans и Server:ForecastingAlgorithm работят заедно децентрализирайки работата необходима за съставянето на плановете и препоръките.

4.13. Данните в сървъра на системата трябва да бъдат защитени от нерегламентиран достъп.

Освен стандартните сървърни функционалности за сигурност сме добавили специален модул AdminSupervision, в който ръчно могат да се предприемат действия, които не могат да бъдат взети категорично от автоматичните системи.

4.14. Ако има грешка при хардуерно устройство да се уведомят администраторите в рамките на 10 минути, със съобщение съдържащо id на устройството и името на съответния потребител.

DeviceManger има отговорност да отчита повреда при работата на сензор освен това са организирани регулярни рутинни проверки на всички сензори в базата данни на фермата.

4.15. Администраторите, при получаване на съобщение за грешка, се грижат за отстраняването ѝ, ако става въпрос за софтуер, или се обръщат към екипа за поддръжка на устройствата, отговарящ за района. Системата да дава възможност за специализирана връзка между администраторите и екипа по поддръжка.

AdminSupervision тук отново намира своето приложение

4.16. Допуска се профилактика веднъж годишно. През останалото време, системата трябва да е 99.9% налична.

Не случайно е инвестирано в BackUp Server , който изпълнява функцията на MainServer при повреда или по време на профилактика, като трафика бива пренасочван в текущия активен сървър.

4.17. Архитектурата трябва да позволява лесно добавяне на нови видове сензори (например за осветеност и т.н.).

Добавянето на сензор чрез DeviceManagera, не зависимо от типа му е невъзпрепятствано поради синхронизиращите и форматиращите подмодули в модула.