

**„Софийски Университет  
Свети Климент Охридски“**



**Проектиране на Човеко-Машинен Интерфейс  
2022-2023г.**

**Задание 4: Анализ на потребностите**

**Автори на проекта:**

1. Алекс Георгиев, фак. № 62604, III курс, Софтуерно инженерство
2. Павлин Димитров, фак. № 62614 , III курс, Софтуерно инженерство
3. Емил Цанев, фак. № 62620, III курс, Софтуерно инженерство
4. Лъчезар Кеманов, фак. № 62615, III курс, Софтуерно инженерство

**Име на софтуерен проект: RemoteFarm**

## **1.Основни цели и задачи**

“RemoteFarm” е система, която подпомага всеки един фермер който я използва. Основната задача, която тя решава е поливането на растенията. Тя позволява отдалечено управление на цялата напоителна система която един земеделец използва. Потребителите на системата могат да използват готови планове за поливане, както и те самите да създават свои собствени такива като ги настройват според тяхните специфични нужди и желания.

Ролята на потребителите е да имат различен достъп до различни функционалности предоставени от системата. Идеята е че потребителите ще могат да заплащат различни суми и ще получават различни функционалности. Като колкото повече се заплати толкова повече ще му бъде дадено на потребителя. Ще има възможност за четене на статии по теми за земеделие. Също ще може да се наблюдават данни в реално време от сензорите на полето.

Основната цел на проекта е да автоматизира и оптимизира процеса по отгелждане на земеделска продукцията, като това намали разходите и времето за производство.

## **2.Целева група потребители**

### **2.1 Регистриран потребител (Без пакет)**

Този вид потребители ще имат достъп до много ограничена част от системата. Понеже те ще могат да използват системата без да заплащат.

### **2.2 Регистриран потребител (Иконимичен пакет)**

Потребителите които са заплатили “Икономичен пакет” ще имат достъп до почти цялата функционалност която се предоставя.

### **2.3 Регистриран потребител (Плюс пакет)**

Заплатилите “Плюс пакет” ще разполагат с всичко което се предоставя от системата.

## **3.Анализ на нуждите на потребителите**

- Отдалечено управление на поливането (пускане или спиране)
- Автоматизирано поливане
- Възможност за финна настройка на поливане
- Препоръки за поливане
- Достъп до статии за земеделие
- Прогноза за времето

## **4. Изисквания към системата**

### **Функционални изисквания**

- Регистриран потребител (Без пакет)
  1. Прогноза за времето
  2. Плащане с банкова карта
  3. Преглед на данни от сензор
  4. Обновяване на системата и софтуера на сензори
  5. Добавяне на устройство към системата (само 1)
  6. Премахване на устройство
- Регистриран потребител (Икономичен пакет)
  1. Базовите функционалности
  2. Преглед на данни от сензори
  3. Създаване на план за поливане
  4. Пускане и спиране на дадени устройствата
  5. Пускане и спиране на цялата напоителна система
  6. Добавяне на устройство към системата (до 10 устройства)
  7. Обновяване на системата и софтуера на сензори
- Регистриран потребител (Плюс пакет)
  1. Функционалността на икономичния пакет
  2. Неограничен брой устройства
  3. Статистика от сензорите за последната година

### **Нефункционални изисквания:**

- Системата е активна през 99.9% от времето
- Системата трябва да предоставя сигурност на клиентите и да гарантира защитата на лични данни (имена, пароли, мейли, банкови детайли)
- Системата трябва да има инстанции в различни зони на света, за да спомага бързодействието и достъпността.
- Ако една от инстанциите падне, потребителите трябва да бъдат прехвърлени към най-близката географски зона, без да има down time
- Активиране на напоителната система трябва да се случва до 30 секунди след избиране на опцията, през приложението
- Системата трябва да може да поддържа до 1000 заявки в секунда
- Информацията, предоставена от клиентите не трябва да бъде губена, дори и при срив. Трябва да има резервно копие.
- Системата трябва да работи на различни операционни системи
- Потребителите трябва да могат да направят всичко, което системата предлага с до 15 клика

## **5.Списък на случаи на употреба**

### **5.1.USE-CASE: Преглед на данни от сензори**

#### **Brief Description**

На потребителя се показва по подходящ начин данни от сензорите в реално време.

#### **Actor Brief Descriptions**

Потребител - човекът който желае да види данни от сензорите

Система – програмата която обработва заявките на потребителя

#### **Preconditions**

Потребителя да е влязъл в системата.

Наличен е поне един сензор

#### **Basic Flow of Events**

1.Потребителят влиза в секцията за живи данни от сензори.

2.Избира кои от наличните сензори да наблюдава.

3.Системата показва по подходящ начин данните.

Потребителският случай е приключил успешно

#### **Alternative Flows**

##### **1.Няма налични сензори**

Ако на стъпка 2 не се покажат имена на сензори: Системата ще покаже подходящо съобщение.

Потребителския случай ще приключи неуспешно.

#### **Post-Conditions**

Успешно завършване: Показани са успешно данни от сензорите.

Неуспешно завършване: Не са показани данни от сензори, а подходящо съобщение.

#### **Special Requirements**

## **5.2.USE-CASE: Създаване на план за поливане**

### **Brief Description**

Потребителя определя в кои часове да се полива, кои устройства да поливат и колко интензивно да поливат.

### **Actor Brief Descriptions**

Потребител – човекът който иска да си създаде план за поливане.  
Система - програмата която обработва заявките на потребителя.

### **Preconditions**

Потребителя да е влязъл в системата.

### **Basic Flow of Events**

- 1.Влизане в секцията за създаване на план за поливане
- 2.Наименуване на плана за поливане
- 3.Задаване на времеви интервал
- 4.Задаване на кои устройства да работят в избрания времеви интервал
- 5.Задаване на интензивност
- 6.Финализиране на план.

Потребителския случай е приключил успешно

### **Alternative Flows**

#### **1.Повече от един интервал от време**

Ако на стъпка 6 потребителя желае да добави още един интервал от време то той не натиска бутона за финализира, а натиска бутона за добавяне на нов интервал и се връща на стъпка 3.

(Потребителския случай е завършил успешно)

#### **2.Няма устройства**

Ако няма устройства които да поливат то плана няма да се създаде.

(Потребителския случай е завършил неуспешно)

### **Post-Conditions**

Успешно завършване: Планът е записан успешно в системата.

Неуспешно завършване: Планът не е записан в системата.

### **Special Requirements**

[SpReq 2-1] Планът за поливане трябва да има уникално име.

## **5.3.USE-CASE: Препоръки за поливане**

### **Brief Description**

Системата предоставя на потребителя готов план за поливане.

### **Actor Brief Descriptions**

Потребител – човекът който иска да получи препоръки за поливане  
Система – приложението което изпълнява заявките на потребителя

### **Preconditions**

Потребителя да е влязъл в системата.

### **Basic Flow of Events**

- 1.Влизане в секция с препоръки за поливане.
- 2.Избор на локация за която ще се изготви препоръката.

### **Alternative Flows**

### **Post-Conditions**

Потребителят може да избере препоръчания план, ако му е харесал.

### **Special Requirements**

## **5.4.USE-CASE: Пускане и спиране на устройства**

### **Brief Description**

Потребителят решава кои устройства да бъдат спряни или стартирани.

### **Actor Brief Descriptions**

Потребител – контролира кога да спре или пусне устройствата през системата

Системата – спрямо указаните инструкции извършва пускане/спиране на устройствата

Устройства – инсталираните устройства, които ще бъдат пуснати/спряни

### **Preconditions**

Потребителя да е влязъл в системата

Потребителят да има инсталирани устройства

### **Basic Flow of Events**

- 1.Влизане в секцията Irrigation management
- 2.Избиране на група от устройства, които искат да бъдат пуснати или спряни
- 3.Селектира избраните устройства

Потребителския случай е приключил успешно

### **Alternative Flows**

#### **1.Всички устройства могат да бъдат контролирани**

Ако на стъпка 2 потребителят иска да пусне или спре всички налични устройства, той може да го направи чрез бутоните Stop all/Start all

(Потребителския случай е завършил успешно)

## **2.Няма устройства**

Ако няма устройства които да поливат, то потребителят няма да има лист с показани устройства в секцията Irrigation management

(Потребителския случай е завършил неуспешно)

### **Post-Conditions**

При успешно завършване потребителят е пуснал/спрял своите устройства за поливане. След това обаче той трябва да контролира отново тяхното пускане и спиране.

### **Special Requirements**

Не

## **5.5.USE-CASE: Плащане с банкова карта**

### **Brief Description**

Заплащане на абонамент чрез кредитна/дебитна карта през системата.

### **Actor Brief Descriptions**

Потребител – плаща за абонамент чрез банкова карта

Системата – приема детайлите за плащане и ги обработва

### **Preconditions**

Потребителя да е влязъл в системата

Потребителят да предостави правилни детайли за банковата карта

### **Basic Flow of Events**

1.Влизане в секцията за плащане

2.Въвеждане на правилни детайли – име, номер на карта, дата на изтичане и т.н.

3.Заплащане на нужната сума

4. При нужда потребителят трябва да потвърди плащането в банката си

Потребителския случай е приключил успешно

### **Alternative Flows**

#### **1.Потребителят иска да плати чрез Apple Pay**

Вместо въвеждане на детайлите за плащане, потребителят може да плати директно с Apple Pay

(Потребителския случай е завършил успешно)

#### **2.Няма достатъчно средства в сметката на клиента**

Ако клиентът няма достатъчно средства в сметката, транзакцията няма да бъде успешна.

(Потребителския случай е завършил неуспешно)

### **Post-Conditions**

При успех, клиентът вече е обновил абонамента си.

### **Special Requirements**

Потребителя да има достатъчен баланс по сметката, за да плати

## **5.6USE-CASE: Пускане и спиране на напоителна система**

### **Brief Description**

Потребителят решава на коя нива да бъде пусната/спряна напоителната система.

### **Actor Brief Descriptions**

Потребител – контролира напоителната система

Системата – приема заявките от клиента относно пускане/спиране на напоителната система

### **Preconditions**

Потребителя да е влязъл в системата

Потребителят да има инсталирана напоителна система на поне една нива

### **Basic Flow of Events**

- 1.Влизане в секцията за поливане
- 2.Избиране на нивата, която иска да бъде поливана
- 3.Пускане на напоителната система

Потребителския случай е приключил успешно

### **Alternative Flows**

**1.Потребителят иска да пусне/спре напоителната система за всички ниви**

Вместо селектиране на отделни ниви, може да цъкне Start All/Stop All

(Потребителския случай е завършил успешно)

**2.Клиента няма нива, на която да пусне/спре напоителната система**

Ако клиентът няма нива, на която да контролира напоителната система, на стъпка 2 листът с ниви ще е празен

(Потребителския случай е завършил неуспешно)

### **Post-Conditions**

При успех, клиентът вече пуснал или спрял напоителната система

## **5.7.USE-CASE: Прогноза за времето**



### **Brief Description**

Потребителят вижда прогноза за времето, за територията, в която е разположил своите сензори.

### **Actor Brief Descriptions**

Потребител – лицето, което се интересува от прогнозата за времето

Система – Агрегира информация от външен, надежден източник и я показва по подходящ начин.

### **Preconditions**

1. Потребителят е влязал в системата.
2. Налице са данни за прогнозата за времето

### **Basic Flow of Events**

1. Потребител влиза в системата
2. Навигира до секцията с прогноза за времето
3. Ако има сензори в дадена локация, се показва прогноза за нея.
4. Може да избере различна локация, ако желае

### **Alternative Flows**

1. Потребител влиза в системата
2. Навигира до прогнозата за времето
3. Ако няма сензори, прогнозата се показва за неговото местоположение
4. Може да избере различна локация, ако желае

### **Post-Conditions**

Потребителят е получил информация за времето, за региона, който го интересува

### **Special Requirements**

Приложението да има достъп до локацията на потребителя, за да определи автоматично локацията му.

## **5.8.USE-CASE: Добавяне на устройство към системата**

### **Brief Description**

Потребителят добавя ново устройство, след което може да го контролира през системата и да следи информацията от сензорите му.

### **Actor Brief Descriptions**

Потребител - лице, с профил в системата, желаещо да добави ново устройство към

нея

Системата - Прави връзка между устройството и профила на клиента

### **Preconditions**

1. Потребителят има профил в системата.
2. Устройството е съвместимо със системата

### **Basic Flow of Events**

1. Потребителят влиза в профила си
2. Отива на секцията с устройства в приложението
3. Избира опция "Добави устройство"
4. Отваря се страница, която иска информация за новото устройство
5. Потребителя въвежда необходимите данни
6. Потвърждава добавянето на ново устройство
7. Системата установява връзка с устройството

(Потребителския случай е завършил успешно)

### **Alternative Flows**

1. Потребителят влиза в профила си
2. Отива на секцията с устройства в приложението
3. Избира опция "Добави устройство"
4. Отваря се страница, която иска информация за новото устройство
5. Потребителя въвежда необходимите данни
6. Потвърждава добавянето на ново устройство
7. Системата не може да се свърже с устройството
8. Информативно съобщение е показано на потребителя, което му подсказва какъв е проблемът с неговото устройство и как може да го разреши

(Потребителския случай е завършил неуспешно)

### **Post-Conditions**

При успех ново устройство е добавено, към профила на клиента. То може да бъде използвано от него в зависимост от правата, с които разполага.

### **Special Requirements**

Устройството и приложението да са свързани към една и съща интернет мрежа.

## **5.9.USE-CASE: Премахване на устройство**

### **Brief Description**

Потребителят премахва устройство от системата, и вече не може да го контролира  
или да вижда информация и статистика за него

### **Actor Brief Descriptions**

Потребител - лице искащо да премахне устройство от системата

Устройство - хардуер, който е свързан с приложението и може да бъде контролиран

през него

Системата - Прави връзката между клиента и устройството

### **Preconditions**

1. Потребителя е регистриран
2. Устройството е добавено към профила на клиента

### **Basic Flow of Events**

1. Потребителя влиза в системата
2. Навигира до секцията с устройства
3. Намира устройството, което иска да изтрие
4. Клика бутона за изтриване
5. Системата иска потвърждение на действието
6. Клиента дава съгласие
7. Устройството е премахнато

### **Alternative Flows**

1. Потребителя влиза в системата
2. Навигира до секцията с устройства
3. Намира устройството, което иска да изтрие
4. Клика бутона за изтриване
5. Системата иска потвърждение на действието
6. Клиента отказва да изтрие устройството
7. Потребителят е върнат към списъка с устройства

## **5.10.USE-CASE: Достъп до статии**

### **Brief Description**

Потребителят отваря приложението и чете статии свързани със земеделие.

### **Actor Brief Descriptions**

Потребител - Лице искащо да прочете статия в приложението.

Система - Предоставя предварително написани статии на различни теми

### **Preconditions**

Потребителят е влязъл в приложението и заплатил за правилния пакет, даващ му достъп до статиите

### **Basic Flow of Events**

1. Потребителя влиза в приложението
2. Избира секцията за статии
3. Прилага филтри (заглавие/ дата на публикуване/ тема)

4. Избира статия
5. Прочита статията

#### **Alternative Flows**

1. Потребителя влиза в приложението
  2. Избира секцията за статии
  3. Няма достъп
  4. Подканен е да надгради сегашния си план, до такъв даващ му право да чете статии
- 
1. Потребителя влиза в приложението
  2. Избира секцията за статии
  3. Прилага филтри (заглавие/ дата на публикуване/ тема)
  4. Няма намерени статии
  5. Подканен е да изчисти филтрите за търсене

#### **Post-Conditions**

#### **Special Requirements**

## **6.Речник на термините**

**Без пакет** – потребителя използва системата безплатно.

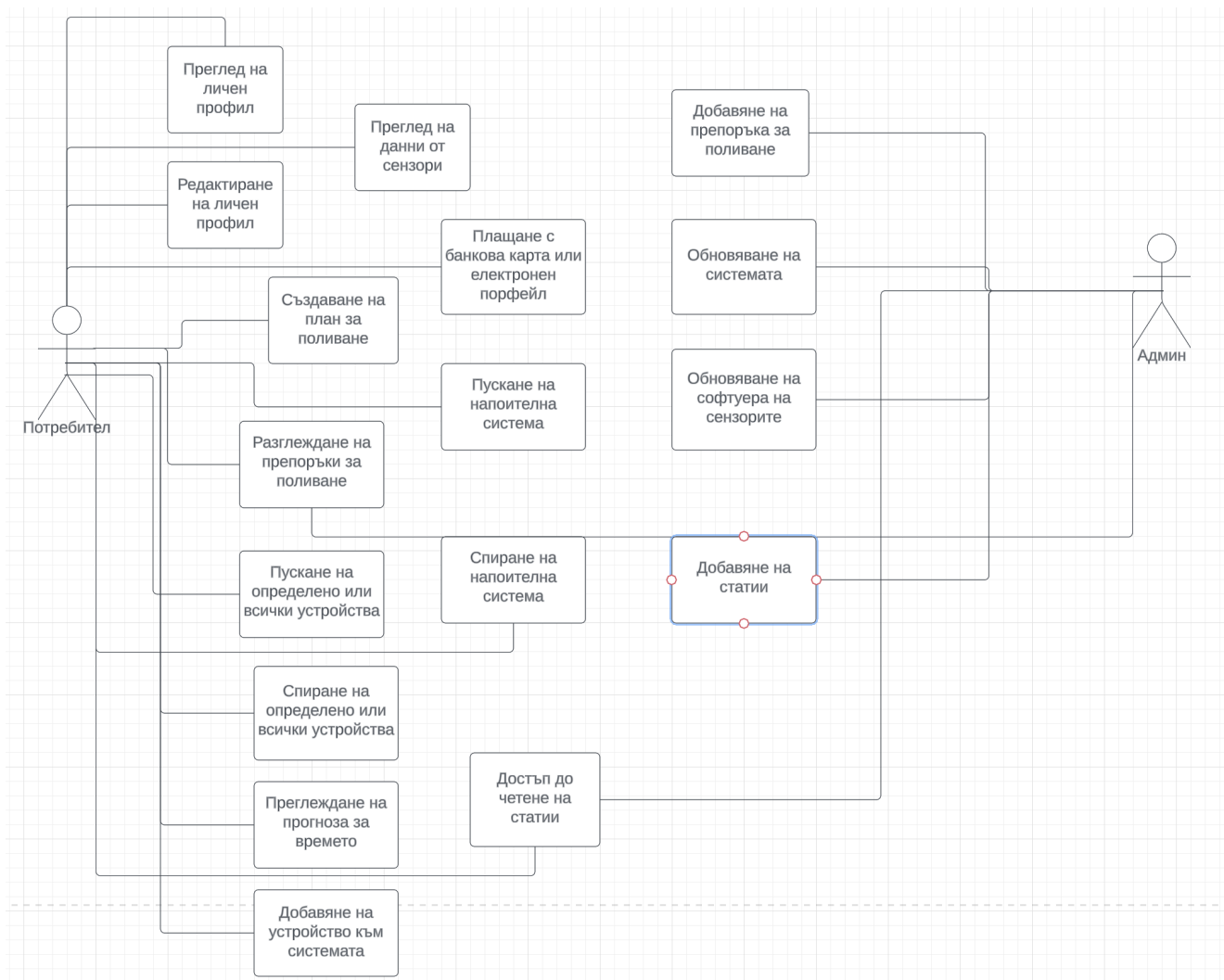
**Икономичен пакет** - минималната тарифа която може да се заплати.

**Плюс пакет-най** - високата тарифа която може да се заплати.

**Сензор** – устройство което може да определи дадени параметри като влажност и да ги изпрати.

**Живи данни** – данните които се измерват в реално време.

фиг.1 UML use case diagram



фиг.2 Activity diagram



фиг.3 Activity diagram

