Czech University of Life Sciences Prague

Faculty of Economics and Management

Department of Information Engineering

Diploma Thesis

System integration of farm information system

Andrey Khirnyy

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management

DIPLOMA THESIS ASSIGNMENT

Bc. Andrey Khirnyy, BA

Systems Engineering and Informatics
Informatics

Thesis title

System integration of farm information system

Objectives of thesis

The main objective of the thesis is to design a system integration procedure to improve the functionality of the farm information system. The design of the system integration process will be adapted to the conditions of the Agriculture of the Russian Federation.
The partial objectives of the thesis are:
- Create a literature review in the field of information systems and system integration.
- Create an analysis of user requirements and design a system architecture in a selected area.
- Synthesize the results of practical part and propose recommendations for agriculture practice

Methodology

Methodology of the thesis is based on study and analysis of information resources. In accordance with the intended purpose and objectives of the thesis, the author will use the following methods: analytical research, modeling, simulation, concretization and synthesis. On the basis of theoretical knowledge and author s own work, the conclusion of the thesis will be formulated.

The proposed extent of the thesis

60 – 80 pages

Keywords

System integration, agriculture, diagrams, integration plan, Russia, information system, system architecture

---

Recommended information sources

Ananyev M.A. Application of information technology in the agricultural sector / M.A. Ananyev, Yu.V. Ukhta. [Electronic resource] – URL: www.sisupr.mrsu.ru.

Matveev D.M. Technical and technological re-equipment of agriculture is necessary / A.T. Stadnik, D.M. Matveev, M.G. Krokhta, P.P. Kholodov // AIC: Economics, Management. – 2012. – No. 5.

Matveev D. M. The role of consulting activity in the technical and technological re-equipment of agriculture / A. T. Stadnik, D. M. Matveev, M. G. Krokhta, P. P. Kholodov; Novosib. state agrarian. un-t – Novosibirsk: Publishing House of the NSAU, 2013 .

Menyakin D.V. Information systems and their application in the agricultural sector / D.V. Menyaykin, A.O. Talanova // Young scientist. – 2014. – No. 3.

System Integration, Jeffrey O. Grady, CRC Press, Jul 8, 1994 – Technology & Engineering

---

Expected date of thesis defence

2020/21 SS – FEM

The Diploma Thesis Supervisor

Ing. Jan Tyrychtr, Ph.D.

Supervising department

Department of Information Engineering

Electronic approval: 7. 3. 2021

Ing. Martin Pelikán, Ph.D.

Head of department

Electronic approval: 7. 3. 2021

Ing. Martin Pelikán, Ph.D.

Dean

Prague on 29. 03. 2021

Declaration

I declare that I have worked on my diploma thesis titled "System integration of farm information system" by myself and I have used only the sources mentioned at the end of the thesis. As the author of the diploma thesis, I declare that the thesis does not break any copyrights.

In Prague on 31.03.2021 ______________________________

Acknowledgement

I would like to thank Ing. Jan Tyrychtr, Ph.D., for his constructive criticism and significant advice during the planning and development of this work. His professionalism and competence have been very much appreciated. I would also like to thank my family and friends for their invaluable support and faith in me. Special thanks to Boris Kirpichenko and Maxim Kirpichenko for their time and prompting in solving technical issues.

System integration of farm information system

Abstract

This thesis is devoted to the integration of modern technologies into the agricultural sector to optimize production processes and reduce the workload on staff.

The theoretical part is based on the study of scientific literature on information systems, information technologies and the integration of these technologies in the agriculture of the Russian Federation.

The practical part starts with the study and structuring of the requirements of Russian farmers to optimize facility for edible crops growing. The main part of the case study is creation of class model, state model, and interaction model to describe the architecture design and operation processes of the integrated system according to UML methodology. This UML model is then consistently supplemented with elements of the user interface specification: the information architecture design that is presented using card sorting method and wireframe in the form of several mockups. Author's recommendations are given in the final part which allow this work to serve as a practical guide for integrating a greenhouse control system in order to simplify the growing process.

Keywords: System integration, agriculture, diagrams, integration plan, Russia, information system, system architecture.

Systémová integrace zemědělského informačního systému

Abstrakt

Tato práce je věnována integraci moderních technologií do zemědělského sektoru s cílem optimalizovat výrobní procesy a snížit pracovní zátěž zaměstnanců.

Teoretická část je založena na studiu vědecké literatury o informačních systémech, informačních technologiích a integraci těchto technologií do zemědělství Ruské federace.

Praktická část začíná studiem a strukturováním požadavků ruských farmářů na optimalizaci zařízení pro pěstování jedlých plodin. Hlavní částí případové studie je vytvoření modelu třídy, státního modelu a modelu interakce k popisu návrhu architektury a provozních procesů integrovaného systému podle metodiky UML. Tento model UML je poté důsledně doplňován prvky specifikace uživatelského rozhraní: návrh informační architektury, který je prezentován pomocí metody třídění karet a drátového modelu ve formě několika maket. Doporučení autora jsou uvedena v závěrečné části, která umožňuje tuto práci sloužit jako praktický průvodce pro integraci systému kontroly skleníků za účelem zjednodušení procesu pěstování.

Klíčová slova: Systémová integrace, zemědělství, diagramy, integrační plán, Rusko, informační systém, architektura systému.

Table of content

Table of figures

List of tables

1. Introduction

More and more attention is paid to the quality of people's lives in modern society. The concept of life quality includes such an area as nutrition, which is of great importance for a comfortable life of a person, which further affects the development of society. Agriculture is one of the ways to provide people with food.

More and more technologies appear for the development of this industry. New planting systems are being developed for them, monitoring the growth process, equipment for assembly, processing, packaging, and storage.

Innovation in this area is the development of smart greenhouses or agricultural farms. It is aimed at obtaining a high-quality harvest, increasing the environmental friendliness of production, and reducing the number of resources spent.

The use of smart greenhouses in Russia is quite relevant and profitable. Increasingly, there are industrial agricultural complexes that resort to smart greenhouse technology. Even some enterprising citizens build similar structures on their plots or windowsills. The area occupied by plants could be used more productively, then a high-quality product would be sufficient not only for the domestic needs of the country but also for import abroad. And if the system had a low cost and ease of management, then it would be available to almost all potential users and would be used not only on an industrial scale but for private households.

2. Objectives and Methodology

2.1.1 Objectives

The main objective of the thesis is to design a system that can be integrated to improve the functionality of the farm that grows various edible crops on hydroponics. The design of this system is adapted for integration into an enterprise located in the territory of the Russian Federation. Various distinctive features and nuances of the Russian population and industry were taken into account for this. It is necessary to create a literature review in the field of information systems and system integration. The main goals are creation an analysis of user requirements and design a system architecture in a selected area, which will be accompanied by synthesize the results of the practical part and propose recommendations for agriculture practice.

2.1.2 Methodology

The methodology of the thesis is based on the study and analysis of information resources in the field of information systems and technologies, system integration, and the UML language. Following the intended purpose and objectives of the thesis, the author uses the following methods: analytical research, modeling, simulation, concretization, and synthesis. Based on theoretical knowledge and the author's own work, the recommendations and the conclusion of the thesis are formulated.

3. Literature Review

Information system - an information processing system and related organizational resources (human, technical, financial, etc.) that provide and disseminate information.

The information system is designed to provide the right people with the right information on time, that is, to meet specific information needs within a certain subject area, while the result of the functioning of information systems is information products - documents, information arrays, databases and information services.[6]

3.1.1 Classification of information technologies and systems

Information technologies can now be classified according to several characteristics, in particular: the method of implementation in the information system, the degree of coverage of control tasks, the classes of technological operations implemented, the type of user interface, the options for using the computer network served by the subject area.

Consider the relationship between information systems and information technology.

Management is the most important function, without which the purposeful activity of any socio-economic, organizational, and production system (enterprise, organization, territory) is inconceivable.

The system that implements the control functions is called the control system. The most important functions implemented by this system are forecasting, planning, accounting, analysis, control, and regulation.

Management is associated with the exchange of information between the components of the system, as well as the system with the environment. In the process of management, information is obtained about the state of the system at each moment, about the achievement (or not achievement) of a given goal to influence the system and ensure the implementation of management decisions.

The agrarian information system is a set of internal and external flows of direct and feedback information communication of an agricultural object, methods, means, specialists involved in the process of information processing and the development of management decisions. [7]

An automated information system is a collection of information, economic and mathematical methods and models, technical, software, technological tools and specialists, designed to process information and make management decisions.

INFORMATION TECHNOLOGY	By the method of implementation in information systems	Traditional
		New information technologies
	By the degree of coverage of management tasks	Electronic data processing
		Automation of control functions
		Decision support
		Electronic office
		Expert support
	By the class of implemented technological operations	Working with a text editor
		Working with a tablet processor
		Working with DBMS
		Working with graphic objects
		Multimedia systems
		Hypertext systems
	By user interface type	Batch
		Dialogue
		Network
	By the way the network is built	Local
		Multilevel
		Distributed
	By subject areas served	Tax activity
		Accounting
		Insurance activity
		Others

Table 1 Classification of information systems (Source: [7])

Thus, an information system can be defined from a technical point of view as a set of interrelated components that collect, process, store and distribute information to support decision making and management in an organization. In addition to supporting decision making, coordination, and control, information systems can also help managers conduct problem analysis, make complex objects visible, and create new products.

Information systems contain information about significant people, places, and objects within an organization or in the environment. We call information data that is transformed into a form that is meaningful and useful to users. In contrast, data are streams of raw facts that represent results that occur in organizations or the physical environment before it has been organized and transformed into a form that users can understand and use.

Information can be divided into external and internal according to the sources of receipt. External information consists of directives from higher authorities, various materials from central and local government bodies, documents from other organizations and related enterprises. Internal information reflects data on the progress of production at the enterprise, on the implementation of the plan, on the work of shops, service sections, on the scale of production. [10]

All types of information required to manage an enterprise are an information system. The control system and the information system at any level of control form a unity. Management without information is impossible.

Three processes in an information system produce the information that organizations need to make decisions, manage, analyze problems, and create new products or services — input, processing, and output. Unverified information is captured or collected from within the organization or from the external environment during the input process. This raw material is transformed into a more meaningful form during processing. The processed data is transferred to the personnel or processes where it will be used during the output stage. Information systems also need feedback, which is then returned processed data, needed to accommodate the elements of the organization to help evaluate or correct the processed data.

There are formal and informal organizational computer information systems. Formal systems rely on accepted and ordered data and procedures for collecting, storing, producing, distributing, and using that data.

Informal information systems (such as gossip) are based on implicit conventions and unwritten rules of conduct. There are no rules on what information is or how it will be collected and processed. Such systems are essential for the life of the organization. They have a very distant relationship to information technology. [22]

Although computer information systems use computer technology to process unverified information into meaningful information, there is a tangible difference between a computer and a computer program on the one hand, and an information system on the other.

Electronic computers and programs for them are the technical basis, tools, and materials of modern information systems. Computers provide equipment for storing and producing information. Computer programs, or software, are collections of service manuals that control the operation of computers. But computers are only part of the information system.

From a business perspective, an information system represents organizational and management decisions based on information technology in response to a challenge posed by the environment. Understanding information systems does not mean being literate in the use of computers, the manager must have a broader understanding of the essence of the organization, management, and technologies of information systems and their ability to provide solutions to problems in the business environment. [14]

3.1.2 Information systems of agricultural enterprises

The transition to digital technologies took place in all subject areas and spheres of life, including the agro-industrial complex (AIC). The process of digitalization in the agro-industrial complex began with accounting, and today this area is also actively robotized, an example of which is the development and implementation of unmanned agricultural machinery.

Technologies aimed at the outside world are actively developing: the integration of social networks, virtual organizations, and teams, e-commerce. Any enterprise can get remote access to software, databases to various services.

Initially, the information system was implemented to solve internal tasks, often disparate tasks. "Information islands" were formed at the enterprise, which require unification or integration and the creation of a single information space. The integration of new technologies into the information systems available at the enterprise is required with the development of technologies oriented to the outside world.

Thus, it is necessary to carry out internal and external integration for the system integration of the information system.[15] For objective reasons, the information system of farm lags behind other enterprises in its development, therefore, additional time and other resources are required to create an integrated information system.

The information system of an agricultural enterprise must have the following characteristics [19]:

- to fully take into account the natural conditions and the capabilities of the enterprise to realize the soil and climatic potential;

- to carry out the implementation of the connection of agricultural machinery with biological objects;
- to take into account and control a large number of parameters that can be geographically distributed;
- to implement accounting, control, and execution of processes and operations in plant growing and animal husbandry.

An agricultural enterprise has several features that determine the accounting procedure. [2]. The main feature is that the land is the means of labor. Also, the use of agricultural machines in production over large areas, which requires an accurate accounting of their activities. The production process is carried out outside the calendar year (settlements with suppliers are made against the future harvest). There is no clear relationship between the number of products produced and the cost in agriculture.

The information system "1C: Enterprise 8. Accounting of an agricultural enterprise" is proposed to automate the activities of accounting in the information technology market. It will provide accounting and tax accounting at agricultural enterprises. [1]

"1C: Accounting of an agricultural enterprise" solves the problems facing not only the accounting service of an agricultural enterprise but also an agronomist, livestock technician, accountant for personnel and wages, for accounting for land and property rent, for accounting for transport and agricultural machinery.

The use of information systems in agriculture based on geoinformation technologies (GIS) will improve the efficiency of agricultural production management. [18]

Databases are developed and used, containing electronic maps of farms where agrotechnical operations are carried out based on GIS. The databases contain information on the state of soils (humus content, soil characteristics), on the relief (information on the microrelief, the steepness of the slopes of their exposure), on crops, on agrotechnical measures in the fields. Databases are designed to provide managers of agricultural enterprises with information for making effective management decisions.

GIS allows to perform agrotechnical planning, namely, measuring fields, calculating the need for machinery and equipment, calculating the required amount, planning tillage operations, applying fertilizers and protective equipment.

Planning allows to build an effective work schedule for staff, reduce the cost of agrotechnical work and improve yield indicators.

It is convenient to monitor agricultural operations and the state of crops, analyze them, and display this information in the form of maps, tables, graphs with the help of GIS.

The process of forecasting crop yields and assessing losses is important for the functioning of an agricultural enterprise, which is based on methods of monitoring the state of crops, taking into account the influence of natural and climatic conditions. Forecasting is based on IS database data, built based on GIS.

Planning, monitoring, and analysis of the use of equipment will make it possible to effectively use agricultural equipment, which will reduce the cost of agricultural products. Planning provides for the construction of a schedule for the operation of equipment, technical inspections, and repairs. Monitoring and analysis allow the efficient use of fuels and lubricants.

GIS makes it possible to improve the processes taking place in the livestock sector. [18] These include grazing planning, prevention of desertification of pastures, degradation of the natural vegetation on pastures, erosion of the soil cover around watering places, and pollution by sewage from livestock complexes.

The use of GIS technologies will help the management staff to carry out remote control over the operation of the farm (to manage processes in real-time), as well as, to analyze the effectiveness of investments in production based on the reports received.

For the dispatch service, the use of these technologies allows to quickly track the location of equipment, coordinate the work of machine operators and drivers, incl. through the establishment of voice communication, as well as control the consumption of fuels and lubricants and the condition of the equipment.

<u>The automated workstation of an agronomist using GIS technologies:</u> [23]

- provides for maintaining the history of fields by yield, crops, applied fertilizers, and protection means;
- makes possible to plan fertilization taking into account the individual characteristics of the fields;
- provides information support in assessing the quality of work and developing proposals for their planning.

Geographic information systems help employees of the economic department to make a comparative analysis of planned and actual data, automate the recording of working hours, and the generation of reports and certificates.

GIS technologies are especially important in the management of agricultural production in regions with risky farming. These territories require constant control over the conditions for the development of crops and the conduct of agrotechnical and agro-chemical measures. Surveillance can be carried out both in individual fields and within a district, oblast, or wider area.

The development of a unified agricultural information system has begun in 2008 in the Russian Federation. [21] The main functions of this information system are:

- monitoring of the use of agricultural land and assessment of crop productivity;
- ranking of lands by monoculture;
- crop-rotation optimization;
- calculation of the need for mineral fertilizers;
- calculation of the yield of planned crops
- control of soil fertility.

The IS includes the following blocks: "State support and payment of taxes", "Economics and analytics", "Investments", "Soils", "Agricultural lands", "Production and processing, sales markets", "Satellite monitoring".

The IS of a farm is a set of functional subsystems:

- financial activities,
- production activities;
- fixed assets;
- staff;
- mechanized equipment. [22]

The creation, implementation, and operation of IS in farms are hampered by conservatism and aging of personnel, lack of necessary knowledge and low computer literacy, underdevelopment of information support services for agricultural enterprises. [8]

The process of creating a farm information system consists of several stages. The first stage is the creation of the material and technical base and the provision of the farm with the necessary software, the second is the formation of information services and the development of technologies that provide information support for the agro-business. The third is the integration of the farm information system into the agro-industrial complex information system and the global information process. [8]

3.1.3 Methods and technologies for integrating an enterprise information system

The problem of integration arises in any enterprise at the time of the implementation of the next corporate application. Three categories of problems can be distinguished during integrating IS: systemic, methodological and organizational. The systemic problems of IS integration include: [14]

- incorrect structuring of business processes;
- the difference in the approaches to describing duplicated business processes;
- the need for deep study of system-wide processes.

Methodological problems of IS integration:

- incorrect representation of the business processes of the enterprise;
- lack of information models of business processes;
- the presence of separation of external interaction processes from the IS.

Organizational problems of IS integration:

- insufficient coordination of performers - IS developers;
- lack of formulation of tasks for the integration of IS;
- lack of resources for solving the problems of integrating IS.

Factors influencing the integration of IS: [12]

- acceleration of processes occurring in the enterprise, which require changing data structures, business processes, design, and user interface;
- distribution - organizations are becoming large, geographically distributed, the tasks being solved are more and more complex;
- heterogeneity - enterprises use applications that are created using different platforms and tools;
- heredity - it is impossible to abandon obsolete technologies, old hardware;
- interactivity - increased requirements for the speed and efficiency of information delivery, most of the processes tend to be performed in real-time;
- mobility - the need to interact with users everywhere through network communication channels;
- high load - the complexity of integration depends on the intensity of the data processing flow, the amount of data, and the resource intensity of the calculations.
- continuity of the work cycle - the integration of IS should almost always be carried out without stopping the operation of the enterprise;
- Inter-system integration - the integration of IS is not limited to the framework of the enterprise.

There are many methods and technologies for solving the problems of IS integration.

Integration technologies classification:

Enterprise Applications Integration (EAI) systems are technologies for integrating systems, applications, and data within a single organization. EAI is an integration software structure, the purpose of which is to combine various kinds of applications that were developed independently of each other, so that they work as a whole, transparent to the user. At the same time, applications can use different technologies and remain independently manageable. EAI is a technology that integrates enterprise applications, integrating the data layer, interfaces, resources, etc.

Integration systems between organizations - technologies for safe, reliable information exchange between IS of different organizations.

Business Process Management (BPM) technologies, which are the result of the natural evolution of classic workflow systems, are based on the assertion that business processes should be the basis of integration. The business processes of a modern enterprise “cross” the boundaries of various applications, departments, and organizations.

Information systems integration methods [7]:

- integration at the broker level;
- integration at the interface level (physical, software, user);
- integration at the functional, application, and organizational levels;
- integration at the level of corporate software applications;
- integration using Web services;
- integration at the data level;
- integration at the service level;
- user-level integration.

The essence of integration at the broker level is the implementation of an additional software module that provides access to information systems via a database or using RPC (remote procedure call).

Horizontal and vertical integration is used when integrating IS. [17]

Horizontal integration provides for the integration of information systems or applications belonging to the same level,

Vertical integration is based on the integration of applications and systems located at different levels of the information pyramid.

The main approaches to building integration solutions are point-to-point, integration through the service bus. [3]

The point-to-point approach involves direct interaction between applications. The main objective of the approach is the implementation of the method for converting data from the format of the source system to the format of the receiving system. There are two methods for this approach. The essence of the first is that data reading and writing is performed directly into the application database. The second method involves the development of a special software interface. As a result, the integration solution is a unique software module that solves all the tasks associated with transforming data formats.

The point-to-point approach is effective with a limited number of integrations. Modern information systems contain a large number of integrations, which increase significantly with expansion. Each integration requires monitoring and support, which increases the cost of the integration.

The second approach is integration via a single service bus (ESB). A single service bus is a software application that ensures the interaction of all applications of the integrated system. It also provides users with a unified means of development, control, and testing of integrations.

The main components of the Service Bus are message brokers, adapters, development environment for integration scenarios, SOA tools, control, and management tools. The approach is widely used since it is scalable, simple, secure, uses open standards and centralized administration tools.

The functionality of the ESB environment should provide interaction with external entities (partners), which entails a lot of problems related to the quality of data transmission channels, with a limited range of data exchange formats, etc. [3]

3.1.4 Integration at the functional-application and organizational levels

Integration at the functional-application and organizational level leads to a redistribution of data and control flows, resources, and mechanisms since functions of the same type are combined into macro functions. This type of integration leads to a restructuring of the organizational structure of an enterprise, business processes, and a change in information support.

The advantages of this type of integration: the processes become more transparent, manageable, the costs of executing processes are reduced, the number of service personnel is reduced, and the quality of information is improved.

Integration at the functional level is designed to ensure the unity of goals, harmonization of criteria and procedures for the implementation of production, and economic processes and functions that ensure the achievement of the set goals.

The purpose of functional integration is to optimize the functional structure, decomposition of the system into subsystems, formalized description of their functional characteristics, and protocols of interaction between subsystems.

Organizational integration consists of creating a rational interaction of personnel, management at all levels of the hierarchy of the information management system, and various subsystems. Organizational integration predetermines the coordinated actions of personnel to achieve the set goals and the consistency of management decisions.

3.1.5 Integration at the interface level

Software applications were combined on a “one to one” basis with integration at the interface level. This made interoperability difficult, created problems when using legacy and embedded systems. This approach is used for small systems.

All the advantages of integrated systems are lost due to the complexity of performing aggregated queries to data with many applications. For example, the POSIX family of international standards is used to develop such unified POSIX interfaces. [10]

POSIX is a set of standards for describing interfaces between an operating system and an application program, a C language library, and a set of applications and their interfaces, designed to ensure compatibility of various UNIX kind operating systems and portability of application programs at the source code level.

The following algorithm is currently more efficient: the data processing layer is separated from the visualization forms associated with them and the applied business logic is implemented in one of the third-generation languages (3GL), the programmatic access to the applied functions is presented as a software interface. [10]

One of the simplest integrations is UI-based integration. [10] The essence of this type of integration is to provide the user with a unified interface for accessing various applications. The interface is a portal that provides single sign-on in integrated systems. The software can adapt content for devices of different formats, translate it into different languages and personalize it for each user.

3.1.6 Data Layer Integration

Integration of IS data - providing a unified interface for access to a set of heterogeneous data sources. [11]

The main problem of data integration is the abundance of formats and types of data, the rapid increase in data volumes. These problems are followed by problems associated with collecting, structuring, processing, analyzing, storing, archiving, and transferring to the user for making a business decision.

Data integration is the process of combining data from different sources to obtain their consistent presentation, as well as the process of organizing the exchange of data between different ISs.

IS data integration methods include file-based exchange, data replication, service-oriented technology (SOA), integration servers. [24]

File exchange is the simplest and most common integration method, but it has several disadvantages. Specialized file formats are being developed when it is necessary to exchange complex structures, which leads to a large dependence of systems from each other. File exchange requires to upload and download operations, which means that personnel is needed to carry out, record, and monitor these operations. File exchange is also performed using physical media in the absence of a network, and this reduces the reliability and security of the IS.

Data replication is the process of replicating IS data or bringing database data to an identical state. There are two categories of replication: replication of data between servers and between server and client. Replication between servers is performed to solve the following tasks. The scaling task, the essence of which is the creation of copies for reading. At the same time, duplication and redundancy of data are justified by an increase in the efficiency of the IS. In addition, the task of storing data, compiling reports, executing transactions, combining data from several nodes, and combining heterogeneous data are solved.

An integration system built on SOA principles is a set of software components (services) that are accessed over the network through service interfaces. SOA includes the Simple Object Access Protocol (SOAP).

An integration server is an intermediate server or gateway, the purpose of which is to process message data streams, as well as to distribute data between applications that have different interfaces. The core of the integration server consists of the rules according to which computational processing, analysis, and decision support are performed. The integration broker implements the interaction of applications with the integration server. Its purpose is to perform call operations, addressing via various interfaces, addressing, and converting data formats.

Two approaches are needed to address data integration: syntactic and semantic. Syntactic is the easiest to implement, as it is associated with content similarity. The semantic approach is based on a meaningful data component. The problem arose because the IS uses databases with different data models. The main question is in what form the information will be presented in a single information space.

The advantages of semantic data integration technology include:

- the simplicity of data exchange between ISs;
- the exchange procedure is not violated when changing the structure of information in the IS;
- independent configuration of data exchange;
- the simplicity of forming analytical data slices based on data from various ISs.

Semantic data integration technology is based on the construction of metamodels based on domain ontologies. [16] Ontologies are a data model for representing knowledge about the subject area and relationships between objects in the subject area.

Metadata - data describing database data, which allows to create a single information field containing data for each IS and information about each. Another data integration approach is the linked data approach. Linked data is a method of publishing metadata on the Internet in such a way that it can be linked to other data. Data is published using RDF documents using a uniform resource identifier URI. The method is based on a virtual approach to the presentation of information and it is proposed to use the global information space.

Standard interfaces and protocols SQL and JDBC / ODBC, tools of relational databases (Relational Database - RD), end-to-end repositories - databases with an "add-on" containing information about artifacts and design objects, a superset of metadata dictionaries, data (Transparent Repository - TR) and modern data warehouses and factories (Data Warehouse, Data Factory - DW, DF) are used for data integration.

Such technologies create a user-friendly, unified environment for storing and using data. It will be discussed in more detail about systems for the collective use of information futher.

Data from various sources are integrated during the development and operation of IS, the following tasks arise and require the solution: coordination or creation of metamodels, coordination of classifiers and reference books, data unification. [4]

Data integration is provided by the software, but it is necessary to consider the organization of access to data from the point of view of their placement in database files. Common methods are consolidation, distribution, federalization.

The essence of the consolidation method is that data is selected from various sources and collected in one place. The distribution method copies data from one place to another.

Federalization provides a single virtual view of multiple data files. Each of the methods has several advantages and disadvantages, the choice of the method is the developer's decision. [4]

In addition to the listed tasks, the creation of an integrated data system requires solving the following tasks:

- development of the architecture of the data integration system;
- data model integration based on which a unified user interface is built;
- overcoming the heterogeneity of data sources;
- development of mechanisms for semantic integration of data sources.

One of the most popular data integration architectures is the mediated architecture. The mediator is responsible for maintaining a single user interface based on global and local data representation. There are two types of mediated architecture - Global as View and Local as View. The essence of Global as View is a global view of integrated data in terms of data sources specified in local views. The Local as View architecture assumes that the view for each of the local sources is defined in terms of the global views of the data sources [11].

The XML, SQL language extension is SQL / XML and XML platform XQuery query language standard is used for data model integration.

3.1.7 Integration at the level of corporate software applications

Integration problems at the level of corporate software applications are associated with the fact that software systems are complex distributed systems that are built on different platforms using different technologies. It is necessary to establish connections between the elements of the system for their integration, and this must be done almost manually for each specific case.

The process of identifying and analyzing relationships is a long and laborious process, for which there is no unified methodology.

The integration of information systems is closely related to the interoperability of software applications or the ability to interact with other applications. There are two types of interoperability approaches. The first type - structural, meaning structural coordination of elements, is the most common. [20]

The second approach is semantic, which is associated with establishing links between the meanings of the system. The semantic approach is implemented by applying the theory of artificial intelligence, machine learning methods using the "semantic network" paradigm. [20]

Resource Description Framework (RDF) technology is used to describe the semantics of IP resources. The main provision of this technology is a model, which is a set of facts and semantic relationships. RDF, a framework for working with structured metadata, is an XML application. RDF allows to encode, exchange, and reuse metadata. RDF additionally provides a means for publishing and human understanding of metadata, machine-processed vocabularies. RDF provides a model for describing resources. A resource is an object identified by a uniform identifier (URI) and having a set of properties and semantic relationships.

The IS integration process can be represented as the following algorithm. [16] Data schemas are extracted for each IS, and the correspondence of the entities of one IS to the entities of another IS is determined, and the rules for their transformation are also determined. This sequence of actions is typical for the establishment of structural interoperability.

It is necessary to use a constituent part of the metadata for semantic interoperability, namely the conceptual model of the domain, which is an add-on over the data schema and sets the relationship between the entities of the domain. The construction of a conceptual model is carried out based on the ontologies of the subject area, which are defined as a set of the vocabulary of entities of the subject area, and the network of connections between them. This approach allows the presentation of conceptual models in uniform terms, and this makes it possible to automate the process of analyzing relationships between entities.

The last step of the algorithm is the construction of a metamodel for integrable IS, containing conceptual models of each IS.

The implementation of this algorithm allows integration the IS.

Integration of software applications is also performed by transferring parameters from one software application to another. This transfer is done through a remote call to the interface functions of the software applications. Component Object Model (COM) and Common Object Request Broker Architecture (CORBA) technologies are used for remote procedure calls. [6]

COM is a technology that ensures independence from the programming language of all kinds of program objects that are provided by a software application to an external environment.

CORBA is a technology for building a software application from distributed objects developed using various programming languages.

It is convenient for corporate ISs to represent them as a set of COM components that are used repeatedly and simultaneously. These components create a universal software core, which is accessible to each software application through interface modules.

CORBA technology should be used to interface ISs developed in heterogeneous environments.

These technologies are part of Enterprise Integration Methodology - EIM technology. EIM technology provides for the integration of software applications based on the integration of business processes. [10]

3.1.8 Integration with web services

Integration using Web services or Web integration is a form, methods of processing, and presentation of information resources of an organization using Web technologies. Web-integration allows to effectively use the information system of the company, increases its manageability and reduces costs, controls its internal resources, and simplifies the interaction between its structural divisions. The essence of Web-integration - the user gets access to various corporate applications and all information of the organization stored in different databases, as well as to data from other sources. [5]

A Web service is a networking technology that provides Web standards-based interprogramming, identified by a URI string and based on the machine-readable WSDL format.

WSDL is a language for describing external interfaces for representing abstract resources. It is written in XML and is an XML document. WSDL defines a service from an abstract and concrete point of view. A service is defined in terms from an abstract point of view, sent and received messages, described in XML format in a form that is independent of the transport protocol. From a specific point of view, the WSDL service allows to define bindings to transport formats and the physical location.

Interaction with various systems is performed by exchanging SOAP messages. The transport infrastructure is HTPP. UDDI is used to publish and search Web services. [5]

Web services technology eliminates application interoperability issues and enables the development of scalable integration solutions. Web services can exchange data between applications and provide a variety of services.

There are several types of Web services. The first type is method-oriented Web services (or RPC Web services). The second is document-oriented or XML Web services. The third is resource-oriented or RESTful web services that provide access to remote resources using HTTP requests. [17]

Method-oriented Web service (or RPC Web service) is a service in which data exchange is performed using the SOAP protocol, XML message format, WSDL interface format, transport protocols - HTTP. FTP. SMPT. SOAP messages that are involved in the exchange of data between the client and the service have the following structure - the name and parameters of the called procedure, the result of the call. The disadvantage of this type of service is a close connection between the client and the service, therefore, it is necessary to change the client when changing the parameters of the called procedure.

Document-oriented or XML-Web-services - services, oriented to messages, provide low-level processing of XML-messages, which are defined by XML-schema, used format - SOAP, XML. The service has a weak relationship with the client, as possible changes only affect the XML schema, not the service description. At the same time, the complexity of the service implementation increases since it is required to analyze the message structure.

Resource-oriented or RESTful web services are services that provide access to resources using HTPP requests. The service is characterized by a REST architecture. Data transfer protocol - SOAP, service description protocol - WSDL. The service supports the following operations: GET, PUT, POST, DELETE. Operations are intended to perform actions on resources. The disadvantage of the service is the limitations that are imposed on the number of available operations and the presentation of resources.

Web services are characterized by a synchronous and asynchronous model of interaction with the client. The client is blocked after sending a request for synchronous interaction. The block is released after receiving a response. The essence of asynchronous communication is that the client's work is not blocked, and the response is processed after it is received.

<u>The following scenarios of interaction with a client are typical for a Web service:</u>

- one-way request;
- synchronous request-response;
- asynchronous request-response;
- RPC call;
- error message;
- request with confirmation;
- transfer through intermediaries;
- caching;
- additional functionality.

4. Practical Part

4.1.1 User requirements for smart farm information system

Many volumes of scientific research are devoted to the methodology of creating complex software systems. Without going into the intricacies of this process, it will only be pointed out that to create a system, it is necessary to plunge into the subject area, that is, it is necessary to imagine the work of the entire mechanism with which the software system will deal. It's no secret that programmers by hook or by crook are trying to move away from the morally difficult process of communicating with the customers of the system. However, the dream of a software project manager is to receive from the customer all the necessary specifications and descriptions of work algorithms, so that they can then be passed on to the programmer. Customers themselves often have difficulty imagining the operation of the future system and cannot answer the questions: what functions need to be automated, what algorithms will be used. It's no secret that users idealize the future system, believing that everything they dream about will be embodied in it (often without informing the developer about these dreams) and even a little more. The task of the system designer is to "get" all the necessary information from the customer even at the design stage before significant resources are invested in the creation of the code. And the unified modeling language UML can play an important role in this. So, the first step in creating a system is to determine the initial user requirements for this system.

Let's consider the main conditions taken into account when choosing an automatic system for controlling the microclimate of a smart farm.

1. The cost of the control system. This factor is always considered in the first place and, often, is decisive.

2. Modern automatic control systems consist of a control device, sensors, and auxiliary equipment.

Control device - microcontrollers and computers with embedded and application software, various sensors, and any method of displaying information on a screen or print.

Devices - relays, switching equipment, electric motors, solenoid valves.

Accessories - Power and control cables, junction boxes and housings, and mounting materials.

Obviously, for large greenhouses with a lot of technological systems and a variety of functional capabilities, the composition and number of components, and, accordingly, the cost of control systems, will be higher than for small greenhouses, which also have fewer systems and the requirements for them are simpler. Therefore, the cost of control systems is determined individually for specific greenhouses and analyzed in comparison.

3. The functionality of the control system. Almost all modern systems for automatic control of the microclimate of greenhouses have a sufficient set of functionalities necessary for controlling the microclimate, but differ in details:

a) the ability to automatically control the technological systems of heating, ventilation, irrigation, electric supplementary lighting, pH replenishment available in the farm;

b) the ability to provide simultaneous control of various nodes of the system;

c) the quality of maintaining the required climate in greenhouses, or the ability with the available means to achieve a minimum deviation from the setting of the supported climatic parameters: temperature, humidity, pH concentration, lighting.

d) the possibility of adjusting the set parameters of the microclimate in the greenhouse, depending on the state of the plants, time of day, year, and weather conditions to increase the yield in the greenhouse;

e) the possibility of coordinated with the needs of the farm management of auxiliary heating points, power generating plants, and energy-saving systems.

It should be noted that if for small greenhouses it is sufficient the possibility of automatic control of existing actuators, then for large greenhouses this possibility is considered as unconditional and additional control possibilities are discussed in boiler houses, power plants, etc. Advanced customers evaluate the quality of dynamic processes for controlling microclimate parameters; in the case of automatic control systems for small greenhouses, this is most often not required.

4. The degree of automation and management methods.

The word “Automated” in the name of any automated system for managing the microclimate of greenhouses is misleading for many inexperienced users, making them believe that the automated system itself will manage all processes in the greenhouse. It should be borne in mind that by the existing terminology: "automatic system" - a system capable of functioning without human participation; "Automated system" - a system that operates with human participation. Differences between different there are more automated control systems in terms of ways to achieve the set goals than similarities.

<u>The main conditions characterizing the functioning of the control system</u>:

5. The degree of automation of management functions.

The higher the degree of automation, the less operator intervention in control processes is required. Ideally, the agronomist needs to set the name of the variety of the planted crop, and all actions to control the devices to ensure the required climate will be done by the automatic control system. Really, in practice, an agronomist or the operator, at the direction of the agronomist, sets the setting schedules for maintaining the microclimate and operating modes of the equipment, which are automatically adjusted by the control processes, the state of plants, and external meteorological conditions. Systems in which an agronomist, like a great "guru", constantly conjures over the percentages of opening the vents or the degrees of the coolant are weak systems that require constant and qualified intervention in the operation of the system. Strong systems, based on the setting of microclimate parameters and the introduced restrictions, themselves determine the necessary modes of operation technological systems and ensure their adjustment by automatically controlled processes of equipment operation and the state of plants in greenhouses, creating objective prerequisites for a reliable increase in yield and quality of the crop.

6. Methods of automatic control

Greenhouses with many parameters, cross-influences, distributed over large areas, wearing out equipment, susceptibility to unpredictable, and sometimes unmeasurable, influences belong to the class of complex, multidimensional, distributed, nonlinear and non-stationary control objects. Management methods should be appropriate: optimal - ensuring the optimization of the quality criterion, multi-connected - taking into account the multidimensionality of the object, dynamic - taking into account the length of control processes in time, adaptive and self-learning - taking into account the nonlinearity and variability of the parameters of the control object, with feedback on control and disturbances. The best results in terms of the quality of control processes are provided by systems built based on dynamic modelling of control processes in greenhouses. Moreover, the more parameters the used model takes into account, the more accurately the processes in greenhouses are calculated and the more precisely the control commands are determined. This applies to both processes the formation of a microclimate in greenhouses, which creates conditions for plant growth, and to the processes of plant growth itself.

The degree of automation and the methods used in control systems impose certain requirements on the technical support and of course, significantly affect the cost of control systems. Therefore, if it is economically expedient to impose maximum requirements on control systems for industrial greenhouse plants, then control systems for medium (farm) and small (individual) greenhouses, the requirements should be significantly limited.

7. The technical basis of the control system.

The technical basis or technical support of control systems includes two main components: means for monitoring and entering data, converting, processing, and issuing control actions (intellectual component) and means for electric power supply, switching, protection, and connection of controls (power component). There are several basic characteristics of technical support with many specific features.

8. Reliability of the element base.

As a rule: the lower the reliability of the elements, the lower the capital costs for creating the system and the higher the cost of operating costs for energy and equipment repair. Conversely, reliable equipment allows to pay less attention to its performance and more - to growing crops;

9. Availability of stock in terms of functional and quantitative indicators;

The technical base is formed, first of all, for reasons of ensuring the functionality of the control system. Secondly - from the condition of optimizing the price/quality ratio.

10.Convenience of operation and maintenance.

Control systems, like everything else, are not eternal. They have to be repaired over time, and it is also advisable to perform preventive maintenance to prevent malfunctions. Ease of operation and maintenance is determined by the presence and completeness of technical and technological documentation, including electrical schematic diagrams, equipment certificates, descriptions and operating instructions. These conditions and their relevance depend, in turn, on other factors: the degree of automation, structure, reliability of technical support, availability of design and operational documentation.

№	Title	Description	User/Role	Complexity	Priority
1.	Ability to turn on the heater.	User can turn on the heater to increase the temperature in the greenhouse.	Agronomist	Medium	High
2.	Ability to turn on the cooler.	User can turn on the fan to reduce the temperature in the greenhouse.	Agronomist	Medium	High
3.	Ability to turn on the light.	User can turn on the light to switch to the “Day mode”	Agronomist	Medium	High
4.	Ability to turn off the light.	User can turn off the light to switch to the “Night mode”.	Agronomist	Medium	High

5.	Ability to add water to the solution.	User can add water to the solution to decrease pH level.	Agronomist	Medium	High
6.	Ability to add fertilizers to the solution.	User can add fertilizers to the solution to increase pH level.	Agronomist	Medium	High
7.	Ability to plan the yield.	User can monitor current and scheduled plantings.	Agronomist	Medium	Medium
8.	Ability to receive orders from customers	User can view and operates with orders	Agronomist	Medium	Medium

Table 2 User requirments (Source: author)

This table summarizes the discussion above. It contains the minimum user requirements, based on which the system of this project will be developed, user specification, who will be able to use these functions, the complexity of implementation of these functions (in theory, all the listed requirements are approximately the same in terms of the level of complexity of implementation and are not as complex as possible) and priority of the described requirements, which mainly belongs to the requirements for the climate control system.

4.1.2 Information system design creation

The objective of this project is to provide a design of a smart system for a farm information system. It is also needed to take into account the fact that the hydroponic method is used for growing on this farm. Plants in such a farm are grown without soil in a special nutrient solution. This method was chosen because growing in hydroponics requires fewer resources, which leads to lower costs. Compliance with the growing regime is necessary for the normal growth and maturation of the crop. Managing various parameters of a greenhouse economy is a rather laborious task for a person because they need to be monitored around the clock. Therefore, the control of the mode of the greenhouse installation is carried out using automatic devices to comply with the mode of growing specific plants. The cultivation regime is influenced by various external indicators, which must be maintained in a given range. This can be temperature, humidity, lighting, soil acidity indicators, and other factors that are not considered in this case. Sensors from which information enters the system are used to measure these indicators. Sensors are the "eyes" and "ears" of the system; information will have to be entered with the help of a human operator without them, then there can be no question of any automation. Imagine an operator who walks around a greenhouse every ten minutes and records thermometer readings in a logbook. The presence of executive devices such as heaters, illuminators, fans, fertilizer controllers is mandatory for this kind of system. These devices are the "hands" of the system, with the help of which external conditions, such as temperature or soil acidity, are changed. The change in conditions is carried out based on a plant growing plan, which stores information about the points in time and the necessary actions at these moments. It is necessary for a certain plant maintaining a temperature of 25 ° C on the 15th day of growth, of which 14 hours with lighting, and then lowering the temperature to 18 ° C the rest of the day. It is also necessary to display the current state of the system with the possibility of operator influence and to record actions in the log to control the ongoing processes. Now that we have a basic understanding of what the system should do, we can use the Use Case diagram to define the objects in the system and the actions that these objects should perform.

4.1.2.1 Use case diagrams

System use case diagrams are intended to create a list of the operations that the system should perform. These diagrams are often called function diagrams because based on a set of such diagrams, a list of functions performed by the system is compiled, scenarios of the behavior of objects interacting with the system are described. Each such diagram is a description of a scenario of behavior followed by the actors when implementing a certain function of the system or when solving a certain problem.

The Actor icon is often used to represent the users of the system to identify the tasks performed by the users and their interactions in the Use Case diagram. Usually, the Actor icon denotes an object that:

- Interacts with the system or uses the system;
- Sends or receives information to the system;
- Is external to the system.

<u>The totality of the actors allows to find out</u>:

- Who uses the system;
- Who is responsible for maintaining the system or for the external hardware that is used by the system;
- what other systems should interact with this system.

<u>Decomposition of greenhouse control system</u>

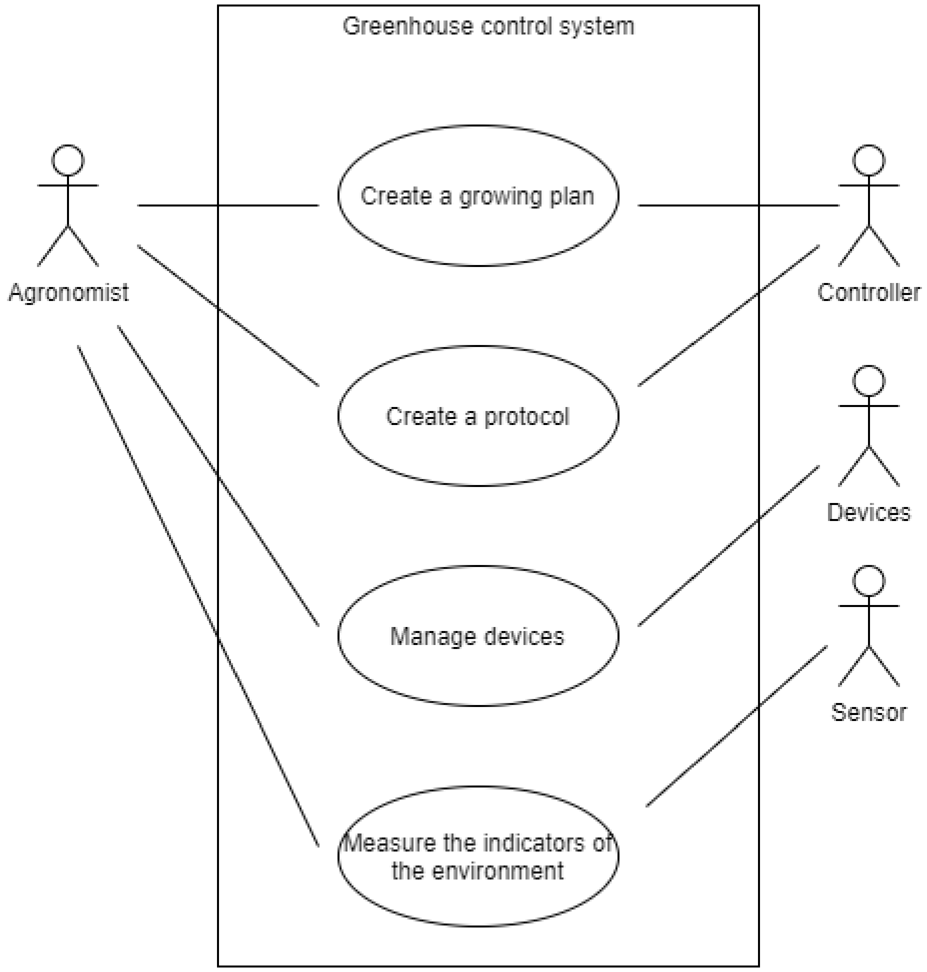

Figure 1 Use case diagram of greenhouse control system (Source: author)

There is some plan for growing a plant according to the problem statement. It must be entered into the system by the operator. The "agronomist" and the "create a growing plan" use case exist to represent this process.

Note that the growth plan must go into the system and be processed. Also, the operator should be able to view the log of the system. The "Controller" actor, "Create growing plan" and "Create protocol" use cases represent this process.

The controller must control the executive devices following the task set. A use case named "Manage devices" and the actor "Devices" exist to represent this process. It is also necessary to create a new actor "Sensor" and a use case "Measure the indicators of the environment".

The information determined with the help of the diagram under consideration is used in the development of the initial requirements for the designed system and technical specifications for its development.

The final version of the diagram should define the following initial requirements for the greenhouse management system:

1) the operator should be able to set a growing plan;

2) the operator must be able to view the log of the system;

3) the system should receive information from sensors;

4) the system should be able to control external devices based on the indicators of the sensors and the entered growing plan.

Decomposition of greenhouse control system

The next use case diagram represents the decomposition of controller device management. All the shown devices are connected to the controller and help it to test and adjust the environment of a smart farm.

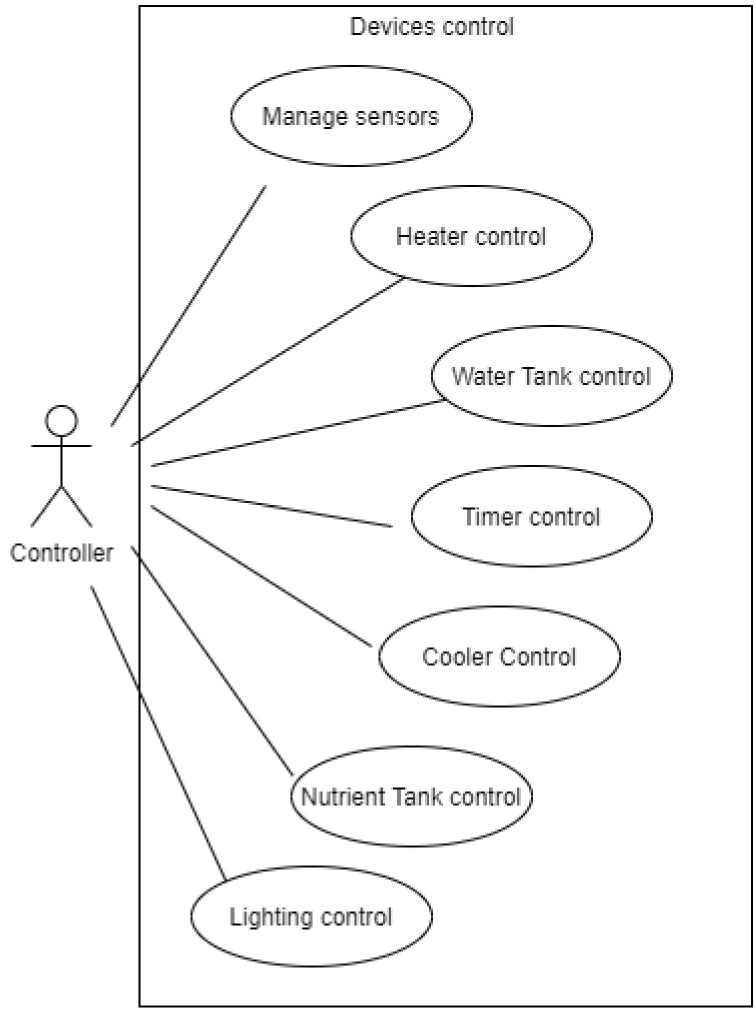

Figure 2 Use case diagram of controller device management (Source: author)

4.1.2.2 State diagrams

A state diagram is a model of the behavior of objects that helps to fully describe all the states of the system and transitions to these states during operation, as well as some algorithms for interaction between objects. The behavior model helps to look at the resulting software object from the outside because the main purpose of object-oriented programming is to create objects endowed with certain behavior that will adequately represent the processes of the domain. This type of diagram allows representing all the behavior of the future software object in the form of graphical state icons.

State diagram for class „Controller“

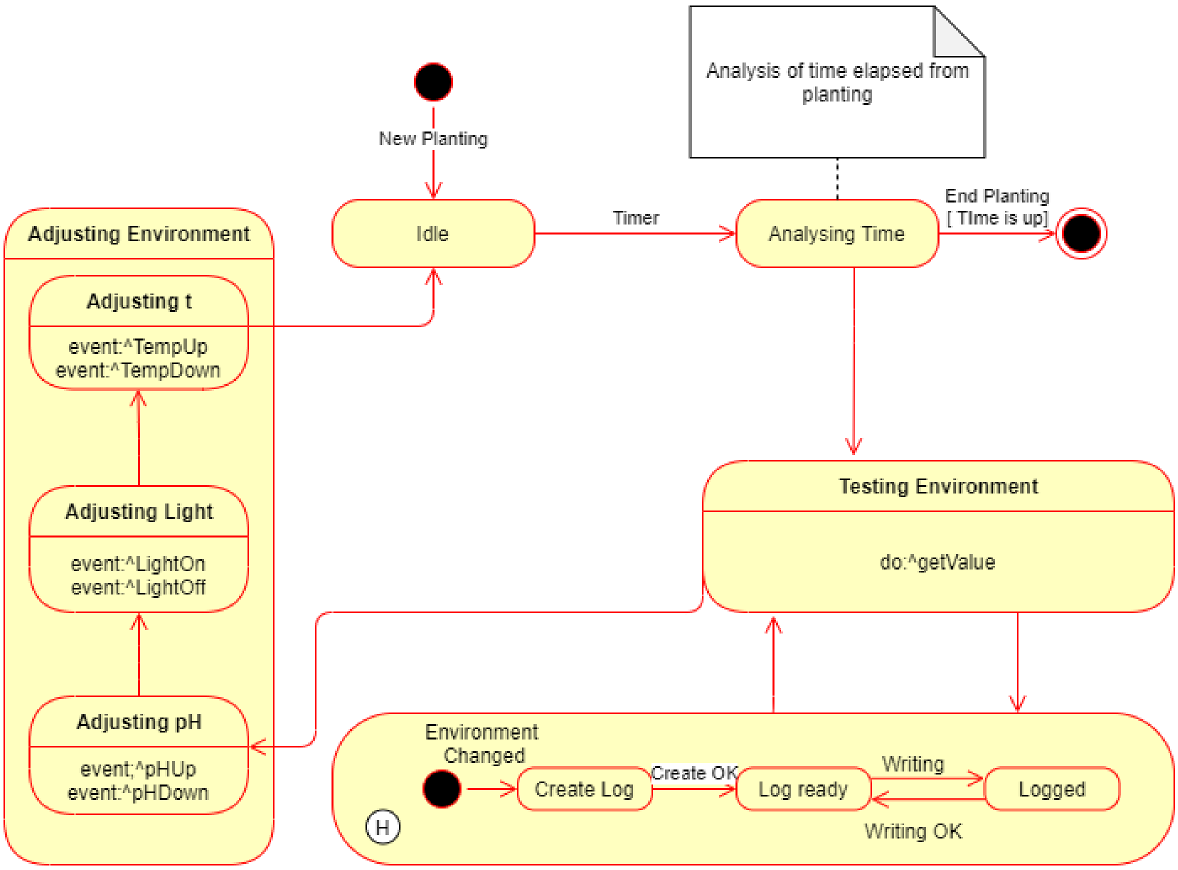

Figure 3 State diagram for greenhouse management system environment controller (Source: author)

There are two sensors planned in the system: a temperature sensor and an acidity sensor, which will be further referred to as a pH sensor, and t sensor. Devices are represented by a lamp, heater, fan, water tank and fertilizer tank. The lamp should turn on when there is a lack of lighting, however, the light sensor is not provided, which means that the lamp should always be on when the system enters the "day" mode and off when the system enters the "night" mode. The heater and fan must be at the correct temperature for growing. The heater should turn on when the temperature drops, and when the temperature rises, the fan should turn on. The water tank and fertilizer tank are planned to be used to change the acidity level. Water must come from the water reservoir when acidity rises, and the acidity level will decrease. Fertilizers come from the tank when the acidity decreases and the acidity level rises.

The task of the controller is to maintain the specified values of the parameters of the greenhouse environment: temperature, lighting, pH concentration. It is necessary to read the current state of the environment using sensors to maintain these parameters according to the plan. In this case, it will not count on the greater intelligence of the sensors. The sensors will not have their processor and will only issue changed parameters at the request of the controller. This process will be displayed using the "Testing Environment" state connected by the “Idle” state with the “Timer” event. It means that the controller has a built-in clock that initializes this event after a specified period of time. Now let's take a closer look at what needs to be done to test the environment settings. The controller polls the temperature and pH sensors when activated. Actions “do:^getValue“ represents the following procedure.

It is necessary to bring the indicators back to normal by turning on the appropriate devices after the state of the environment has been tested. “Adjusting Environment” state entered to represent this process.

The system waits for the next point in time after setting up the environment. This process is indicated by an arrow that connects the Adjusting Environment and Idle states.

It is logical to assume that the system should finish work at the moment when the crop is ripe. The key to maturation is the expiration of the growing time of the plant according to the growing plan. “Analysing Time” represents this process.

The system should go through the following states for the environment following the growing plan with the following assumptions:

- it is necessary to turn on the heater to increase the temperature in the greenhouse;
- it is necessary to turn on the fan to reduce the temperature;
- it is necessary to turn on the lighting to switch to the "Day" mode;
- it is necessary to turn off the lighting to switch to the "Night" mode;
- it is necessary to add water to the solution to decrease the pH;
- it is necessary to add fertilizers to the solution to increase the pH.

Moreover, it is possible to simultaneously turn on the heater, bulbs and a water shutter, but it is impossible to turn on the heater and the fan at the same time, as well as the simultaneous flow of water and fertilizers.

After analyzing the necessary actions, it is not difficult to notice that the listed set can be divided into three independent parts, these are:

• adjusting temperature;

• adjusting light;

• adjusting pH.

Since it is assumed that there is only one processor, these states will be traversed sequentially. But if a different processor is used for each part, then it would be possible to create stand-alone classes to manage these states.

The temperature is only adjusted when it changes, as shown by the tempDown and tempUp functions. If one of these events occurs, then the temperature is compared with the one that should be according to the plan, and if it is less, then the heater is turned on, and if it is more, the fan is turned on. However, if the temperature has reached a normal level, then the fan and heater are turned off. The same happens with the pH level.

The light just turns on when the lighting time comes and turns off when the night time comes for lighting.

When it first enters the message logging mode, it is needed to create a log file, which it does not need to recreate with subsequent calls to this mode. The adjusted setting is indicated by the letter H in a circle. Logging starts when conditions change (Environment Changed). The Log file (Create Log) is created after that and the system goes into the waiting state (Log ready). If it is necessary to write the changes, they are recorded (Logged), and the system enters the idle state again. In this case, it is needed to start with the Log ready state, and not with the creation of the protocol the next time.

State diagram for class „Order“

The following state diagram is showing the behavior of an order in an order processing system. The diagram shows the different states in which an order can be.

The figure shows three conditions:

- if not all items included in the order are checked, then get the next item and return to the Checking order items state;
- if all items have been checked and all of them are in stock, then go to the Issue of a purchase order state;
- if all items are checked, but not all of them are in stock, then go to the Waiting state.

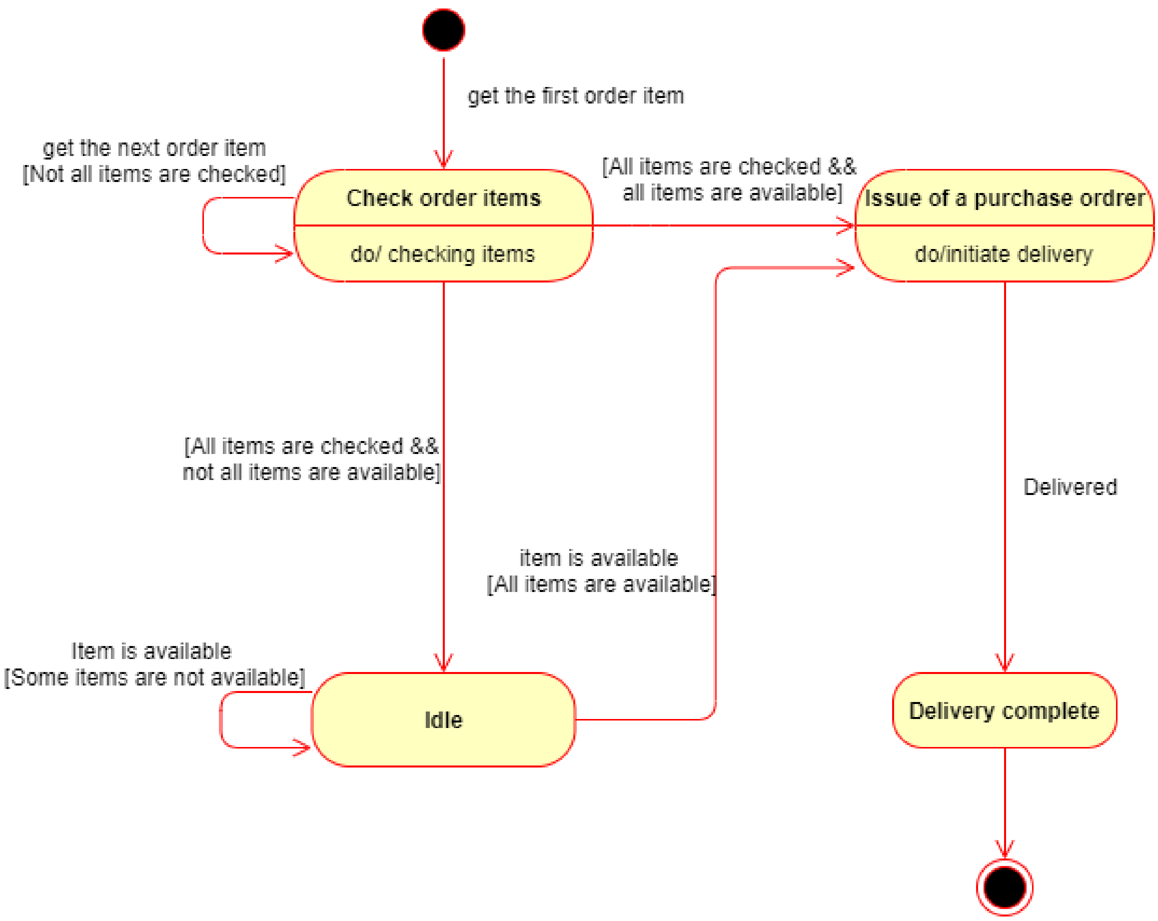

Figure 4 State diagram for „Order“ class (Source:author)

4.1.2.3 Activity diagram

An activity diagram allows modelling a sequence of business processes or actions implemented by class methods. These sequences can represent alternative branches of the data processing or branches that can be executed in parallel. It is advisable to use activity diagrams to analyze: the content of scenarios for the application of the designed system, the interaction of workflows of different scenarios, the execution of scenarios in multiprocessor computing environments. For example, when analyzing the activities of a financial company, this type of diagram is advisable to use to model flows of financial documents reflecting the passage of payment of invoices or orders.

The main elements of an activity diagram are activity states. Activity states are represented by rounded rectangles. Action states differ from activity states in that they cannot be decomposed. Action states are connected by lines with arrows that define the directions of transitions from one action state to another. It is believed that these transitions are carried out when some actions are completed and others are started without time-consuming. In addition, the diagram also shows thickened horizontal segments of straight lines; These line segments determine the division of the data processing process into parallel branches or the assembly (synchronization) of parallel branches of the data processing. Diamond images are used to represent the division of a data processing process into alternative branches or assemblies of alternative branches of a specified process. The main difference between activity diagrams and state diagrams is that in the first case, the main elements are actions, and in the second case, the static states of objects. At the same time, activity diagrams are more suitable for modeling a sequence of actions, and state diagrams for modeling discrete states of an object.

Activity diagram for the environmental controller

This diagram describes the process of interaction between objects „timer“, „controller“ and „growing plan“ during which the controller periodically tests the greenhouse environments at regular intervals and adjusts it. The process contains setting of the timer, getting the current time from the timer, testing the environment, adjusting the environment.

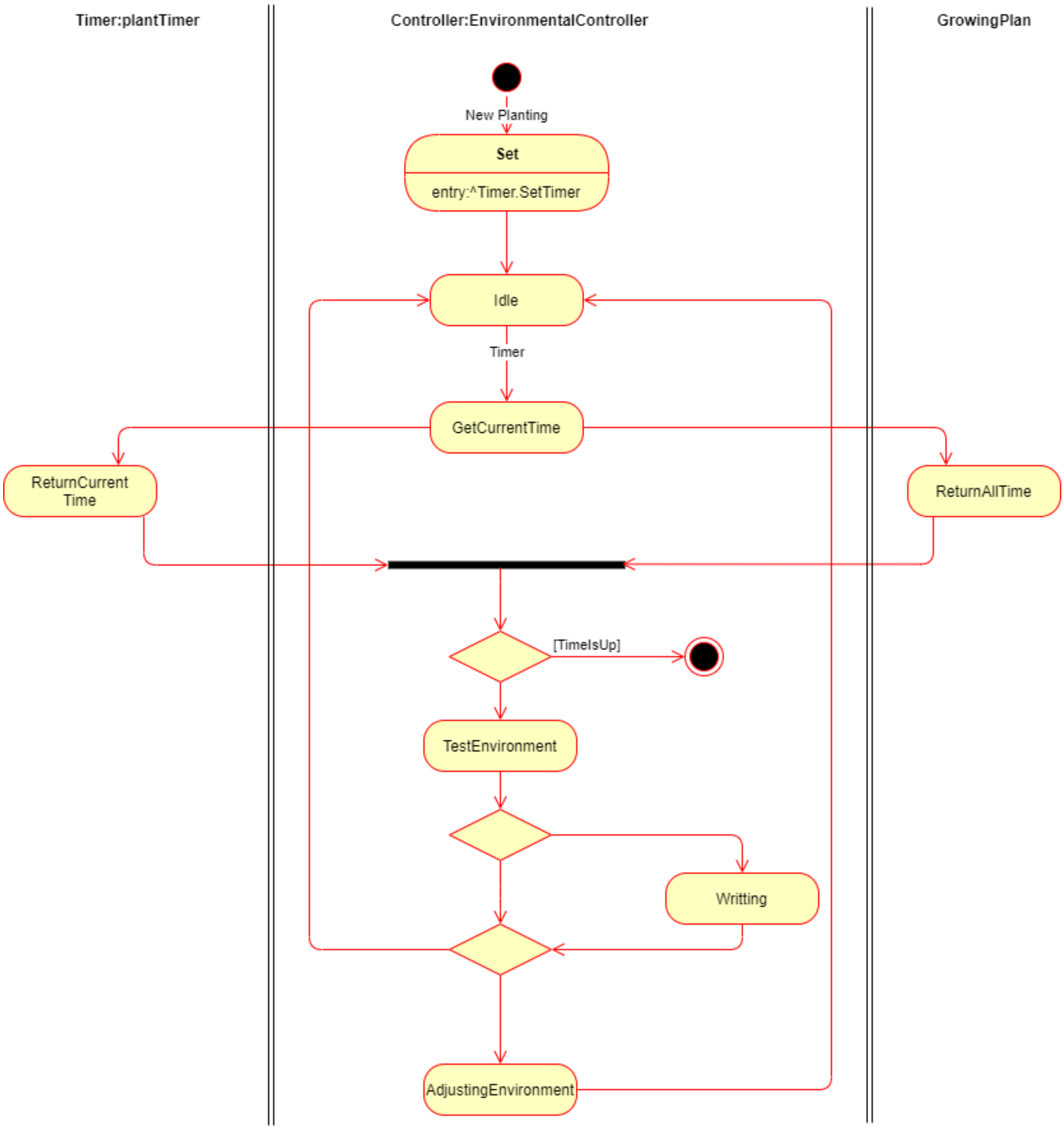

Figure 5 Activity diagram for environmental controller (Source:author)

4.1.2.4 Sequence diagram

The diagram of the interaction between objects will continue to create a hydroponic system design. In addition to the scenario of the behavior of each object of the system, it is necessary to accurately represent the interaction of these objects with each other, the definition of clients and servers and the order of the exchange of messages between them. Messages are exchanged in a specific sequence, and the Sequence Diagram allows a reflection of this exchange over time. Client objects send different messages to each other, and server objects process them during the operation of a complex system. In the simplest case, it can consider a message as a call to a method of some class; in more complex cases, the server has a message queue handler, and messages are processed by it asynchronously, that is, the server accumulates several messages in the queue if it cannot process them at once. The multitasking "Windows" is based on the reception and transmission of messages, and in this case, for simplicity of demonstration of creating an application, it will be assumed that messages are processed immediately in the order in which they are issued by users.

Sequence diagram for environmental controller

Usage scenario: controller - devices interaction

Description: Controller exchanges messages with sensors and devices.

Goal: Adjust Environment

Actors: Controller, Devices, Sensors

Condition: The scenario repeats after a certain period of time

Scenario: 1.Controller sets the timer.

2.Controller counts down the time from the start of planting.

3.Controller asks for information on the current temperature.

4.Controller asks for information on the current pH level.

5.1. IF the temperature is low, the controller turns on the heater.

5.2. IF the temperature is high, the controller turns on the cooler.

5.3. IF pH level is high, controller opens the water tank.

5.4. IF pH level is low, controller opens the nutrient tank.

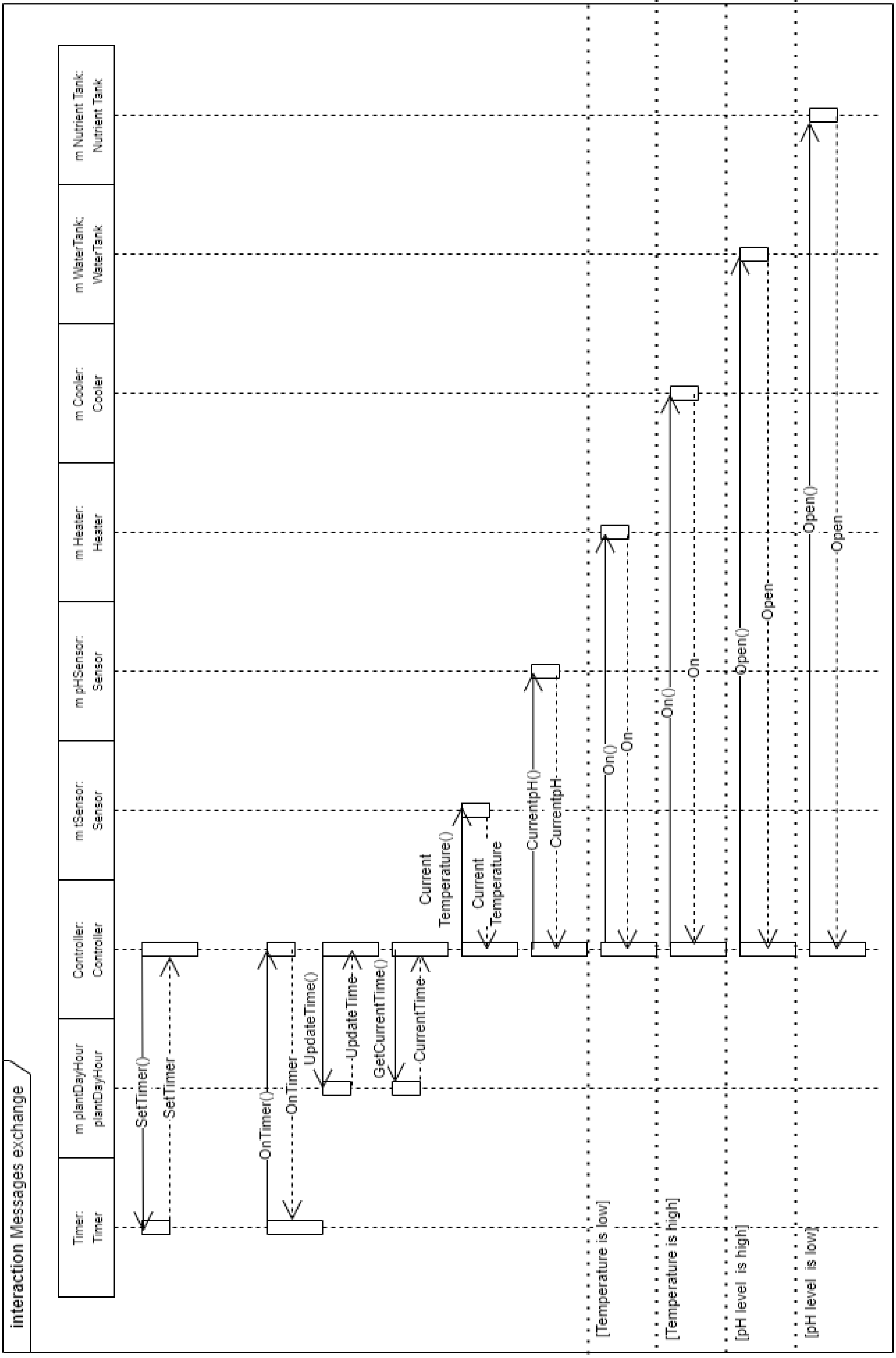

Figure 6 Sequence diagram for environmental controller (Source:author)

- the object “Timer” belongs to “PlantTimer” class;
- the object “plantDayHour” belongs to “plantDayHour” class;
- the object “Controller” belongs to “EnvironmentalController” class;
- the object “tSensor” belongs to “Sensor” class;
- the object “pHSensor” belongs to “Sensor” class;
- the object “Heater” belongs to “Heater” class;
- the object “Cooler” belongs to “Cooler” class.
- the object “WaterTank” belongs to “WaterTank” class.
- the object “NutrientTank” belongs to “NutrientTank” class.

This sequence diagram shows how the controller sends requests for information to sensors and disposes of turning on devices. The object "m_plantDayHour" of class "plantDayHour" simulates the countdown of time elapsed from the beginning of planting.

Sequence diagram for Order process.

Usage scenario: Order process

Create order

Goal: Create order

Actors: Customer

Condition: Customer registration completed

Scenario: 1. Login to the system
2. Fill order information
3. Save all information in database

Action: Click Save button

Modify order

Goal: Update order information

Actor: Agronomist

Conditions: Order info properly filled and saved in the database

Scenario: 1. Find order
2. Click Update button

3. Modify necessary fields

4. Update information in database

Action: Click Update button

Receive order

Goal: Complete order

Actor: Customer (Receiver)

Condition: Order delivered to the destination

Scenario: 1. Login into the system

2. Find order

3. Check delivery status

4. Receive order

Action: Order delivery notification

It shows the different events happening in the process of delivering order starting with the order creation by customer and ending with full order info in the following sequence diagram. It is illustrated as follows:

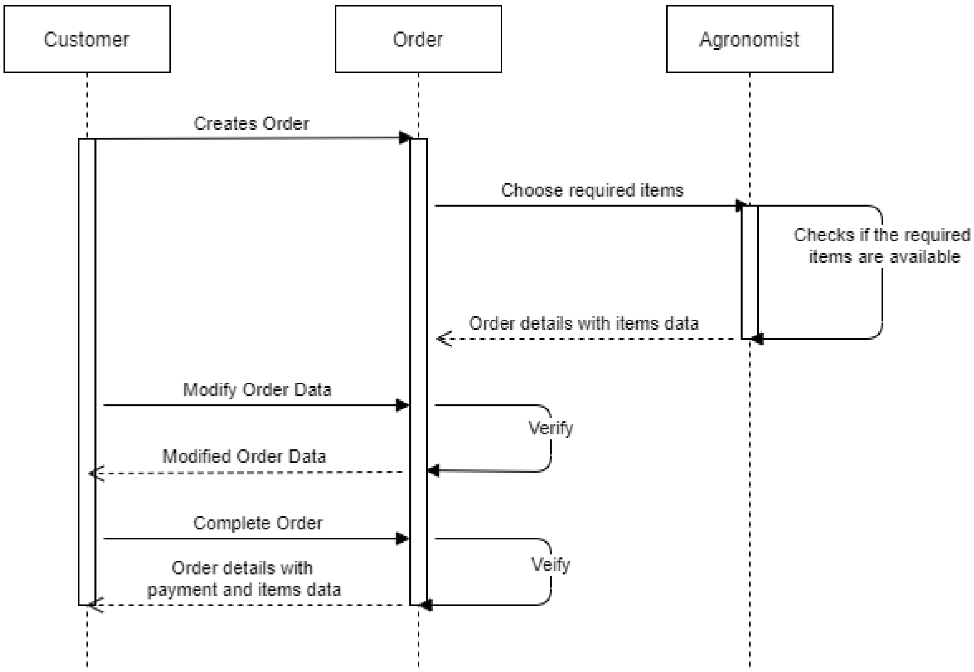

Figure 7 Sequence diagram for Order process (Source:author)

4.1.2.5 Class diagram

Class diagram is a structural diagram of the UML modeling language that demonstrates the general structure of the system's class hierarchy, their interactions, attributes (fields), methods, interfaces and the relationships between them. It is widely used not only for documentation and visualization but also for design by means of forward or reverse engineering [32].

The purpose of creating a class diagram is a graphical representation of the static structure of the declarative elements of the system (classes, types, etc.) It also contains some elements of behavior (for example, operations), but their dynamics should be reflected in diagrams of other types (communication diagrams, state diagrams). For ease of perception, the class diagram can also be supplemented with the representation of packages, including nested ones [33].

When presenting real-world entities, the developer needs to reflect on their current state, their behavior, and their mutual relationships. At each stage, an abstraction is carried out from unimportant details and concepts that do not relate to reality (performance, encapsulation, visibility, etc.). Classes can be viewed from the perspective of different levels. As a rule, they are divided into three main ones: analytical level, design level and implementation level [34]:

- at the level of analysis, the class contains only a sketch of the general outlines of the system and works as a logical concept of the domain or software product.
- at the design level, the classroom reflects the main design decisions for information distribution and planned functionality, integrating status and operations information.
- at the implementation level, the class is finalized to the form in which it is most convenient for implementation in the chosen development environment; at the same time, it is not forbidden to omit in it those general properties that do not apply in the selected programming language.

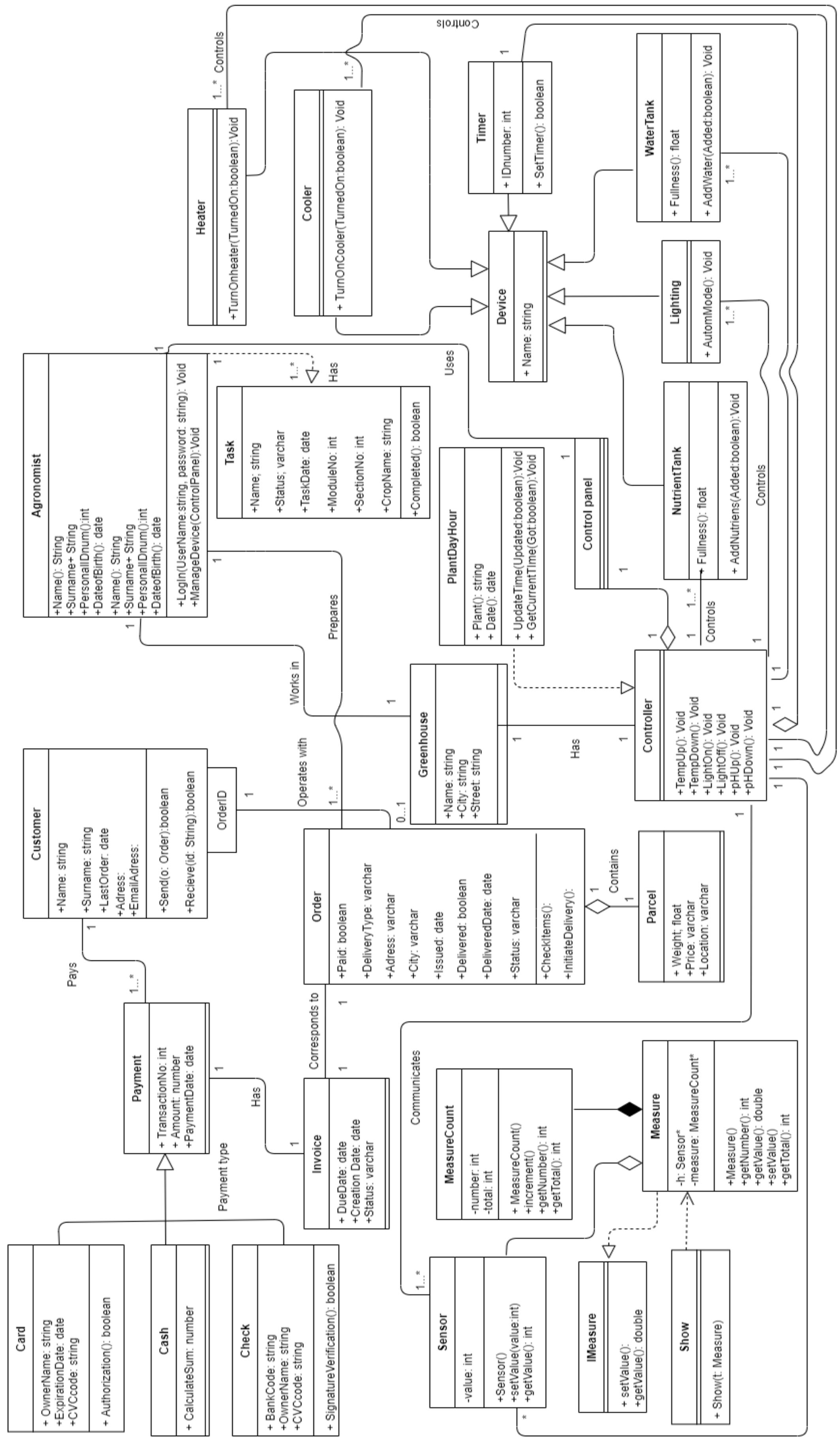

Figure 8 Class diagram of farm information system (Source: author)

Data dictionary

<u>Class Agronomist</u>

This class represents a user, for which the following system are designed. Agronomist is able to use devices to control the process of production in a greenhouse.

<u>Class Customer</u>

Class that contains all personal data: name, address, email etc. A customer needs to have his/her profile completed to be able to make the order. Profile is completed if all the data is properly filled out.

<u>Class Payment</u>

Class that is used for store information about payment details and contains methods for its proceeding.

<u>Class Card</u>

Subclass of class Payment that defines one of the payment types. Contains attributes such as owner's name, expiration date and CVC code. Contains method of authorization.

<u>Class Cash</u>

Subclass of class Payment that defines payment type. Contains a method of calculating the sum for the order.

<u>Class Check</u>

Subclass of class Payment that defines payment type. Contains attributes such as bank code and owner name. Contains method of signature verification.

<u>Class Invoice</u>

Each payment has an invoice that is being sent to the customer after payment. An invoice corresponds to an order.

<u>Class Order</u>

One of the main classes for the system that is used for company information about the order. It contains methods to operate with the order.

<u>Class Measure</u>

Measure class is the main class among classes, which explains how a sensor is counting the information. The class "measurement" is a temperature and pH meter. The arithmetic average of all measurements is formed as the measured value - the sum of all measurements divided by their number.

<u>Class Sensor</u>

The Sensor class (as a temperature and acidity sensor) is used to receive measurements and add them. The measurements themselves are passed to this class for summation in a console task. The class consists of an aggregation relationship with the main “class Measure”.

<u>Class Measure Count</u>

The number of measurements is formed by the MeasureCount class, which contains a static property "total" for counting the total measurements, as well as a count property for counting the number of meters for a specific "Measure" object. The MeasureCount class is in relation to "composition" with the "class Measure".

Class IMeasure

The IMeasure class is an interface to the Measure class and is kind of an implementation provider.

Class Show

The Show class is in a "dependency" relationship with the Measure class because the implementation of Show's class only Show method depends on the structure of the Measure class.

Class Parcel

This class contains all information about the product, which is being sent to the customer (weight, price, location). Class Parcel is in an aggregation relationship with class Order.

Class Greenhouse

This class is used to store basic information about the company.

Class Controller

Another of the most important classes of the system. The basic purpose of the controller is to create the right environment and supply crops with needed substances in the greenhouse.

Class Task

Although performs the basic functions of plant care, the agronomist still occupies a leading role in the greenhouse hierarchy. The "job" class exists to structure the responsibilities of the agronomist.

Class Control Panel

Agronomist interacts with the controller with the help of a control panel.

Class Plant Day Hour

This class was created to simulate the time elapsed from the start of planting.

Class Heater

A heater increases the temperature in the greenhouse. A heater is connected to the controller, from which it receives signals in case the temperature is needed to be increased.

Class Cooler

A cooler decreases the temperature in the greenhouse. A cooler is connected to the controller, from which it receives signals in case the temperature is needed to be increased.

Class Device

A device class is a class that outperforms all devices connected to an intelligent system. Each device connected to the farm system has its own name and properties. This class uses a generic relationship for each device.

Class Timer

The controller has a timer that initiates testing of the greenhouse environment after a specified time.

Class Nutrient Tank

Nutrient Tank increases the level of pH in the environment as directed by the controller.

Class Water Tank

Water Tank decreases the level of pH in the environment adding more water as directed by the controller.

Class Lighting

This is a mostly automated device that is regulated by the timer in the controller. It has two modes:

- Day mode
- Night mode

4.1.3 Information architecture design

The information architecture of the smart farm has been interpreted using the card sorting method. The three main categories of control are shown in this diagram: orders, growing plan and control panel. The rest of the categories represent more detailed user interaction with the process of growing food and interaction with customers. All provided categories were identified during the analysis of the requirements for the developed system and help to produce products with minimal intervention.

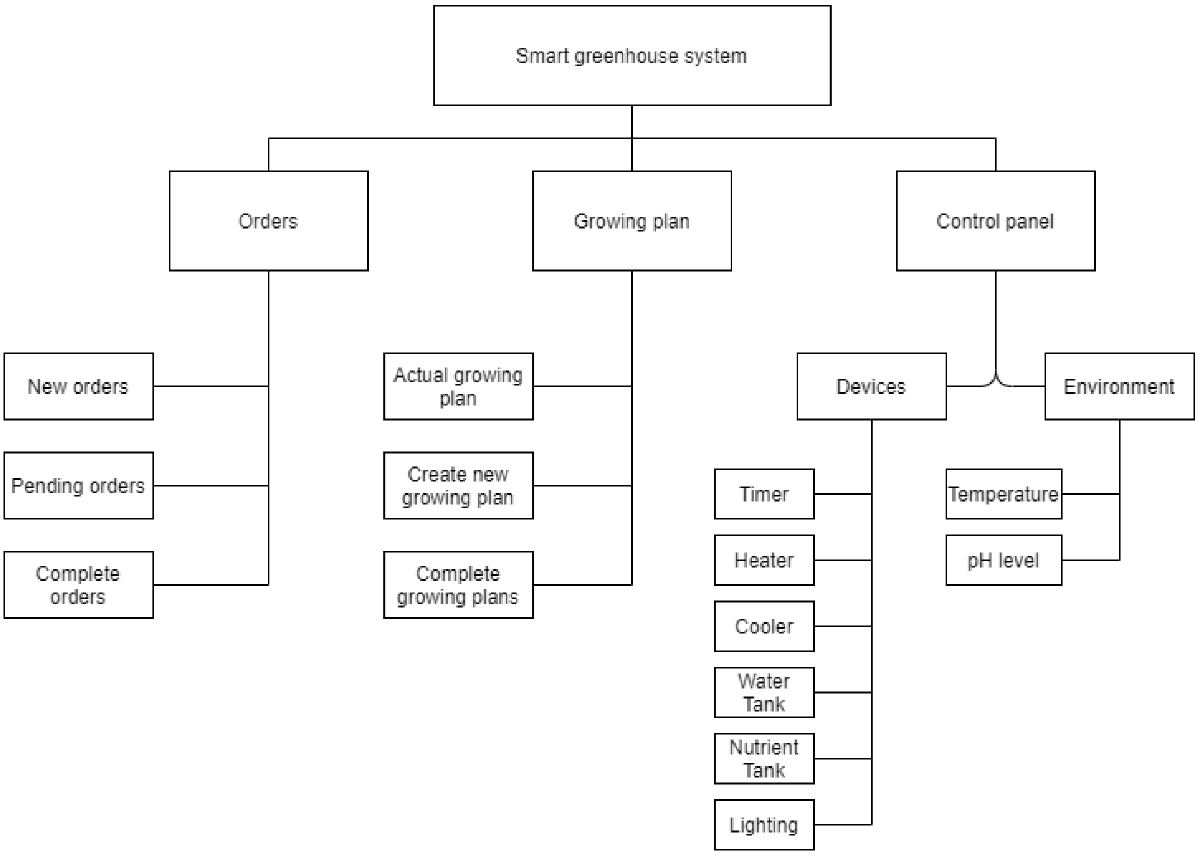

Figure 9 Information architecture design (Source:author)

Wireframe

A wireframe is a diagram or drawing that represents the "skeleton" of a website or application page. There are no decorations, just the location and approximate sizes of headers, text blocks, illustrations, multimedia and navigation bars.

Mockup: Orders and Growing plan.

As planned, the agronomist will control the greenhouse using a control panel that can be installed on a tablet as an application. The left picture below shows the orders category. The user can view new orders that come from customers. Also, the user can view the orders that are in the process of execution. Also, the user can view the history of orders and view orders that have been completed or canceled for specific reasons.

In the right picture, the Growing Plan category is open. The user can get acquainted with the current growing plan (instructions and terms for growing a certain crop, which is in the process of growing at this moment). Also, the user has the opportunity to create a new growing plan (for the future or in case the current growing plan has been completed). The user can view the completed growing plans.

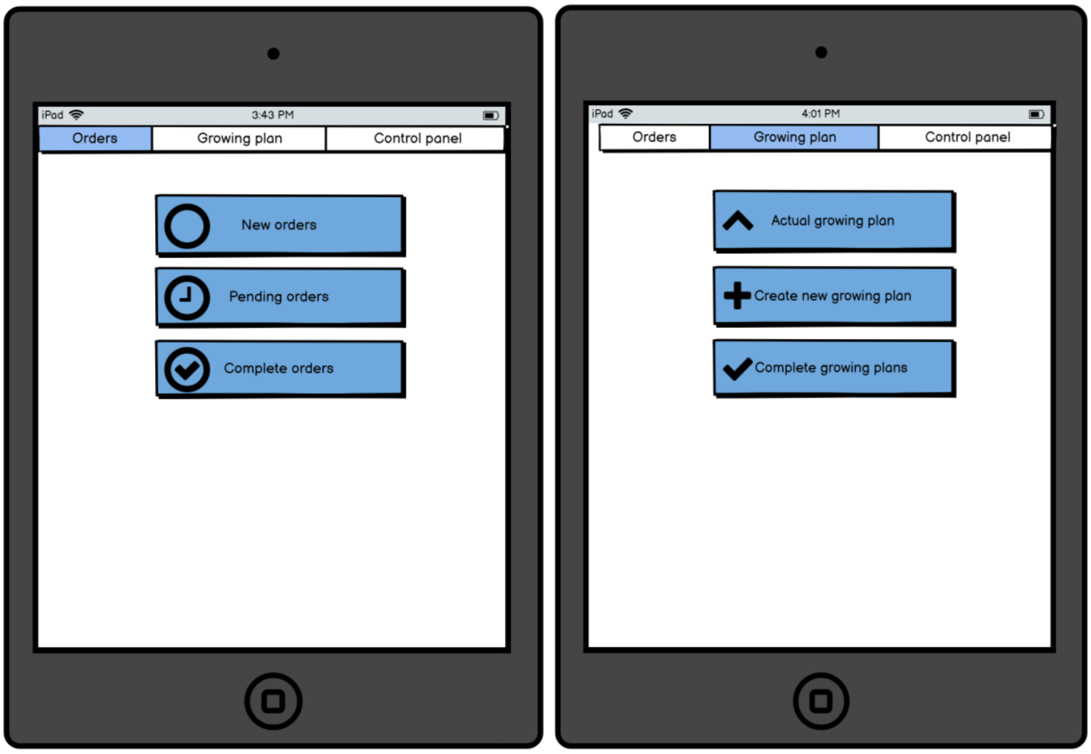

Figure 10 Mockup for Orders and Growing plan (Source:author)

Mockup: Control panel and Devices.

The left picture shows the layout that the user will see by clicking on the "control panel". The user can go to the list of devices, or get acquainted with the indicators of temperature sensors, acidity sensors, etc.

The right picture shows the layout that the user will see by selecting the "devices" category. Thus, he will be able to switch to manual control of any of these devices and adjust the greenhouse environment to specific requirements.

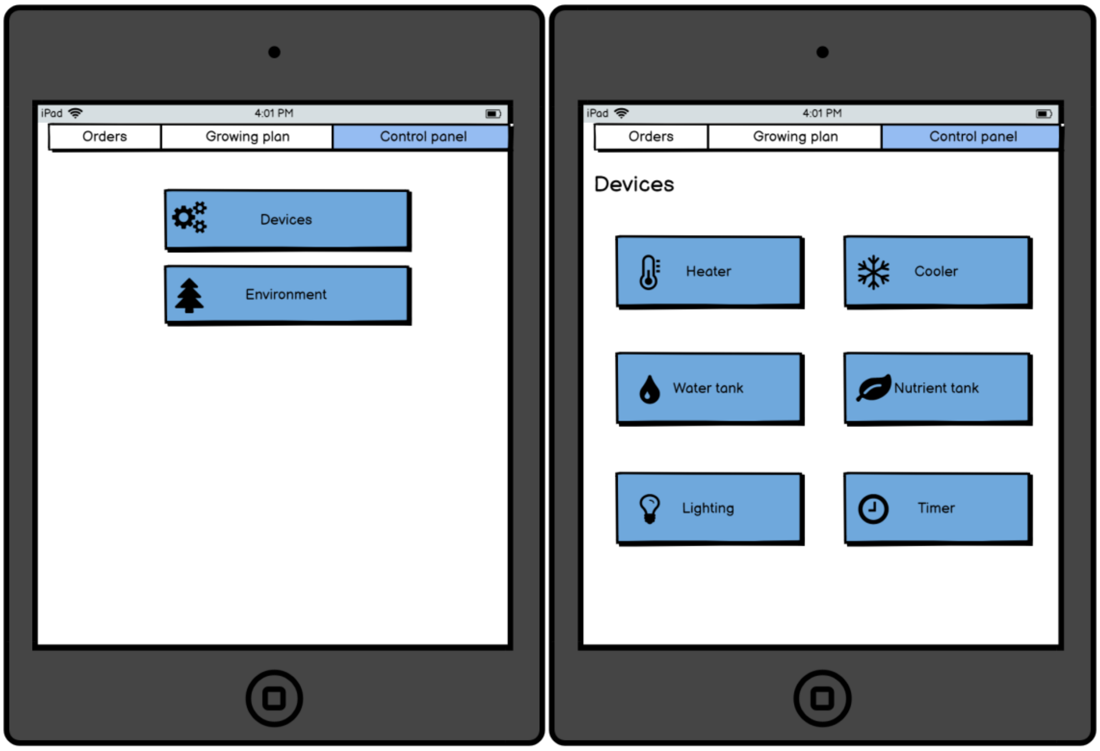

Figure 11 Mockup for Control panel and Devices (Source:author)

Mockup: Heater and New orders.

The left picture shows the heater control panel. The user can come to it by pressing the "heater" button in the list of devices. First of all, he will see the readings of the temperature sensor (temperature in the greenhouse). The user can go to manual control to set the temperature manually or turn off the heater by turning off the auto mode.

The right picture shows the interface for viewing new orders. Each order has its number, date, customer name, details, etc. The user can accept the order, change the order or contact the customer directly.

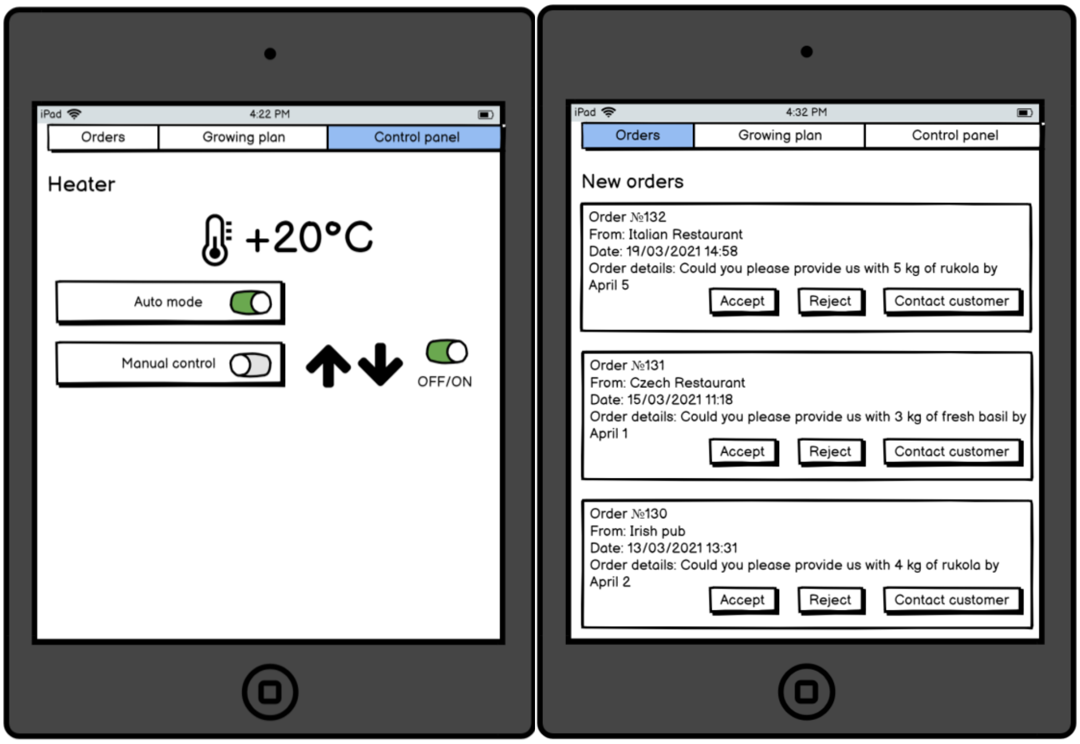

Figure 12 Mockup for Heater and New orders (Source: author)

5. Results and Discussion

5.1.1 The scope of the smart farm system model

The smart farm system proposed in this thesis is a generalized model of the greenhouse system based on a thorough analysis of requirements. It was not possible to describe the entire system completely due to the limitations within the framework of this work. Only individual parts have been selected and described by diagrams of the UML language. Diagrams have been simplified, and some freedom of action has been left in the implementation. All these actions were carried out following the recommendations for the UML specification in the UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design [35], which states that models can be scaled down:

1. diagrams may not contain the entire background (for example: for simplicity);
2. incomplete - some elements may be completely absent from the model;
3. inconsistent - the model may contain contradictions.

These points are based on the principle that diagrams are used by developers to display certain an object that develops and changes over time. When it changes one part of the system usually needs to change all the diagrams, which in the case of large systems can be problematic. Therefore, the UML standard leaves some leeway in the presentation, completeness and consistency of individual diagrams.

5.1.2 Smart farm system implementation design

The implementation of the smart farm system itself is not the goal of this thesis. Nevertheless, the author of this work advises to take into account the following recommendations:

1. Implementation should be based on the proposed model.
2. Implementation must comply with the requirements received.
3. The author recommends using a uniprocessor system to save money and simplify the overall system. The Russian population is noticeably different in terms of solvency from the citizens of Western states. That is why low prices are especially tempting for Russian consumers. The average citizen of the Russian Federation will make his choice in favor of a low price, not a high quality.

4. Since this system is intended for deployment in Russia, it should be borne in mind that the Russian infrastructure is far from ideal. Frequent power outages are caused by inadequate maintenance and the use of outdated equipment. The author recommends that an alternative source of electricity be stored. Loss of power can lead to huge losses, because the entire system is completely dependent on electricity (controller, sensors, heater, cooler, etc.). That is why it is worth having a generator running on fuel in the greenhouse, which can provide production with power for a while in case of emergency.
5. The author recommends placing a greenhouse in an area with a good Internet connection. The system is almost completely autonomous, but the fact is that the agronomist accepts orders via a kind of email, and interruptions in the Internet connection can lead to a disruption of customer service.
6. It is recommended to use a room with a glass ceiling for econimy. Using sunlight will dramatically reduce electricity bills for lighting.
7. This system is designed for collaboration with a hydroponic growing method. This method of cultivation does not require soil, which significantly reduces the total amount of costs.
8. It is recommended to locate an enterprise with this control system in the northern regions of the Russian Federation. Automated climate control allows cultivation all year round in all weather conditions. The absence of the need to bring fresh products from remote regions will significantly reduce the cost of products, which will have a good effect on the well-being of the population.

6. Conclusion

The purpose of the thesis was to integrate technology into Russian agricultural enterprise to optimize production processes and reduce the workload on staff. The choice fell on the design of a cheap and full-featured microclimate control system that would provide complete and autonomous control over the main microclimate parameters: temperature, the concentration of nutrients, lighting inside the greenhouse.

The basic concepts of information systems and technologies and their varieties were analyzed in the theoretical part. Information technologies in the field of agriculture were also considered. Further, in this work, it was narrated about the integration of information systems in the agriculture of the Russian Federation, their methods and technologies.

The practical part of the thesis was devoted to the design of an information system. First, the potential requirements for the system being developed were formulated and analyzed. Then the information system was presented in a set of UML diagrams, which showed in detail the principle of operation of several processes. The user interface was presented in the form of diagram which was created with help of card sorting method and converted to mockups.

In the final part of the thesis, the designed system was subject to discussion, and the author also gave several practical recommendations for working with the developed system.

The result of the presented thesis is a set of UML diagrams that can serve as a starting model for designing a smart farm system. The resulting model can be expanded according to the requirements of individual customer systems, but the following project’s main part covers all important aspects of greenhouse system.

7. References

1. 1C: Enterprise 8. Accounts department of an agricultural enterprise. Access mode: http://solutions.1c.ru/catalog/agr-buh

2. Baydybakova S.K. Implementation of an automated accounting system at enterprises of the agro-industrial complex / Baidybakova S.K.// Privolzhsky Scientific Bulletin. - 2013. - No. 3 (19) - p.58-62

3. Besedina K.V. Modern approaches to the integration of corporate modern approaches to the integration of corporate applications / K.V. Besedina // Academy. - 2017. - No. 5. - P.41-43.

4. Bozhinsky I.A. Methods and technologies for the integration of information systems and distributed databases / I.A. Bozhinsky // Information Technologies. - 2015. - No. 2. - P.30-36

5. Web services as a means of integrating applications into the WWW. - Access mode: https://studopedia.net/2_24530_veb-servisi-kak-sredstvo-integratsii-prilozheniy-v-WWW.html

6. Vichugova A.A. Methods and tools for integrating information systems within a single information design space. Vichugova, V.N. Vichugov, E.A. Dmitrieva // Bulletin of Science of Siberia. - 2012. - No. 5 (6)

7. Dumchenkov I.A. Review of methods for integrating information systems, their advantages and disadvantages / I.A. Dumchenkov // Young Scientist. - 2018. - No. 23 (209). - p.176-177

8. Ermakova N.Yu. Information support of the farm: state, problems, development prospects / N.Yu. Ermakova, I. V. Ermakov A.N. Ermakova. // Development of the agro-industrial complex: theory and practice. - 2009. - No. 41 (134). p.57-63.

9. Zinchuk T.A. Modern directions of informatization of agriculture in Ukraine. / T.A. Zinchuk // Nikon Readings. - 2012. - No. 17. - p.368-369.

10. Integration of enterprise information systems. - Access mode: http://www.intuit.ru/studies/courses/4115/1230/lecture/24065

11. Kogalovsky M.R. Methods of data integration in information systems / M.R. Kogalovsky // Access mode: http://www.ipr-ras.ru//articles/kogalov10-05.pdf

12. Korotkov A.P. Integrated information systems in business // A.P. Korotkov - Moscow - 2006 .—640p.

13. Krasikov V.E. Review of the main characteristics and problems in the integration of information systems / V.E. Krasikov // Problems of Science and Education. - 2020. - No. 15 (99). - P.9-30.

14. Kusov A.A. Integration problems of corporate information systems / A.A. Kusov // Management of economic systems: electronic scientific journal, 2011. - № 4 (28). - state number reg. Article 0421100034/0124. - Mode of access to the journal: http://uecs.mcnip.ru. (date of access: 16.09.2020).

15. Makhotlova M. Sh. Information technologies in agriculture / M. Sh. Makhotlova // Young scientist. - 2015. - No. 10 (90). - p. 108-111. - Access mode: https://moluch.ru/archive/90/18664/ (date of access: 09/13/2020).

16. Mikhailov I.S. Research and development of methods and software for ensuring the structural and semantic interoperability of information systems based on metamodels: dissertation for the degree of candidate of technical sciences. - Moscow, 2008 .—235p.

17. Morozova O.A. Integration of corporate information systems: a tutorial // O.A. Morozov - Moscow: Financial University, 2014 .-- 140 p.

18. Review of digital technologies for the agro-industrial complex from GIS to the Internet of things. - Access mode: http://integral-russia.ru/2020/07/30/tsifrovaya-platforma-razvitiya-agropromyshlennogo-kompleksa-kontseptsiya-i-osnovnye-tezisy

19. Savchenko O.F. Methodological aspects of creating information systems in agriculture / O.F. Savchenko. // Achievements of science and technology in the agro-industrial complex. - 2006. - No. 11. - .WITH. 5-9

20. Semerkhanov I.A. Methods and algorithms for the automated integration of information resources based on the ontological approach. St. Petersburg, 2014 .-- 27p.

21. Information support systems for analysis and forecasting in agriculture. Scientific works of the A.A. Nikonov, Issue 50 / M .: VIAPI named after A.A. Nikonov. - 2019. – 120p.

22. Turdyshov D.Kh. Farm management information system. / D.Kh. Turdyshov. // Modern problems of science and education. - 2012. - No. 6. URL: http://science-education.ru/ru/article/view?id=7616 (date of access: 16.09.2020).

23. Fedorenko V. F. Information technologies in agricultural production / V. F. Fedorenko. - pos. Pravdinsky: Rosinformagrotech, 2014 .-- 224 p. - ISBN 978-5-7367-1021-8. - Text: electronic // Lan: electronic library system. - URL: https://e.lanbook.com/book/104340 (date accessed: 09/13/2020).

24. Shibanov S.V. Yarovaya M.V., Shashkov B.D., Kochegarov I.I., Trusov V.A., Grishko A.K. Review of modern methods of data integration in information systems / International Symposium "Reliability and Quality". - Penza, 2010 .-- p. 292-295.

25. Kari Spencer. City Farming: A How-to Guide to Growing Crops and Raising Livestock in Urban Spaces. Sheffield 2017.

26. Annette Cottrell and Joshua McNichols. The Urban Farm Handbook: City-Slicker Resources for Growing, Raising, Sourcing, Trading and preparing What You Eat. Seattle 2011.

27. Novella Carpenter. Farm City: The Education of an Urban Farmer. New York 2009.

28. Food and agriculture data - Access mode: http://www.fao.org/faostat/en/#home.

29. Ivanov programming: –M.: Publishing house of MSTU im. , 2002 .-- 320 p.

30. "UML. User's Guide". - M.: DMK, 2000.

31. Rambeau, J., UML: A Special Reference. - SPB .: Peter, 2002.-656 p.

32. Booch, Rambeau, Jacobson, 2006, Class Diagram, p. 120.

33 Booch, Jacobson, Rambeau 2006, class diagram, p. 226.

34 Booch, Jacobson, Rambeau, 2006, Classes, p. 68.

35. J. Arlow, I. Neustadt. UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design, 2nd Edition. 2005 by Addison-Wesley Professional

36.Ananyev M.A. Application of information technology in the agricultural sector / M.A. Ananyev, Yu.V. Ukhta. [Electronic resource] - URL: www.sisupr.mrsu.ru.

37.Matveev D.M. Technical and technological re-equipment of agriculture is necessary / A.T. Stadnik, D.M. Matveev, M.G. Krokhta, P.P. Kholodov // AIC: Economics, Management. - 2012. - No. 5.

38.Matveev D. M. The role of consulting activity in the technical and technological re-equipment of agriculture / A. T. Stadnik, D. M. Matveev, M. G. Krokhta, P. P. Kholodov; Novosib. state agrarian. un-t - Novosibirsk: Publishing House of the NSAU, 2013 .

39.Menyakin D.V. Information systems and their application in the agricultural sector / D.V. Menyaykin, A.O. Talanova // Young scientist. - 2014. - No. 3.

40. System Integration, Jeffrey O. Grady, CRC Press, Jul 8, 1994 - Technology & Engineering

41.Online service https://app.diagrams.net/ was used to create diagrams.

42.Online service https://balsamiq.com/ was used to create mockups.