

КОМПЬЮТЕРНАЯ МОДЕЛЬ СОЗДАНИЯ И УПРАВЛЕНИЯ ПРОДУКЦИОННЫМ ПОТЕНЦИАЛОМ АМПЕЛОЦЕНОЗОВ*

Петров В.С., *д-р с.-х. наук*, **Ильина И.А.**, *д-р техн. наук*,
Соколова В.В., *канд. с.-х. наук*

*Федеральное государственное бюджетное научное учреждение
«Северо-Кавказский федеральный научный центр садоводства, виноградарства,
виноделия» (Краснодар)*

Попова Д.В.

*ООО «Малое инновационное предприятие «Ампелоинформпродукт»
(Краснодар)*

Реферат. Обоснована необходимость создания компьютерной модели управления ростовыми процессами и продукционным потенциалом винограда на основе эффективного использования научной информации. Разработана системная архитектура прототипа программного обеспечения методов управления комплексной устойчивостью виноградных агроэкосистем для обеспечения максимальной реализуемости биологического потенциала и самовоспроизводства ампелоценозов в условиях техногенной интенсификации производства и изменений климата.

Ключевые слова: ампелоценоз, продукционный потенциал, комплексная устойчивость, компьютерная модель, программное обеспечение

Summary. The necessity of creating a computer model for managing growth processes and production potential of grapes based on the effective use of scientific information is substantiated. The system architecture of the prototype software for managing the integrated stability of grape agroecosystems was developed to ensure the maximum feasibility of the biological potential and self-reproduction of ampeloceneses under conditions of technogenic intensification of production and climate change.

Key words: ampelocenosis, production potential, integrated stability, computer model, software

Введение. Для оптимальной реализации продукционного потенциала виноградному растению необходимы определенные условия воспроизводства. На территории Российской Федерации возделывать виноград в промышленных объемах возможно только на площади 2,2 % от всей территории страны в пределах границ южных регионов: в Краснодарском и Ставропольском краях, Ростовской области, республиках Крым, Дагестан, Кабардино-Балкария, Чечня. В этих регионах преобладающим является умеренно-континентальный климат с характерными проявлениями нестабильных условий окружающей среды, частыми аномалиями в форме низкотемпературных и водных стрессов, которые в свою очередь сопровождаются нарушением ростовых и продукционных процессов, снижением продуктивности винограда. Недобор урожая винограда, при среднем уровне реализации потенциала хозяйственной продуктивности 60 %, только в Краснодарском крае составляют 36 тыс. тонн в год [1].

* Поддержано грантом №19-44-230021 р_а Российского фонда фундаментальных исследований и администрации Краснодарского края

В большинстве случаев, применяемые технологии в Российском промышленном виноградарстве не отвечают современным требованиям конкурентоспособного производства по уровню реализации потенциала хозяйственной продуктивности, качеству винограда и винопродукции, экологической и эдафической устойчивости ампелоценозов, естественному процессу воспроизводства малого биологического круговорота и почвенного плодородия, экономическим показателям. Объективной причиной этого можно считать возделывание винограда в жестких условиях умеренно-континентального климата юга России, в отличие от западноевропейских мягких погодных условий, что сопровождается увеличением финансовых издержек на получение единицы продукции, ростом энергозатрат в технологическом процессе, сокращением нормативного срока эксплуатации насаждений. Среди субъективных причин, снижающих конкурентоспособность отечественного виноградарства, - это неэффективное использование возобновляемых природных ресурсов (свет, тепло, вода, питание) в продукционном процессе винограда, создание насаждений без учета ресурсного потенциала агротерриторий и биологических особенностей сортов, использование некачественного посадочного материала винограда, применение энергоемких, трудозатратных технологий и т.д.

В современных условиях углубления интеграции российской экономики в мировой рынок, для обеспечения конкурентоспособности отечественного виноградарства необходим переход к новому технологическому укладу, на новый уровень агротехнологий, что согласуется с приоритетными направлениями Стратегии научно-технологического развития РФ, утвержденной Указом Президента РФ от 01.12.2016 г. № 642.

Современная концепция стабильного конкурентоспособного производства винограда основана на формировании устойчивых саморегулирующихся высокоадаптивных агроценозов, рациональном природопользовании, улучшении среды произрастания винограда.

В современной науке накоплен большой объем научных знаний, позволяющий решать актуальные проблемы в отрасли виноградарства. В силу разрозненности информации, необходимости большого количества времени на её поиск и анализ, отсутствие специальных механизмов она используется не в полном объеме. Для эффективного использования научных знаний в организации современного производства винограда необходимы условия и специалисты, оборудованное рабочее место, время и специальные компьютерные модели – то, чего нет в современных хозяйствах.

Объекты и методы исследований. Объекты исследований – зоны промышленного возделывания винограда Краснодарского края и сорта винограда различного эколого-географического происхождения В качестве методологической базы для исследования использовали современные методы моделирования и информационные технологии.

Обсуждение результатов. Положительная динамика в современном виноградарстве достигается при нарастающей антропогенной интенсификации производства, использовании отечественных и интродуцированных сортов, высокоинтенсивных технологий, применении современных средств механизации, широкого спектра удобрений, средств защиты растений, физиологически активных препаратов. В большинстве случаев применяемые технологии несовершенны и не отвечают современным требованиям эффективно конкурентоспособного производства по критериям продуктивности, качеству винограда и винопродукции, экологической и эдафической устойчивости ампелоценозов, экономическим показателям. Антропогенная нагрузка в ампелоценозах возросла до масштабов, угрожающих воспроизводству природных ресурсов, обозначились тенденции деградации почвы, нарушения экологии, ухудшения качества пищевой продукции.

В связи с этим требуется разработка системных биоинформационных технологий управления ампелоценозами, учитывающих комплексное использование в производственном процессе природных, биологических и антропогенных ресурсов, а также необходимых для полноценной оценки предлагаемых сортовых и технологических решений оцифрованных банков данных генетических ресурсов винограда, их соответствия различным почвенно-климатическим условиям среды, сортовых технологий и тенденций изменения климата.

Использование системного подхода в наибольшей степени обеспечивает реализацию всей совокупности факторов в производственном процессе, определяющих стабильность плодоношения, продуктивность и качество винограда, а создание единого банка данных генетических ресурсов винограда и их качественных показателей, проявляемых в различных почвенно-климатических условиях зон возделывания, даст возможность оперативно подбирать оптимальный сортимент для всех зон возделывания культуры с учетом изменения климата, а также создавать новые высокоэффективные сортоформы.

Ввиду вышеизложенного требуется создание компьютерной программы для принятия управленческих решений, базирующейся на обширных данных о сортах (учитывается более 80-ти параметров, при этом часть характеристик динамична в зависимости от места возделывания) и условиях среды (более 20 параметров). В этой программе должны быть обобщены статистические данные об генотипе ампелографической коллекции [2], ресурсном агроэкологическом потенциале областей возделывания винограда с помощью созданных моделей расчета базовых и оперативных технологий на основе мониторинга среды, диагностики и накопленной информации. Она должна позволять генерировать рекомендации по управлению продуктивным потенциалом ампелоценозов.

Для разработки прототипа программы управления продуктивным потенциалом ампелоценозов (создания алгоритмов расчетов, баз данных и систем управления базами данных), оцифровки и внесения в базы собранных данных, должен быть использован методический инструментарий по оптимизации параметров функциональной устойчивости производственных процессов в ампелоценозах по критериям рационального природопользования и ресурсосбережения, содержащий аналитический, конструкционный и математический аппараты с использованием аналитической и статистической базы, программные пакеты прикладных программ по обработке информации.

Первостепенным при создании программного продукта является разработка его системной архитектуры, так как от того, насколько хорошо организована программа, во многом зависит то, как она будет работать.

Архитектура программы – это базовая организация системы, воплощенная в ее компонентах, их отношениях между собой и с окружением, а также принципы, определяющие проектирование и развитие системы [3].

Архитектура разрабатываемого проекта и ее выбор зависит от задач, стоящих перед разработчиками, а также функциональных и эксплуатационных требований.

На этапе анализа требований к системе, основное внимание уделяется определению конкретной информации, которую необходимо получать для управления продуктивностью ампелоценозов вне зависимости от того, как это сделать.

Основными пользователями разрабатываемой программы будут являться:

- специализированные виноградопроизводящие хозяйства,
- винодельческие заводы,
- крестьянско-фермерские хозяйства,
- научные учреждения,
- министерства, департаменты сельского хозяйства и отраслевые управления,
- питомники и коммерческие компании, производящие посадочный материал винограда.

Информация, которую эти пользователи хотели бы получить:

– хозяйственно-ценные характеристики сортов (физико-химический состав, дегустационные оценки – для качественного виноделия, сроки созревания, стабильность плодоношения и период созревания – для подбора сортимента, гарантирующего целесообразность, стабильность и экономический успех закладок виноградников, обеспечивающий равномерное поступление продукции в торговлю и на перерабатывающие предприятия (т.е. сортовой конвейер), донор устойчивости к различным болезням и вредителям – для создания новых сортов, обладающих устойчивостью к определенным заболеваниям и вредителям и т.д.);

– характеристики областей возделывания (почвенные и климатические характеристики, рельеф местности – для выявления наиболее подходящих для виноградников земель, климатические характеристики – для выбора сортимента, наиболее подходящего для данной местности, а также культуры ведения и др.);

– площади посадок различных сортов винограда и показатели их урожайности (количественные и качественные) – для выявления наиболее благоприятных зон возделывания для конкретных сортов, исключения переизбытка конкретных сортов в ассортименте региона (или области), выявления недостающих сортов для создания сортового конвейера области;

– базовые технологии возделывания – перечень рекомендаций, включая оптимизацию размещения сортов, подбора подвоев, формирования и ведения кустов, содержание и обработку почвы, режимы минерального питания, системы защиты растений винограда от вредных организмов на этапе закладки виноградников;

– оперативные технологии возделывания – перечень рекомендаций, включая обрезку лоз, нагрузку кустов глазками и урожаем, планируемую урожайность, оросительную норму, регламенты режима питания и внесения минеральных удобрений, защиты растений от болезней и вредителей, экономическую целесообразность, контроль качества продукции и экологической безопасности агротерриторий на этапе вегетации виноградников;

– оценку эффективности действующих виноградников – для выявления причин низкого или недостаточно высокого уровня продуктивности и качества получаемого урожая, построения путей решения данной проблемы;

– прогнозирование вспышек очагов болезней и вредителей винограда на долгосрочную и краткосрочную перспективу – для заблаговременного принятия мер по защите ампелоценозов и предотвращения потери урожайности.

Исходя из основных требований, предъявляемых к программе ее пользователями, требуется выявить основные блоки, из которых должен состоять прототип разрабатываемого программного обеспечения, а именно:

–база данных сортов;

–база данных агроэкологических зон возделывания;

–база данных площадей посадок сортов (с данными об их урожайности и качестве урожая);

–СУБД (система управления базами данных);

–модули подбора сортов по заданным критериям;

–расчета базовых технологий, оперативных технологий;

–оценка эффективности насаждений (с использованием самообучающихся систем), прогнозирования вспышек очагов болезней и вредителей;

–оценка природного потенциала агроэкологических территорий Краснодарского края (с использованием методов ГИС).

Для построения прогнозов по вспышкам болезней и вредителей, необходимы:

–база данных погодно-климатических условий (включая метеорологические характеристики);

- база данных экономико-географических характеристик территорий;
- база данных об очагах вспышек заболеваний за несколько лет.

Для оценки природных потенциалов сельскохозяйственных зон, необходимы:

- база данных виноградо-винодельческих хозяйств;
- база данных площадей посадок сортов винограда с разбивкой по хозяйствам (с данными об их урожайности и качестве урожая);
- база данных погодно-климатических условий, включая метеорологические характеристики;
- база данных почвенных условий;
- база данных характеристик рельефа;
- база данных экономико-географических характеристик территорий.

Чтобы разработать систему, способную удовлетворить требования пользователей необходимо определить, как будут реализованы задачи, решаемые на этапе анализа.

Все запросы, которые пользователи могут предъявлять системе, делятся на два сценария:

1) запрос к базам данных – пользователь обращается к СУБД, сообщая критерии подбора данных, далее (→) СУБД извлекает из нужной базы данных данные, удовлетворяющие заданным критериям и возвращает их пользователю;

2) запрос к модулю расчета – пользователь делает запрос к модулю соответствующего расчета, сообщая необходимые для проведения расчета данные → модуль формирует критерии отбора данных из баз данных и делает запрос к СУБД на получение данных соответствующих сформированным критериям → СУБД извлекает из нужной базы данных данные, удовлетворяющие заданным критериям, и передает их модулю → модуль производит расчет, используя полученные из базы данных данные и параметры, указанные пользователем и передает результаты расчетов пользователю (рис. 1).

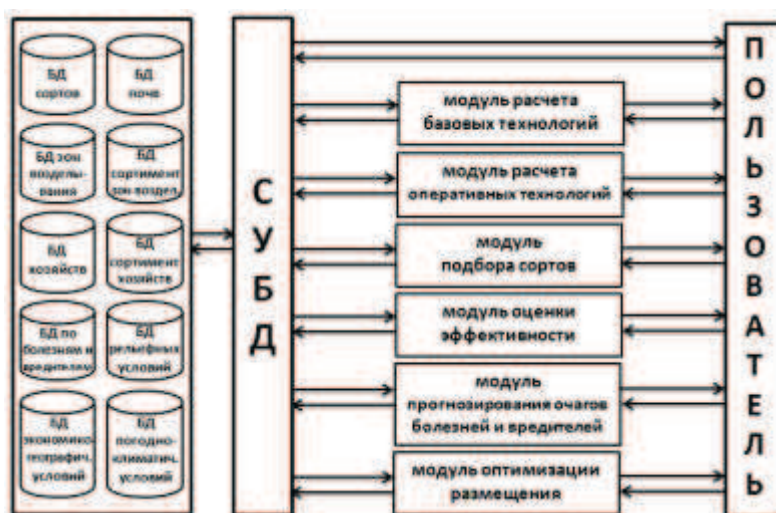


Рис.1. Схема получения информации из программы

Помимо данных, которые пользователи ожидают получить из системы, она должны удовлетворять и другим их требованиям:

1. Быстрота проведения расчетов;
2. Достоверность получаемой информации;
3. Понятный интерфейс;
4. Удобство получения данных;
5. Безопасность.

Для удовлетворения проектируемой системы различным атрибутам качества применяются различные архитектурные шаблоны. Каждый шаблон имеет свои задачи и свои недостатки. Ввиду того, что разрабатываемая программа имеет многопользовательскую (сетевую) архитектуру (есть ограниченное число ресурсов, к которым требуется ограниченный правами доступ большого числа потребителей), ее эффективно реализует система, построенная по принципу «клиент – сервер», где клиент – заказчик услуги, а сервер – поставщик услуги.

В такой системе клиенты и сервер могут быть различными физическими устройствами и взаимодействовать через компьютерную сеть или сеть Интернет посредством сетевых протоколов. Данная система удобна в первую очередь тем, что в ней можно разделить программный код поставщика и заказчика, а также нагрузку между участниками процесса обмена информацией (рис. 2).

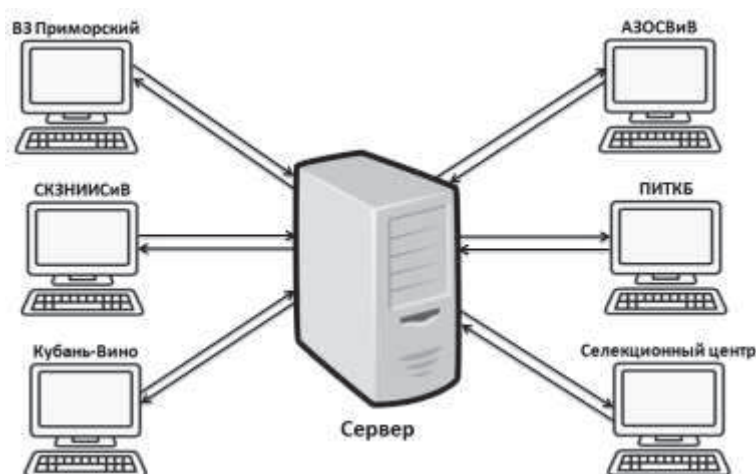


Рис. 2. Схема взаимодействия Клиентов и Сервера

К одному серверу может обращаться сразу несколько клиентов. Их количество зависит от мощности сервера и от того, что хочет получить клиент от сервера.

Существует два вида архитектуры взаимодействия «клиент-сервер»:

1. Двухуровневая архитектура – обработка запроса происходит на одной машине, без использования сторонних ресурсов (клиент делает запрос, сервер обрабатывает запрос клиента), данная архитектура предъявляет жесткие требования к серверу (в части его производительности), но является наиболее надежной.

2. Трехуровневая архитектура – обработка запроса происходит сразу несколькими серверами (клиент делает запрос к движку СУБД (серверу 1), СУБД интерпретирует запрос и реализует взаимодействие с файловой системой (сервером 2) – хранилищем данных).

В отличие от первого подхода, во втором случае идет четкое разделение программ, отвечающих за хранение и за обработку данных, такой подход позволяет снизить нагрузку на сервер, т.к. распределяет операции, но он менее надежен, чем в первом случае.

Преимущества трехуровневой архитектуры:

- меньшая нагрузка на клиентские приложения («тонкий клиент»);
- между клиентом и сервером приложений (СП) передается минимальный поток данных (входные параметры (аргументы) и получаемые по итогам запроса значения);
- снижение нагрузки на сервер данных (увеличение быстродействия системы);

- дешевый трафик между СП и СУБД (сам трафик может быть большим, но т.к. это трафик локальной сети, то его пропускная способность велика и дешева (в случае, если СП и СУБД расположены на одной машине, имеем нулевой трафик));
- сервер приложения может быть запущен на одном или нескольких компьютерах (разделение нагрузки);
- возможность обновлять ПО и наращивать мощность дешевле, чем в двухуровневой системе (масштабируемость).

Недостатки: сервер может стать узким местом системы, при его недоступности становится недоступна вся система и поддержка работы данной системы требует отдельного специалиста – системного администратора [4].

Так как преимуществ такой системы больше, чем недостатков и в связи с обширностью итоговых баз данных, принято решение о применении трехуровневой архитектуры «клиент–сервер».

На основе проведенного анализа, системная архитектура разрабатываемого приложения (прототипа программы управления продуктивностью виноградных насаждений) должна быть модифицирована введением дополнительного блока – сервера приложений.

Можно выделить четыре основных блока проектируемой системы:

- пользователь (клиент) – осуществляет вызов функций СП, формирование запросов к программе,
- СП – осуществляет выполнение функций, формирование запросов к СУБД, содержит модули расчета технологических процессов и прогнозов,
- СУБД (сервер СУБД) – осуществляет выполнение запросов, обработку транзакций, хранение и управление файловой системой,
- БД – хранилище данных (рис. 3).



Рис. 3. Схема трехуровневой архитектуры взаимодействия в системе «клиент-сервер»

Следующим этапом разработки системной архитектуры программы управления продуктивностью ампелоценозов является выбор языков программирования, программ, посредством которых будет создан прототип программного обеспечения.

Во-первых, требуется определить, посредством каких инструментов будут реализованы базы данных и СУБД.

База данных – это файл (или группа файлов), представляющий собой упорядоченный набор записей, имеющих единообразную структуру и организованных по единому шаблону (как правило, в табличном виде). База данных может состоять из нескольких таблиц.

Существует три основных типа организации данных:

- иерархическая (один элемент считается главным, остальные – подчиненными, поиск данных осуществляется последовательным списком от уровня к уровню и может оказаться довольно трудоемким);

- сетевая (позволяет устанавливать дополнительно к вертикальным иерархическим связям горизонтальные, что существенно облегчает процесс поиска требуемых элементов);

- реляционная (под записью понимается строка прямоугольной таблицы, элементы записи образуют столбцы этой таблицы (столбцы имеют уникальное имя, элементы одного столбца имеют одинаковый тип), наличие одинаковых строк в таблице исключено, преимущество – наглядность и понятность организации данных, скорость поиска нужной информации) [5].

В разрабатываемой программе использована реляционная база данных с сетевым доступом.

Язык программирования баз данных – эта группа языков отличается от алгоритмических языков прежде всего решаемыми задачами. Они обеспечивают выполнение следующих операций: создание/модификация свойств, удаление таблиц, поиск, отбор, сортировка информации по запросам пользователей, добавление новых записей, модификация существующих записей, удаление существующих записей.

SQL (Structured Query Language) – структурированный язык запросов, который позволяет выполнять эффективную обработку баз данных, манипулируя не отдельными записями, а группами записей.

Для управления большими базами данных и их эффективной обработки разработаны СУБД. Практически в каждой СУБД помимо поддержки языка SQL имеется также свой уникальный язык, ориентированный на особенности этой СУБД и не переносимый на другие системы.

Для выбора оптимальной СУБД, рассмотрены шесть наиболее распространенных: Oracle, MySQL, SQL Server, PostgreSQL, DB2, Microsoft Access (так называемая, настольная СУБД) [6-10].

Проведен сравнительный анализ их преимуществ и недостатков, по итогам которого для разработки приложения выбрана СУБД MySQL (основными критериями отбора были быстрдействие, функциональность, доступность, безопасность).

Данная СУБД является одной из самых популярных для веб-приложений, прекрасно документирована, предлагает много функций, имеет простой в использовании интерфейс, и пакетные команды, которые позволяют удобно обрабатывать огромные объемы данных, надежна и не стремится подчинить себе все доступные аппаратные ресурсы, а кроме того это бесплатный пакет программ.

На сегодняшний день наиболее популярная и эффективная связка для создания мощных динамических сайтов – это язык веб-программирования PHP и СУБД MySQL. Поэтому, алгоритмы расчетов технологических процессов и оценок эффективности действующих виноградников будут исполнены на PHP. У данного языка есть ряд преимуществ конкретно для разрабатываемого проекта: имеет развитую поддержку баз данных; может быть развернут почти на любом сервере; является свободным программным обеспечением; имеется огромное количество библиотек и расширений языка [11-13].

Для отработки основных процессов разработки сайта производится установка локального веб-сервера Denwer, включающего сразу набор всех атрибутов Apache, PHP, MySQL [14-15].

Для создания Web-страниц предпочтителен язык **HTML**, так как это общеизвестный язык для оформления документов. Он очень прост и содержит элементарные команды форматирования текста, добавления рисунков, задания шрифтов и цветов, организации ссылок и таблиц. Однако возможности HTML сильно ограничены, этот код не обладает ни динамичностью, ни гибкостью. HTML в чистом виде не позволяет создавать действительно привлекательные сайты, и тем более не позволяет динамически управлять контентом. Средствами HTML можно управлять логической структурой веб-страниц. Для задания внешнего вида используется отдельный язык CSS. Раздельное описание логической структуры и представления документа позволяет более гибко управлять внешним видом документа и минимизировать объем повторяющегося кода.

Так как разметка HTML не дает возможности создавать персональные настройки, для создания сценариев, позволяющих динамически управлять сайтом, будет использован PHP, позволяющий генерировать оригинальные страницы в зависимости от самых разнообразных настроек, включая тип операционной системы, браузер и время суток. Язык сценариев PHP разработан для создания WEB-приложений с возможностью генерирования HTML-кода и внедрения скриптов в HTML-код. PHP относится к языкам программирования, сценарии на которых исполняются на стороне сервера.

Для придания интерактивности веб-страницам, наиболее широкое применение в браузерах нашел язык программирования JavaScript, основной задачей которого является манипулирование элементами DOM-модели web-страницы. Это язык, который позволяет активно управлять структурой страницы, манипулировать ее элементами. Кроме того, в контексте разрабатываемой модели JavaScript используется и для первичной проверки данных, которые пользователь вводит в формы [15-17].

Таким образом, прототип разрабатываемой компьютерной программы должна состоять из четырех блоков:

- электронная база данных и СУБД (реализация с помощью MySQL);
- алгоритмы расчетов (реализация на языке PHP);
- серверная часть сайта (реализация средствами языка PHP);
- клиентская часть сайта (реализация средствами HTML, JavaScript, CSS).

Схема работы системы в таком случае будет иметь следующий вид:

1. Браузер клиента отправляет запрос.
2. На стороне web-сервера осуществляется идентификация и авторизация пользователя, а затем определяется тип запрашиваемого ресурса и загружает обработчик, который создает объектную модель запрашиваемой страницы и обрабатывает действия пользователя.
3. В ходе генерации ответа может потребоваться обращение к БД, в этом случае выполняющая среда обращается к серверу БД.
4. Запрос на операцию передается с БД передается серверу MySQL.
5. Сервер MySQL осуществляет обработку запроса, выполняя соответствующие операции с БД.
6. Провайдер данных передает результаты запроса объекту страницы.
7. Объект страницы (с учетом полученных данных) осуществляет визуализацию графического интерфейса страницы и направляет результаты в выходной поток.
8. Сервер IIS отправляет содержимое сгенерированной страницы клиентскому браузеру [15].

Итоговый вид системной архитектуры прототипа создаваемой программы представлен на (рис. 4).

Для создания прототипа программы управления продуктивностью виноградных насаждений, баз данных, алгоритмов расчетов, а также для доступа пользователей к разрабатываемой программе, необходимо следующее программное обеспечение: Windows 7, Internet Explorer (браузер для просмотра веб-страниц), Dreamweaver (программа для работы с HTML-кодом), STYLIZER (редактор CSS), Java-IDE Eclipse (среда разработки Java), готовый дистрибутив Denwer (содержит в себе сразу веб-сервер Apache, интерпретатор PHP и базу данных MySQL), MapInfo (программа ГИС).

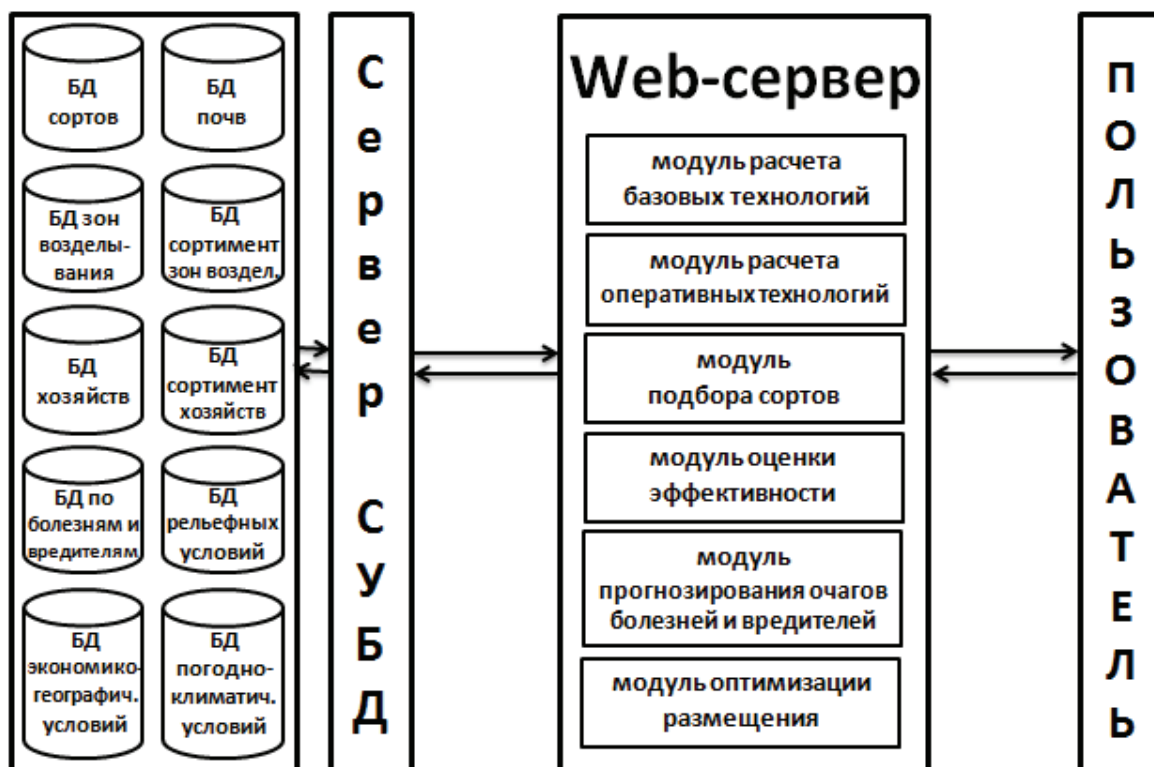


Рис. 4. Системная архитектура прототипа разрабатываемой программы по управлению продуктивным потенциалом ампелоценозов

Проектирование пользовательского интерфейса с последующим его развитием в процессе разработки самого программного продукта не менее важная часть, чем разработка системной архитектуры или алгоритмов расчетов. Работа пользователя осуществляется с экранными формами, содержащими объекты управления, панели инструментов с пиктограммами режимов и команд обработки, простой и лаконичный интерфейс обеспечит возможность работы с программой пользователей всех уровней подготовки. Полнота использования потенциальных возможностей имеющихся ресурсов зависит от качества пользовательского интерфейса. Хорошо разработанный интерфейс должен отвечать следующим требованиям:

- обеспечивать удобный доступ ко всему перечню функциональных возможностей, предусмотренных данным приложением;
- навигационные элементы должны быть выполнены в удобной для пользователя форме;

- ввод-вывод данных, прием управляющих команд и отображение результатов их исполнения должны выполняться в интерактивном режиме;
- интерфейс должен быть рассчитан на преимущественное использование манипулятора типа «мышь», клавиатурный режим ввода должен использоваться главным образом при заполнении и/или редактировании текстовых и числовых полей экранных форм;
- графические объекты должны сохранять свое стандартизованное назначение и местоположение на экране;
- в случае ошибок, вызванных неверными действиями пользователей, неверным форматом или недопустимыми значениями входных данных, пользователю должны выдаваться соответствующие сообщения, после чего программа должна возвращаться в рабочее состояние, предшествовавшее неверной команде или некорректному вводу данных;
- сообщения и результаты, выдаваемые приложением, не должны требовать дополнительных пояснений.

Большинство программных продуктов прикладного характера, ориентированных на конечного пользователя, работают в диалоговом режиме взаимодействия с пользователем таким образом, что ведется обмен сообщениями, влияющими на обработку данных. В диалоговом режиме под воздействием пользователя осуществляются запуск функций (методов) обработки, изменение свойств объектов, производится настройка параметров выдачи информации на печать и т.п.

Наиболее эффективны и просты для реализации диалоговые системы с жестким сценарием диалога, которые предоставлены в виде:

- *меню* – диалог инициируется программой; пользователю предлагается выбор альтернативы функций обработки из фиксированного перечня; предоставляемое меню может быть иерархическим и содержать вложенные подменю следующего уровня;
- *действия запрос-ответ* – фиксирован перечень возможных значений, выбираемых из списка, или ответы типа Да/Нет;
- *запрос по формату* – с помощью ключевых слов, фраз или путем заполнения экранной формы с регламентированным по составу и структуре набором реквизитов осуществляется подготовка сообщений.

В составе инструментальных средств СУБД содержатся построители меню, с помощью которых создается ориентированная на конечного пользователя совокупность режимов и команд в виде главного меню и вложенных подменю. Конструктор экранных форм СУБД используется для разработки форматов экранного ввода и редактирования данных базы данных и входной информации, управляющей работой программного продукта.

Цель разработки хорошего пользовательского интерфейса – формировать на экране среду, которая не только содействовала бы пониманию пользователем представленной информации, но и позволяла бы сосредоточиться на наиболее важных ее аспектах.

Разработанный пользовательский интерфейс отвечает указанным выше правилам.

На главной странице прототипа программы управления потенциалом ампелоценозов необходимо расположить ссылки на формы документов, которые доступны пользователю, для удобства поиска необходимых форм, ссылки должны быть сгруппированы по функциональным блокам, также должны быть ссылки на инструкции по работе с программой с описанием ее возможностей, кроме того, должна быть указана информация о том, что данный программный продукт разработан при поддержке СКФНЦСВВ и Фонда поддержки инноваций.

Выводы. На основе анализа основных требований пользователей к программному обеспечению процесса управления базами данных по интересующему вопросу и средств для получения необходимой информации разработана системная архитектура прототипа программы: выбран оптимальный архитектурный шаблон, удовлетворяющий всем атри-

бутам качества, предъявляемым к системе; определен тип организации данных проектируемых баз данных; проведен сравнительный анализ наиболее применяемых языков систем управления базами данных, на основании которого определен наиболее подходящий; выбран язык программирования алгоритмов расчетов для эффективной связи с подобранным языком СУБД, обладающий преимущественными характеристиками для разрабатываемого проекта; определены группы языков для написания динамического и интерактивного web-сайта. На основе анализа основного контингента потребителей создаваемого продукта, разработан интерфейс главной страницы прототипа программы управления продуктивным потенциалом виноградников.

Литература

- 1 Петров В.С. Потенциал хозяйственной продуктивности винограда, его реализация в условиях умеренно континентального климата юга России // Магарач. Виноградарство и виноделие. 2016. № 1. С. 20 – 22.
- 2 Егоров Е.А., Ильина И.А., Петров В.С., Панкин М.И., Ильницкая Е.Т., Талаш А.И., Лукьянов А.А., Лукьянова А.А., Коваленко А.Г., Большаков В.А., др. Анапская ампелографическая коллекция (биологические растительные ресурсы): монография. Краснодар: ФГБНУ Северо-Кавказский федеральный научный центр садоводства, виноградарства, виноделия, 2018. 194 с. <https://elibrary.ru/item.asp?id=36539666>
- 3 Гвоздева Т.В., Баллод Б.А. Проектирование информационных систем. Ростов-на-Дону: Феникс, 2009. 512 с.
- 4 Березюк, А.Т., Приходько, С.А. Выбор и конфигурирование комплекса технических средств информационной системы. Хабаровск, 2010. 122 с.
- 5 Бурмистров А.В., Белов Ю.С. Недостатки реляционных баз данных // Электронный журнал: наука, техника и образование. 2015. № 3 (3). С. 25 – 34.
- 6 Драч В.Е., Родионов А.В., Чухраева А.И. Выбор системы управления базами данных для информационной системы промышленного предприятия // Электромагнитные волны и электронные системы. 2018. Т. 23. № 3. С. 71 – 80.
- 7 Петухов И.С. Алгоритм определения необходимых индексов для оптимизации запросов с соединением двух таблиц в СУБД MYSQL (INNODB) // Научный вестник ГосНИИ ГА. 2017. № 16. С. 98 – 107.
- 8 Панченко И. PostgreSQL: вчера, сегодня, завтра // Открытые системы. СУБД. 2015. № 3. С. 34 – 37.
- 9 MySQL. Справочник по языку / Компания MySQL AB. – М.: Вильямс, 2005. – 432 с.
- 10 Веллинг Л., Томсон Л. MySQL. Учебное пособие / Люк Веллинг, Лора Томсон. М.: Вильямс, 2005. 304 с.
- 11 Аткинсон, Л. PHP 5. Библиотека профессионала / Л. Аткинсон, З. Сураски – М.: Вильямс, 2006. 944 с.
- 12 Гутманс, Э., PHP 5. Профессиональное программирование / Э. Гутманс, С. Баккен, Д. Ретанс. С.-Пб.: Символ-Плюс, 2006. 704 с.
- 13 Котеров, Д. PHP 5 / Д. Котеров. С.-Пб.: BHV, 2004. 1120 с.
- 14 Конверс, Т. PHP 5 и MySQL. Библия пользователя. PHP5 / Т. Конверс, Дж. Парк, К. Морган. М.: Диалектика, 2008. 1216 с.
- 15 Ломов, А.Ю. Apache, Perl, MySQL: практика создания динамических сайтов. Самоучитель / А.Ю. Ломов. С.-Пб.: БХВ-Петербург, 2007. 368 с.
- 16 Дунаев, В. Сценарии для Web-сайта. PHP и JavaScript / Вадим Дунаев – С.-Пб.: БХВ-Петербург, 2006. 576 с.
- 17 Дари, К. AJAX и PHP. Разработка динамических веб-приложений / Д. Кристиан, Б. Бринзаре, Ф. Черчез-Тоза, М. Бусика [и др.]; С.-Пб.: Символ-Плюс, 2009. 336 с.