ГБПОУ Республика Марий Эл

«Марийский радиомеханический техникум»

Допущен к защите

Зам.директора по УМР

\_\_\_\_\_\_\_\_И.Ю. Бурханова

«\_\_\_»\_\_\_\_\_\_\_\_\_ 2018 г.

РАЗРАБОТКА ВЕБ-ОРИЕНТИРОВАННОГО МОТИВАЦИОННОГО СЕРВИСА С ЭЛЕМЕНТАМИ ГЕЙМИФИКАЦИИ

Пояснительная записка к дипломной работе

МРМТ.09.02.03.018 ПЗ-ДР

Рецензент Разработчик:

\_\_\_\_\_\_\_\_\_\_\_ Г.Н. Гайфутдинова студент группы П-41

\_\_\_\_\_\_\_\_П. Е. Шевелев

Нормоконтролер: Руководитель

\_\_\_\_\_\_\_\_\_\_\_\_Е.В. Матвеева \_\_\_\_\_\_\_\_А.Н. Савинов

Оценка Экзаменационной комиссии по защите\_\_\_\_\_\_\_\_\_\_\_

Председатель ГЭК\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.Л. Цивин

2018

ГБПОУ Республики Марий Эл «Марийский радиомеханический техникум»

Допущен к защите

Зам.директора по УМР

\_\_\_\_\_\_\_\_И.Ю. Бурханова

«\_\_\_»\_\_\_\_\_\_\_\_\_ 2018 г.

ЗАДАНИЕ НА ДИПЛОМНУЮ РАБОТУ

Студент группы\_\_\_\_*П-41*\_\_\_\_ специальность\_\_\_\_\_\_\_\_\_\_*09.02.03*\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*Шевелев Павел Евгеньевич*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия имя отчество)

Тема дипломной работы\_\_\_\_\_\_\_\_\_\_\_\_\_*Разработка веб-ориентированного*\_\_

\_\_\_\_\_*мотивационного сервиса с элементами геймификации*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Исходные данные\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Содержание дипломной работы:

Введение \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Теоретический раздел\_\_\_\_\_\_\_\_\_\_\_\_\_*Теоретические основы*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_*фенологических наблюдений*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Аналитический раздел\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*Анализ проблем*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_*фенологических наблюдений*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Исследовательский раздел \_\_\_\_\_\_\_\_\_\_\_\_\_*Разработка сайта для*\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_*сбора фенологических данных*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Заключение\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель дипломной работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

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

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

(должность, место работы, ФИО)

Задание принял к исполнению\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата, ФИО)

Задание рассмотрено на заседании цикловой комиссии информационных и радиотехнических дисциплин «\_\_\_»\_\_\_\_\_\_2018 г.

Председатель ЦК\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Е.В. Матвеева

АННОТАЦИЯ

Целью данного проекта является разработка сайта для мотивации пользователей с использованием элементов геймификации. Проведено исследование актуальности данной темы и состоянии современных сервисов по достижению целей. Проанализированы существующие системы мотиваций пользователей. Произведен анализ и выбор программных средств.

THE SUMMARY

The purpose of this project is to develop a website for motivating users using elements of gaming. The study of the relevance of this topic and the state of modern services to achieve the goals. The existing systems of user motivation are analyzed. The analysis and choice of software are made.

ПЕРЕЧЕНЬСОКРАЩЕННЫХУСЛОВНЫХОБОЗНАЧЕНИЙ

РГО – РусскоеГеографическое сообщество

ГО – Географическое общество

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 6](#_Toc453323353)

[1ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 8](#_Toc453323354)

[1.1 Исследование актуальности сбора фенологических данных 8](#_Toc453323355)

[1.2 Исследование современного состояния фенологии в России и в мире 11](#_Toc453323356)

[2АНАЛИТИЧЕСКАЯ ЧАСТЬ 16](#_Toc453323357)

[2.1 Обзор систем сбора фенологических данных 16](#_Toc453323358)

[2.2 Разработка системы сбора фенологических данных 17](#_Toc453323359)

[3ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ 21](#_Toc453323360)

[3.1 Выбор средств разработки сайта 21](#_Toc453323361)

[3.1.1 Обоснование выбора среды разработки 22](#_Toc453323362)

[3.1.2Обоснование выбора языка разработки 25](#_Toc453323363)

[3.1.3 Обоснование выбора СУБД 26](#_Toc453323364)

[3.2 Разработка движка сайта 29](#_Toc453323365)

[3.3Разработкабазыданных 38](#_Toc453323366)

[3.3.1Реляционная модель данных 38](#_Toc453323367)

[3.3.2 Таблицаusers 38](#_Toc453323368)

[3.3.3 Таблицаphenomens 39](#_Toc453323369)

[3.3.4Таблица comments 40](#_Toc453323370)

[3.4 Разработка интерфейсов пользователя 40](#_Toc453323371)

[3.5 Тестирование интерфейса методом GOMS 45](#_Toc453323372)

[ЗАКЛЮЧЕНИЕ 50](#_Toc453323373)

[СПИСОК ЛИТЕРАТУРЫ 51](#_Toc453323374)

[ПРИЛОЖЕНИЕ А 52](#_Toc453323375)

[ПРИЛОЖЕНИЕ Б 55](#_Toc453323376)

[ПРИЛОЖЕНИЕ В 57](#_Toc453323377)

# ВВЕДЕНИЕ

Мотивация (от лат. movere) — побуждение к действию; психофизиологический процесс, управляющий поведением человека, задающий его направленность, организацию, активность и устойчивость; способность человека деятельно удовлетворять свои потребности.

Мотив (лат. moveo — «двигаю») — это обобщённый образ (видение) материальных или идеальных предметов, представляющих ценность для человека, определяющий направление его деятельности, достижение которых выступает смыслом деятельности. Мотив представлен субъекту в виде специфических переживаний, характеризующихся либо положительными эмоциями от ожидания достижения данных предметов, либо отрицательными, вызванными неполнотой настоящего положения. Для осознания мотива требуется внутренняя работа. Впервые термин «мотивация» употребил в своей статье А. Шопенгауэр.

Мотив — одно из ключевых понятий психологической теории деятельности, разрабатывавшейся ведущими советскими психологами А. Н. Леонтьевым и С. Л. Рубинштейном. Наиболее простое определение мотива в рамках этой теории: «Мотив — это опредмеченная потребность». Мотив часто путают с потребностью и целью, однако потребность — это, по сути, неосознаваемое желание устранить дискомфорт, а цель — результат сознательного целеполагания, выбора предмета (объекта), удовлетворяющего требованиям мотива.

Мотивация является одним из значимых элементов в успешном управлении в современной экономической системе. Она также включает в себя множество других управленческих дисциплин, например: управление персоналом, управленческий учет, информационные технологии и другие. Долгое время о значимости мотивации даже не задумывались, очень долгое время сохранялся метод «кнута и пряника». В настоящее время менеджеры уделяют большое значение мотивации для достижения эффективности в управлении предприятий. В различных университетах вводится отдельная дисциплина, которая называется «Мотивационный менеджмент».

Современные исследования в области психологии, помогают лучше узнавать о мотивах человека, более четко определять потребности сотрудников. Ведь поняв саму суть мотивации работника, менеджер сможет построить правильный процесс мотивации для достижения конкретных целей в работе.

Для достижения поставленной цели необходимо решить ряд задач:

1. Изучить имеющиеся мотивационные системы;
2. Выбрать среду для разработки;
3. Разработать базу данных для хранения информации;
4. Выбрать язык для написания кода;
5. Разработка веб-ориентированного мотивационного сервиса с элементами геймификации.

# 1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

## 1.1 Основные виды мотиваций

Мотивация – это сила, основанная на глубинном желании, которая заставляет людей стремиться к достижению своих целей.

В настоящее время это понятие разными учёными понимается по-разному. Кто-то придерживается мнения, что мотивацией является совокупность процессов, отвечающих за побуждение и деятельность. Другие определяют мотивацию как совокупность мотивов. Мотив – это идеальный или материальный предмет, достижение которого выступает смыслом деятельности. Человеку он представляется в форме специфических переживаний, которые могут характеризоваться положительными эмоциями от достижения этого предмета, или же отрицательными, связанными с неудовлетворённостью в настоящем положении. Чтобы осознать мотив требуется проделать серьёзную внутреннюю работу. Мотив нередко путают с потребностью или целью, но потребность – это подсознательное стремление устранить дискомфорт, а цель – это результат сознательного процесса постановки цели. К примеру, голод является потребностью, желание поесть – это мотив, а еда, к которой тянутся руки человека – это цель. Мотивация является сложным психологическим феноменом, с чем и связано её многообразие.

### Внутренняя мотивация

Внутренняя мотивация (интринсивная) – это движущая человеком сила, которая возникает не из-за внешнего воздействия, а берет свое начало внутри самого человека. Хорошим примером в данном случае, будет такая эмоция, как интерес, то есть, интерес к чему-либо или к кому-либо, благодаря которому мы испытываем желание что-то делать. Когда мы что-то делаем, прежде всего из интереса, или даже можно сказать, из любви к этому делу – мы делаем это очень хорошо, потому что считаем это дело своим и видим в нем смысл. Внутренняя мотивация связана не с внешними обстоятельствами, не со стимулами и подкреплениями, а, в первую очередь, с самим содержанием деятельности.

Маленький ребенок все время что-то пробует, все исследует - и, похоже, что это определяется не столько окружающими его стимулами, а идет из него самого.

Взрослый: "Я работаю не за зарплату, я люблю свое дело. Здесь у меня есть все, чтобы быть счастливым, и мое вознаграждение - моя активность сама по себе!"

### Внешняя мотивация

Внешняя мотивация (экстринсивная) – это сила, которая вызвана в человеке внешними по отношению к нему обстоятельствами. Другими словами, что-то извне вызывает в вас определенную эмоцию, и эта эмоция высвобождает в вас энергию, которая подпитывает ваши желания. Внешнюю мотивацию нужно постоянно поддерживать, для чего необходимо принимать и преодолевать новые вызовы внешнего мира, тем самым пробуждая в себе новые сильные эмоции, как положительные, так и отрицательные. Нельзя довольствоваться достигнутым, нельзя мириться с не устраивающими вас обстоятельствами, подавляя в себе недовольство ими и обманывая себя с помощью “розовых очков”. Если вы чего-то хотите – берите это, а вернее, стремитесь к этому, приемлемым для вас способом. Не надо подавлять свои желания, ищите возможность их удовлетворить.

## Иерархия потребностей Маслоу

В своей работе «Мотивация и личность» предположил, что все потребности человека врождённые, или инстинктивные, и что они организованы в иерархическую систему приоритета или доминирования.

Потребности в порядке их очерёдности (см. рис. 1):

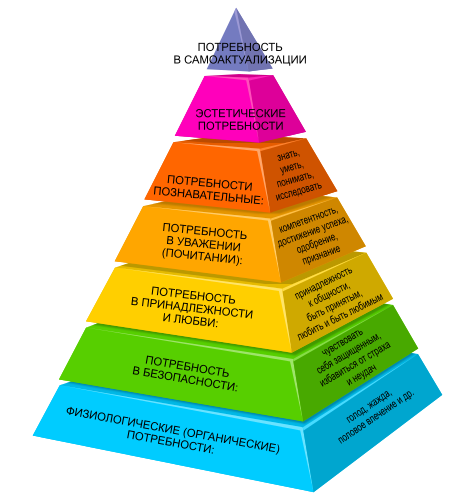


Рисунок 1 – Диаграмма иерархии потребностей

* Физиологические потребности:

Состоят из основных, первичных потребностей человека, иногда даже неосознанных. Иногда, в работах современных исследователей, их называют биологическими потребностями.

* Потребность в безопасности:

После удовлетворения физиологических потребностей их место в мотивационной жизни индивидуума занимают потребности другого уровня, которые в самом общем виде можно объединить в категорию безопасности (потребность в безопасности; в стабильности; в зависимости; в защите; в свободе от страха, тревоги и хаоса; потребность в структуре, порядке, законе, ограничениях; другие потребности).

* Потребность в принадлежности и любви:

Человек жаждет тёплых, дружеских отношений, ему нужна социальная группа, которая обеспечила бы его такими отношениями, семья, которая приняла бы его как своего.

* Потребность в признании:

Каждый человек (за редкими исключениями, связанными с патологией) постоянно нуждается в признании, в устойчивой и, как правило, высокой оценке собственных достоинств, каждому из нас необходимы и уважение окружающих нас людей, и возможность уважать самого себя. Удовлетворение потребности в оценке, уважении порождает у индивидуума чувство уверенности в себе, чувство собственной значимости, силы, адекватности, чувство, что он полезен и необходим в этом мире. Потребности этого уровня подразделяются на два класса.

В первый входят желания и стремления, связанные с понятием «достижение». Человеку необходимо ощущение собственного могущества, адекватности, компетентности, ему нужно чувство уверенности, независимости и свободы.

Во второй класс потребностей мы включаем потребность в репутации или в престиже (мы определяем эти понятия как уважение окружающих), потребность в завоевании статуса, внимания, признания, славы.

* Потребность в самоактуализации

Ясно, что музыкант должен заниматься музыкой, художник — писать картины, а поэт — сочинять стихи, если, конечно, они хотят жить в мире с собой. Человек обязан быть тем, кем он может быть. Человек чувствует, что он должен соответствовать собственной природе. Эту потребность можно назвать потребностью в самоактуализации. Очевидно, что у разных людей эта потребность выражается по-разному. Один человек желает стать идеальным родителем, другой стремится достичь спортивных высот, третий пытается творить или изобретать. Похоже, что на этом уровне мотивации очертить пределы индивидуальных различий почти невозможно.

Можно назвать ряд социальных условий, необходимых для удовлетворения базовых потребностей; ненадлежащее исполнение этих условий может самым непосредственным образом воспрепятствовать удовлетворению базовых потребностей. К ним относятся познавательные и эстетические потребности.

* Эстетические потребности

Эстетические потребности тесно переплетены и с конативными, и с когнитивными потребностями, и потому их чёткая дифференциация невозможна. Такие потребности, как потребность в порядке, в симметрии, в завершённости, в законченности, в системе, в структуре.

Потребности одного типа должны быть удовлетворены полностью прежде, чем другая потребность, более высокого уровня, проявится и станет действующей.

Теория А. Маслоу достаточно четко сочетается с теорией мотивационного комплекса, которая так же предполагает наличие пяти групп потребностей. Однако эти потребности между собой связаны циклическими, а не иерархическими связями по типу 5-элементной схемы в китайской философии, требуют первостепенного удовлетворения, а движение потребностей идёт снизу-вверх (Т)- Альдерфер, в отличие от Маслоу, считает, что движение потребностей идёт снизу-вверх и сверху вниз (); движение вверх по уровням он назвал процессом удовлетворения потребностей, а движение вниз — фрустрацией — процессом поражения в стремлении удовлетворить потребность.

# 2 АНАЛИТИЧЕСКАЯ ЧАСТЬ

## 2.1 Обзор мотивационных веб-сервисов

Мотивационные сервисы – это аналоги тайм-менеджмента, направленные на мотивацию пользователей выполнять поставленные цели. Сервисы помогают людям достичь их целей и направляют пользователя к достижению поставленных задач, получая взамен желаемый результат.

Yourmativation – это школа правильного питания которая занимается разработкой индивидуальных диет для похудения и программ тренировок. Оказывает персональные консультации клиентам в вопросах похудения или набора мышечной массы. Сервис помогает детально рассчитать и создать подробную программу питания, построенную в рамках здорового образа жизни. Еженедельные домашние задания с полезными статьями и командой поддержки, которая всегда поможет вам. Тренировки, построенные с помощью сервиса для дома и тренажерного зала. Программа тренировки разрабатывается индивидуально.

Преимущества:

* Индивидуальная диета;
* Консультация;
* Программа тренировок;
* Полезные сервисы;
* Команда поддержки;

Недостатки:

* Узкая направленность;
* Дорогие услуги;
* Отсутствие надлежащего наблюдения при выполнении упражнений;
* Несвоевременная помощь при возникновении вопросов;
* Отсутствие возможности правильно оценить ваши таланты и недостатки;

«Время Перемен» — веб-сервис, который помогает преодолеть лень и действовать с удовольствием, достигая результата. В основе всей системы лежат знания поведенческой психологии. Понимание, как люди принимают решения, что ими движет, на какие стимулы они реагируют. Система включает в себя готовые программы с ежедневными заданиями, подсказывает, что и когда делать, а сам процесс в игровой форме.

Преимущества:

* Приятный дизайн;
* Консультация;
* Большой выбор программ;
* Калькулятор калорий;
* Большой выбор блюд с подсчетом калорий;

Недостатки:

* Долгая загрузка сайта;
* Непонятно расположение некоторых элементов;
* Узкая направленность;
* Отсутствие заявленного игрового процесса;
* Дорогие услуги;
* Отсутствие надлежащего наблюдения при выполнении упражнений;
* Отсутствие возможности правильно оценить ваши таланты и недостатки;

## 2.2 Разработка веб-ориентированного мотивационного сервиса с элементами геймификации

Проблема мотивационных сервисов в настоящее время имеет особое значение. Большинство мотивационных сервисов имеют узкую направленность и отсутствие интереса к выполнению поставленных задач. Поэтому целью моего диплома было создание системы, в которой пользователь мог выполнять собственные цели и задачи, которые он сам перед собой поставил. Для поддержания интереса к сервису и собственным задачам пользователю необходимо поддерживать жизнедеятельность своего персонажа. В основе системы поддержания жизни и активности персонажа будет лежать собственная мотивация пользователя.

Необходимо разработать веб-сайт, в структуру которого будут входить:

• Регистрация, процесс сообщения сайту своих данные для получения доступа к полному функционалу сайта. Необходима для создания профиля, используемого для хранения заданий и персональных данных, а также информации о персонаже, его уровне, состояния здоровья, полученном опыте, инвентаре предметов и достижениях;

• Аутентификация, процесс проверки подлинности пользователя путем сравнения, введённого им логина и пароля с логином и паролем в сохраненной в базе данных;

• Интерфейс, для ввода пользователями данных о ежедневных делах и разовых задач, так же в интерфейсе расположена фотография персонажа, информация о уровне, состояние здоровья, количестве опыта, количество монет. Достижения – место для хранения наград за выполнения поставленных задач. Магазин для покупки разного вида пополнения здоровья за игровую валюту;

• База данных для хранения информации о пользователе, данных о персонаже: уровне, количестве здоровья, количестве опыта, количестве монет, достижениях и купленных предметах пополнения здоровья. Так же хранится информация о выполненных и невыполненных заданиях;

• Элемент геймификации. Используется для вовлечения, поддержания интереса пользователя к выполнению поставленных ежедневных дел и задач. За выполнение ежедневных дел и задач, персонаж пользователя получает опыт и монеты. При получение необходимого опыта, персонаж пользователя переходит на следующий уровень изменяя фотографию персонажа и восстанавливая полностью здоровье. За полученных монеты, пользователь имеет возможность купить предметы для восстановления здоровья, а также их использовать. При достижении определенных уровней, а также выполнению поставленных задач, пользователь получит награду во вкладе «Достижения».

Разрабатываемый сайт может найти практическое применение в различных областях человеческой жизни, так как сайт многофункционален, не имеет узкого направленного профиля. Сервис подходит для помощи пользователю в достижении его целей. Проект позволяет следовать пользователям поставленной задаче в разных сферах деятельности.

# 3 ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ

## 3.1 Разработка клиентской части приложения

React – открытая JavaScript библиотека для создания интерфейсов, которая призвана решать проблемы частичного обновления содержания веб-страницы, с которыми сталкиваются в разработке одностраничных приложений. React позволяет разработчикам создавать большие веб-приложения, которые используют данные, которые меняются со временем, без перезагрузки страницы. Его цель состоит в том, чтобы быть быстрым, простым, масштабируемым. React обрабатывает только интерфейс в приложениях. Поэтому для разработки клиентской части программного продукта была выбрана данная библиотека.

Redux - открытая JavaScript библиотека предназначена для управления состоянием программы. Чаще всего используется вместе с React или Angular для построения пользовательских интерфейсов. Состояние всей программы сохраняется в дереве объектов в одном хранилище. Одно дерево состояний облегчает налаживание или проверку программы; это также позволяет сохранять состояние вашего приложения в процессе разработки, для ускорения цикла разработки. Редукторы - это только чистые функции, которые принимают прежнее состояние и действие, и возвращают следующее состояние. В процессе разработки редукторы могут быть разделены на более мелкие, которые управляют определенными частями дерева состояний. Поскольку редукторы - это только функции, вы можете контролировать порядок их передачи, передавать дополнительные данные или даже создавать повторяющиеся редукторы для обычных задач, таких как разбиение на страницы.

Основное преимущество React и Redux заключается в их сравнительной простоте и в том, что они направлены на решение одной задачи, на разработку интерфейсов. Если задаться целью найти нечто, что делает что-то одно, но делает это хорошо, можно сказать, что обе библиотеки отлично делают то, чего от них ждут. В то время, как для кого-то подход, связанный с использованием контейнера состояния может показаться незнакомым, большинство разработчиков могут легко разобраться в этой концепции и понять преимущества архитектуры, основанной на однонаправленном потоке данных, и то, как такой подход может упростить приложения со сложными пользовательскими интерфейсами.

### 3.1.1 Обоснование выбора среды разработки

Для тестирования работоспособности сайта было решено использовать виртуальный сервер. Ниже для сравнения приведены программы для создания виртуального сервера:

1. XAMPP — кроссплатформенная сборка веб-сервера, содержащая Apache, MySQL, интерпретатор скриптов PHP, язык программирования Perl и большое количество дополнительных библиотек, позволяющих запустить полноценный веб-сервер .

Полный пакет XAMPPсодержит следующие компоненты:

* Web-сервер Apache с поддержкой SSL;
* СУБД MySQL;
* PHP;
* Perl;
* FTP-сервер FileZilla;
* POP3/SMTP сервер;
* утилиту phpMyAdmin.

XAMPP работает со всеми 32-х разрядными ОС Microsoft (98/2000/XP/2003/Vista/7/10), а также с Linux, Mac OS X и Solaris. Программа свободно распространяется согласно лицензии GNU GeneralPublicLicense и способна обслуживать динамические страницы.

Компоненты программы не требуют настройки. Программа регулярно обновляется для включения в состав новейших версий Apache/MySQL/PHP и Perl. Также в составе XAMPP присутствуют другие модули, включая OpenSSL и phpMyAdmin.

Программа имеет простой и понятный пользовательский интерфейс. Установка XAMPP занимает меньше времени, чем установка каждого компонента в отдельности.

Изначально XAMPP создавался как инструмент для разработчиков, позволяя веб-дизайнерам и программистам тестировать свою работу, не используя Интернет. Для упрощения работы некоторые возможности и настройки безопасности отключены по умолчанию, и в целом XAMPP рекомендуется к использованию только в очень дружественном окружении. Однако XAMPP иногда используется и во всемирной паутине. Также программа поддерживает создание и управление базами данных MySQL и SQLite.

2. Denwer — набор дистрибутивов и программная оболочка, предназначенные для создания и отладки сайтов на локальном ПК под управлением ОС Windows .

Сразу после установки доступен полностью работающий веб-сервер Apache, работающий на локальном компьютере, на котором может работать неограниченное количество сайтов, что очень эффективно для разработки и отладки сценариев PHP без загрузки его файлов на удаленный сервер. Для запуска практически всех утилит «Денвера» используется приложение в подкаталоге корневого каталога установки «Денвера». При запуске создается виртуальный диск, где хранятся все файлы проектов.

Базовый пакет:

* Веб-сервер Apache с поддержкой SSI, SSL, mod\_rewrite, mod\_php;
* Интерпретатор PHP с поддержкой GD, MySQL, SQLite;
* СУБД MySQL с поддержкой транзакций (mysqld-max);
* Система управления виртуальными хостами, основанная на шаблонах;
* Система управления запуском и завершением;
* Панель phpMyAdmin для администрирования СУБД;
* Ядро интерпретатора Perl без стандартных библиотек (поставляются отдельно);
* Эмулятор sendmail и сервера SMTP с поддержкой работы совместно с PHP, Perl, Parser и др.

Особенностью, отличающей Denwer от других WAMP-дистрибутивов, является автоматическая правка системного файла hosts, являющегося локальным аналогом DNS-сервера, что позволяет обращаться к локальным сайтам, работающим под управлением Денвера, по именам, совпадающим с именем папки.

Таблица 1 - Сравнение XAMPP и Denwer

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Protable | Русский язык | Наличие GUI | Полезный GUI | Просмотр логов |
| XAMPP | + | - | + | + | + |
| Denwer | + | + | - | - | - |

На основе проведённых исследований (см. табл. 1) было принято решение в качестве виртуального сервера использовать программу XAMPP так как она обладает всем необходимым функционалом и имеет удобный пользовательский интерфейс для работы.

### 3.1.2 Обоснование выбора языка разработки

PHP известен, как серверный язык программирования. Это означает, что он работает на веб сервере. Большинство языков веб-программирования являются серверными языками, но некоторые, например, JavaScript, работают на стороне клиента, это означает, что они работают в веб-браузере. Серверные языки дают вам больше гибкости, так как вы можете то, что трудно осуществить с помощью JavaScript — например, работа с файлами, базами данных, или работа с изображениями.

Выполнение кода со стороны сервера является более безопасным способом, чем на стороне клиента, как это делает JavaScript. Поскольку код JavaScript отправляется в веб-браузер, для посетителей сайта легко его просмотреть и редактировать. Даже на одной странице сайта можно с легкостью совмещать PHP и JavaScript. Код находящийся на стороне сервера остаётся веб-сервере и недоступен для посетителей сайта. PHP это инструмент, который находится на веб-сервере и там выполняет PHP скрипты.

PHP представляет собой программное обеспечение с открытым исходным кодом. Это означает, что любой пользователь может получить доступ и работать на PHP. Это помогает быть уверенным, что PHP будет работать в течение длительного времени. PHP можно свободно скачать и использовать. Это является причиной того, что многие хостинг-провайдеры широко используют PHP. Вы обнаружите, что подавляющее большинство веб-хостингом поддерживают работу PHP.

В то время как многие языки программирования могут быть использованы для создания веб-приложений, PHP является одним из языков специально разработаных для использования в Интернете. PHP имеет множество полезных веб-функций таких, как:

* Считывание и обработка веб-форм и куки-файлов;
* Функции создания и работы с графикой;
* Установка связи с популярными базами данных такими, как MySQL;
* Функции для работы с HTML.

Одной из замечательных функций PHP является то, что вы можете включить блоки PHP кода в HTML-страницы. Вы можете обособить PHP блоки с помощью специальных символов. Когда веб-сервер получает информацию о странице,все PHP блоки запускаются движком PHP, в то время как, другие части страницы отправляются «как есть» в браузер.Эта функция позволяет легко делать интерактивными обычные веб-страницы. Отличный инструмент для форм с обратной связью и форм с похожими функциями.

Вы можете использовать PHP для записи практически любого типа веб-приложения или сценария. Обычные PHP приложения, включают в себя:

* Программное обеспечение для ведения блогов таких, как WordPress;
* Системы электронной коммерции Magento;
* Система управления контентом, в том числе Drupal и Joomla;
* Программное обеспечение для ведения форумов, например, phpBB.

Кроме того, многие популярные веб-сайты, в том числе Yahoo!, Википедия, Facebook, Digg разработаны с использованием PHP. Также с помощью языка PHP возможно написание объектно-ориентированного кода.

### 3.1.3 Обоснование выбора СУБД

Для хранения информации о фенологических явлениях требуется наличие СУБД. Ниже для сравнения приведены поставщики СУБД:

1. Oracle.

Первым «настоящим» выпуском СУРБД Oracle стал Oracle 2. Система была написана на ассемблере и поддерживала только базовые свойства SQL. Все последующие 11 лет Oracle занималась тем, что выпускала обновления для этого флагманского продукта.

Вероятно, одной из причин, почему СУРБД Oracle так долго удерживала лидерство среди мощных корпоративных СУРБД, было то, что все ее апдейты оказались тесно связанными с удовлетворением насущных потребностей рынка.

Все эти жаргонизмы мира баз данных, такие как «масштабируемая», «распределенная», «программируемая», «переносимая», также тесно связаны с линейкой продуктов Oracle. Например, в 1985 году, в связи с возросшей ролью сетевых коммуникаций, в СУРБД Oracle была добавлена поддержка клиент-серверной модели, а с наступлением Эры Интернет, продукты Oracle стали включать в себя собственную виртуальную машину Java (JVM).

Особенности

В результате релиз первой в мире облачной базы данных OracleDatabase 12c (2013 г), обладал уже следующими особенностями:

* Улучшенный алгоритм защиты чувствительных данных;
* Улучшенное сжатие данных;
* Высокая плотность консолидации данных;
* Максимальная доступность;
* Автоматическая оптимизация данных;
* Средства быстрой разработки веб-приложений с помощью SQL и/или PL/SQL;
* Улучшенная сетевая производительность.

Среди уникальных особенностей – поддержка подключаемых баз данных и живая миграция [9].

2. MySQL.

Огромным преимуществом MySQL, сравнительно с проприетарными продуктами, является открытость ее исходного кода. Согласно маркетинговым отчетам, на сегодняшний день существует более 10 миллионов инсталляций MySQL, что является показателем ее стремительного вхождения в корпоративный сектор.

Особенности

Такие системы, как Oracle и SQL Server, являются крупными корпоративными СУРБД, в то время как MySQL предназначена для поддержки малых и средних проектов. Обновления MySQL выходят практически каждый год [9].

Поворотным для MySQL оказался 2010 год, когда увидела свет MySQL 5.5, обладающая рядом значительных улучшений сравнительно с предыдущими выпусками, а именно:

* Наличие движка InnoDB;
* Полусинхронная репликация;
* Улучшенный механизм секционирования данных;
* Новая система блокировок;
* Оптимизация под многоядерные процессоры.

Таблица 2 - Сравнительная таблица общих характеристик

|  |  |  |
| --- | --- | --- |
| Характеристика | MySQL | Oracle |
| Интерфейс | SQL | GUI, SQL |
| Поддержка языков | C, C++, Java, Ruby, Objetive C идр. | С, С++, Java, Ruby, Objective С и  др. |
| Операционная система | Windows, Linux, OS X, Free BSD, Solaris | Windows, Linux, Solaris, HP-UX, OS X, z/OS, AIX |
| Лицензия | Свободная | Проприетарная |

На основе проведённых исследований (см. табл. 2) было принято решение в качестве СУБД использовать MySQL так как она предназначена для поддержки малых и крупных проектов и имеет свободную лицензию.

## 3.2 Разработка движка сайта

Для разработки движка сайта использовался язык PHP. Ниже приведен код обеспечивающий работаспособность сайта, корректность выполнения функций, и описание к нему.

Подключение к базе данных.

<?php

$db = mysql\_connect("localhost","ash","qwerty") ordie ("НевозможноподключениекMySQL".mysql\_error());

mysql\_select\_db("phenology", $db);

mysql\_query("SET names cp1251");

?>

Данный код обеспечивает подключение к локальному серверу, указывает базу данных для дальнейшей работы и устанавливает кодировку для корректного отображения шрифтов.

Функция mysql\_connect(host, login, password) обеспечивает подключение к хостингу. Она принимает следующие пораметры:

1. Имя хостинга;
2. Логин пользователя;
3. Пароль пользователя.

Функция mysql\_error() сообщает об ошибке возникшей в случае неудачного подключения.

Функция mysql\_select\_db(stringdatabase\_name [, resourcelink\_identifier]) выбрать базу данных для дальнейшей работы с ней. Она принимает следующие параметры:

1. Имя базы данных;
2. Ссылка на подключение к базе данных.

Функция mysql\_query() в данном случае устанавливает кодировку для отображения шрифтов.

Регистрация пользователей.

<?php

if(isset($\_POST['confirm'])){

$login = $\_REQUEST['login'];

$password = $\_REQUEST['password'];

$mail = $\_REQUEST['mail'];

$sex = $\_REQUEST['sex'];

$country = $\_REQUEST['country'];

$town = $\_REQUEST['town'];

$password = md5($password);

require 'connect.php';

$sql = mysql\_query("INSERT INTO users (login, password, mail, sex, country, town) VALUES('{$login}', '{$password}', '{$mail}', '{$sex}', '{$country}', '{$town}')");

setcookie("login",$login,time()+2592000,"/");

header("Location: http://127.0.0.1/fenology/index.php");

exit;

}

?>

Данный код обеспечивает добавление в базу данных сайта информацию о новых пользователях.

Функция isset()определяет была ли нажата кнопка подтверждения регистрации.

С помощью функции $\_REQUEST['имя поля'] в переменные записываются значения из полей находящихся на странице чьи имена записываются в квадратных скобках.

Функция md5() хеширует пароль.

Функция require позволяет включать файлы в PHPсценарий. В данном случае включается скрипт для подключения к базе данных сайта.

Функция mysql\_query() в данном случае посылает запрос в базу данных о вставке новых значений в определённую таблицу с указание полей в которые необходимо записать новые значения и переменных из которых значения будут записаны в таблицу.

Функция setcookie() создает новый cookie при регистрации. Она принимает следующие значения:

1. Имя нового cookie;
2. Значение, которое данное cookie будет хранить;
3. Время, которое будет существовать данное cookie
4. Область его действия.

Функция header() перенаправляет на главную страницу сайта.

Функция exit завершает выполнение скрипта во избежание ошибок и повторной отправки данных.

Авторизациячерезформу.

<?php

if(isset($\_POST['enter'])){

require 'connect.php';

$log = $\_REQUEST['login'];

$pas = $\_REQUEST['password'];

$pas = md5($password);

$result = mysql\_query("SELECT \* FROM users WHERE login = '$log' AND password = '$pas'");

$row = mysql\_fetch\_array($result);

if($row){

setcookie("login",$log,time()+2592000,"/");

}

header("Location: http://127.0.0.1/fenology/index.php");

}

?>

Функция md5() хеширует пароль.

Функция isset()определяет была ли нажата кнопка запроса на авторизацию.

Функция require позволяет включать файлы в PHPсценарий. В данном случае включается скрипт для подключения к базе данных сайта.

С помощью функции $\_REQUEST['имя поля'] в переменные записываются значения из полей находящихся на странице чьи имена записываются в квадратных скобках.

Функция mysql\_query() в данном случае выбирает из базы данных пользователей все поля удовлетворяющие определённым условиям.

Функция mysql\_fetch\_array() расширенная версия функции mysql\_fetch\_row(). В дополнении к хранению значений в массиве с численными индексами, функция возвращает значения в массиве с индексами по названию колонок.

Функция setcookie() создает куки для автоматической авторизации при захлде на сайт.

Функция header() возвращает пользователя на главную страницу.

Авторизациячерезcookie.

<?php

if(isset($\_COOKIE['login'])){

$cookiename = $\_COOKIE['login'];

}

?>

Функция isset() проверяет существует ли cookieс определённым именем. Если cookieс таки именем существует, то переменную $cookiename присваивается значение этого cookieпри заходе на сайт для дальнейшего использования.

Благодаря данной проверке авторизованным пользователям отображается форма приветствия вместо формы авторизации и становится доступна форма для добавления фенологических явлений.

Проверка авторизованного пользователя проходит следующим образом:

if(isset($\_COOKIE['login'])){

echo"

<form id='reg' method='post' action=''>

<table>

<tr>

<td>Приветствуем</td>

<td>".$cookiename."</td>

</tr>

<tr>

<td></td>

<td><input type='submit' value='Выход' name='logout' /></td>

</tr>

</table>

</form>

";

}else{

echo"

<form id='reg' method='post' action=''>

<table>

<tr>

<td>Логин</td>

<td><input class='fields' type='text' name='login' placeholder='Логин' required /></td>

</tr>

<tr>

<td>Пароль</td>

<td><input class='fields' type='password' name='password' placeholder='Пароль' required /></td>

</tr>

<tr>

<td><input type='submit' value='Вход' name='enter'/></td>

<td><a href='registration.php'>Регистрация</a></td>

</tr>

</table>

</form>";

}

Если cookieс определённым именем не существует, то при заходе на сайт на странице показывается форма авторизации (см. рис. 1).

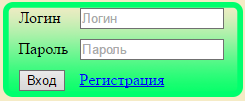


Рисунок 1 - Форма авторизации

Если cookieс определённым именем не существует, то при заходе на сайт на странице показывается форма приветствия (см. рис.2)

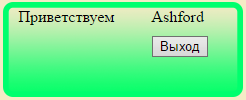


Рисунок 2 –Приветственная форма

Аналогичным образом происходит отображение формы для добавления фенологических данных. Так же при добавлении отзывов логин будет подставлен автоматически.

Разлогинивание на сайте.

if(isset($\_POST['logout'])){

setcookie("login","",time() - 2592000,"/");

header("Location: http://127.0.0.1/fenology/index.php");

exit;

}

Для того чтобы разлогиниться на сайте необходимо нажать кнопку «выход» на форме приветствия (см. рис. 2). После того ка пользователь разлогинится, то он не сможет добавлять фенологические явления на сайте.

Функция isset() проверяет нажата ли на форме кнопка с определённым именем.

Функция setcookie() в данном случае удаляет существующий cookieв котором хранится логин под которым пользователь был авторизован.

Функция header() возвращает на страницу на которой пользователь разлогинился.

Функция exit завершает выполнение скрипта.

Добавление фенологических явлений.

if(isset($\_POST['post'])){

$type = $\_REQUEST['type'];

$phenomen = $\_REQUEST['phenomen'];

$latitude = $\_REQUEST['latitude'];

$longitude = $\_REQUEST['longitude'];

$date = $\_REQUEST['date'];

require 'connect.php';

mysql\_query("INSERT INTO phenomens (type, phenomen, latitude, longitude, date, user) VALUES('{$type}', '{$phenomen}', '{$latitude}', '{$longitude}', '{$date}', '{$cookiename}')");

header("Location: http://127.0.0.1/fenology/phenomens.php");

exit;

}

Данный скрипт позволяет добавлять в базу данных информацию о фенологических наблюдениях.

Функция isset()проверяет нажата ли кнопка «Добавить запись»

$\_REQUEST заполняет переменные значением из соответствующих полей.

Функция require позволяет включать файлы в PHP сценарий. В данном случае включается скрипт для подключения к базе данных сайта.

Функция mysql\_query() посылает запрос в базу данных на вставку в таблицу феноменов с указанием полей в которые необходимо передать значения и полей и переменных которые передают значения.

Функция header() перенаправляет пользователя обратно на страницу добавления

Функция exitзавершает выполнение скрипта во избежание повторной отправки данных.

Добавление отзывов происходит аналогичным образом.

Вывод информации из базы данных

require 'connect.php';

$result = mysql\_query("SELECT \* FROM phenomens");

if(mysql\_num\_rows($result)>0){

echo "<table id='names'>";

while($data = mysql\_fetch\_array($result))

{

echo '<tr>';

echo '<td class="as1">'.$data['type']."</td>";

echo '<td class="as2">'.$data['phenomen']."</td>";

echo '<td class="as3">'.$data['place']."</td>";

echo '<td class="as4">'.$data['date']."</td>";

echo '<td>'.$data['user']."</td>";

echo '</tr>';

}

echo "</table>";

}

Данный скрипт позволяет отображать информацию из базы данных о фенологических явления. При отображении на странице создается таблица в ячейки которой записывается значение полей из базы данных.

Отображение комментариев происходит аналогичным образом.

На сайте имеется яндекс-карта для отображения мест наблюдения фенологических явлений. Для её подключения к сайту используется следующий код:

<scriptsrc="//api-maps.yandex.ru/2.0/?load=package.standard&lang=ru-RU" type="text/javascript"></script>

ДлявыводакартынасайтиспользуетсяJavaScript:

<scripttype="text/javascript">

ymaps.ready(init);

varmyMap;

varsetted = false;

function init ()

{

myMap = new ymaps.Map("map\_canvas", {center: [56.633561, 47.884646],zoom: 11});

myMap.controls

.add('zoomControl')

.add('typeSelector')

.add('searchControl');

}

</script>

## 3.3 Разработкабазыданных

### 3.3.1 Реляционная модель данных

Разработана реляционная модель базы данных (см. рис. 3)

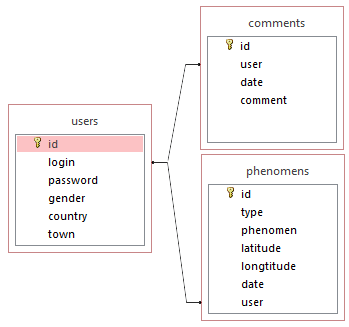


Рисунок 3 - Реляционная модель данных

### 3.3.2 Таблицаusers

SQL– запрос на создание таблицы пользователей:

CREATETABLEusers (idINT(11) NOTNULLAUTO\_INCREMENTFIRST, loginCHAR(40) NOT NULL Primary key, password CHAR(40) NOT NULL, gender CHAR(15) NOT NULL, country CHAR(30) NOT NULL, town CHAR(30) NOT NULL)

Таблица 3 - Таблица пользователей

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Поле | Тип | Длинна | Ключевое поле | Ограничения |
| 1 | id | Int | 11 | Да | Счетчик |
| 2 | login | Char | 40 | Нет | Не может быть пустым |
| 3 | password | Date | 32 | Нет | Не может быть пустым |
| 4 | gender | Text | 15 | Нет | Не может быть пустым |
| 5 | country | Char | 30 | Нет | Не может быть пустым |
| 6 | town | Char | 30 | Нет | Не может быть пустым |

### 3.3.3 Таблицаphenomens

SQL– запрос на создание таблицы феноменов:

CREATE TABLE phenomens (id INT(11) NOT NULL AUTO\_INCREMENT FIRST, type CHAR(40) NOT NULL, phenomen CHAR(40) NOT NULL, latitude CHAR(10) NOT NULL,latitude CHAR(10) NOT NULL,dateDATE NOT NULL, user CHAR(40) NOT NULL)

Таблица4 - Таблица фенологических явлений

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Поле | Тип | Длинна | Ключевое поле | Ограничения |
| 1 | id | Int | 11 | Да | Счетчик |
| 2 | type | Char | 40 | Нет | Не может быть пустым |
| 3 | phenomen | Char | 40 | Нет | Не может быть пустым |
| 4 | latitude | Char | 10 | Нет | Не может быть пустым |
| 5 | longitude | Char | 10 | Нет | Не может быть пустым |
| 6 | date | Char | 10 | Нет | Не может быть пустым |
| 7 | user | Char | 40 | Нет | Заполняется из cookie |

### 3.3.4 Таблица comments

SQL– запрос на создание таблицы комментариев:

CREATE TABLE comments (id INT(11) NOT NULL AUTO\_INCREMENT FIRST, user CHAR(40) NOT NULL, date CHAR(40) NOT NULL, comment CHAR(15) NOT NULL)

Таблица5 - Таблица комментариев

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Поле | Тип | Длинна | Ключевое поле | Ограничения |
| 1 | id | Int | 11 | Да | Счетчик |
| 2 | user | Char | 40 | Нет | Заполняется из cookie |
| 3 | date | Date | 10 | Нет | Не может быть пустым |
| 4 | comment | Text | 500 | Нет | Не может быть пустым |

## 3.4 Разработка интерфейсов пользователя

Интерфейс пользователя, он же пользовательский интерфейс (UI — англ. userinterface) — разновидность интерфейсов, в котором одна сторона представлена человеком (пользователем), другая — машиной/устройством. Представляет собой совокупность средств и методов, при помощи которых пользователь взаимодействует с различными, чаще всего сложными, машинами, устройствами и аппаратурой.

Для удобства пользователей на сайте были разработаны следующие интерфейсы:

1. Меню сайта.

Для перехода между страницами в шапке сайта располагается меню (см. рис. 4). В нем находятся ссылки на следующие страницы:

1. Титульная страница сайта;
2. Фотогалерея сайта;
3. Страница для добавления и просмотра фенологических данных;
4. Страница с отзывами пользователей;
5. Страница с контактами сайта.

Так как данное меню находится в шапке сайта, то на всех страницах оно выглядит одинаково.

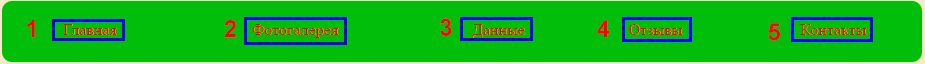


Рисунок 4 - Меню сайта

2. Форма авторизации.

На сайте имеется форма авторизации (см. рис. 5), благодаря данной форме зарегистрированные пользователи могут авторизовываться на сайте. Данная форма содержит следующие компоненты:

1. Поле для ввода логина;
2. Поле для ввода пароля;
3. Кнопка для подтверждения авторизации;
4. Ссылку на страницу регистрации в которой пользователи могут зарегистрироваться, если их нет в системе.

Данная форма располагается на следующих страницах:

* Титульная страница;
* Страница для добавления фенологических явлений;
* Страница отзывов.

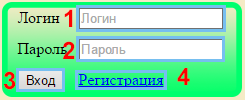


Рисунок 5 - Форма авторизации

3. Форма приветствия.

Для пользователей, авторизованных на сайте вместо формы авторизации показывается форма приветствия (см. рис. 6). Данная форма отображает:

1. Ник авторизованного пользователя;
2. Кнопку разлогинивания.

Форма приветствия располагается на тех же страницах что и форма авторизации.

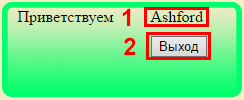


Рисунок 6 - Форма приветствия

4. Форма регистрации.

Форма регистрации (см. рис. 7) предназначена для регистрации новых пользователей. Попасть на данную форму можно нажав на ссылку «Регистрация» в форме авторизации (см. рис. 5). Данная форма содержит следующие компоненты:

1. Поле для ввода логина;
2. Поле для ввода пароля;
3. Селектор для выбора пола пользователя;
4. Поле для указания страны в которой проживает пользователь;
5. Поле для указания города в котором проживает пользователь;
6. Кнопка для подтверждения регистрации.

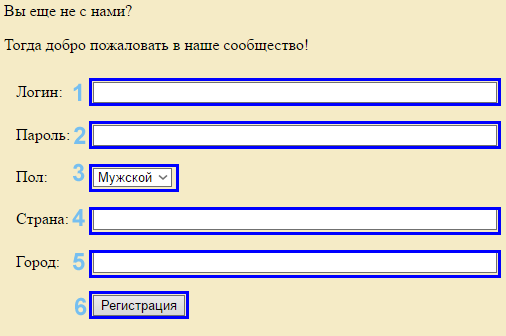


Рисунок 7 - Форма регистрации пользователей

5. Форма для добавления фенологических явлений.

Форма для добавления фенологических явлений (см. рис. 8) позволяет авторизованным пользователям сайта добавлять в базу данных информацию о фенологических явления. Не авторизованные пользователи данную форму не видят. Форма содержит следующие компоненты:

1. Селектор для выбора типа фенологических явлений;
2. Поле для указания наименования явления;
3. Поле для указания координат широты;
4. Поле для указания координат долготы;
5. Поле для ввода даты наблюдения явления;
6. Кнопка для подтверждения добавления.

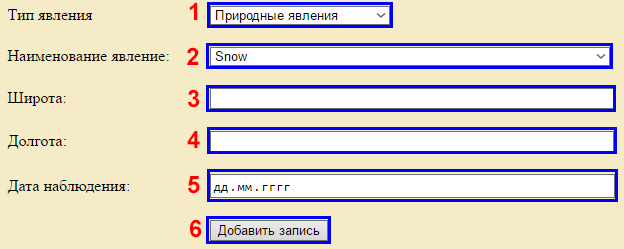


Рисунок 8 - Форма для добавления фенологических явлений

Поля для записи широты и долготы изменяются путем указания точки на карте.

6. Форма карты.

Карта предназначена для указания и отображения фенологических явлений. Данная форма содержит следующие компоненты:

1. Поле для выбора типа отображаемых явлений;
2. Кнопка для отображения маркеров;
3. Кнопка для удаления маркеров;
4. Яндекс карта;
5. Слайдер для изменения масштаба карты;
6. Метка;
7. Окно отображающее информацию о явлении на крте.



Рисунок 9 - Карта

6. Форма для отзывов.

Форма для отзывов (см. рис.10) предназначена для того что бы пользователи могли оставлять отзывы о работе сайта, его оформлении, или пожелания касательно модернизации сайта. Данная форма содержит в себе следующие компоненты:

1. Ник пользователя;
2. Поле для ввода текста сообщения;
3. Кнопка для отправки отзыва.

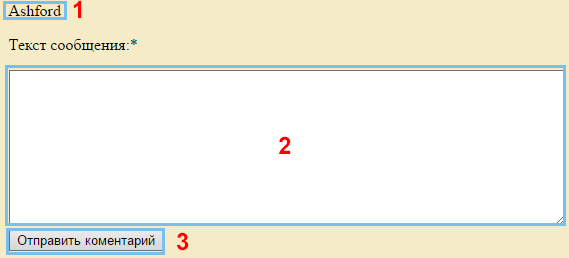


Рисунок 10– Форма для добавления отзывов

## 3.5 Тестирование интерфейса методом GOMS

GOMS это семейство методов, позволяющих провести моделирование выполнения той или иной задачи пользователем и на основе такой модели оценить качество интерфейса (точнее говоря оценить время выполнения задачи как основной критерий качества). GOMS это сокращение от английского Goals, Operators, Methods, and Selection Rules – Цели, Операторы, Методы и Правила выбора. Данный способ был предложен S. K. Card, T. P. Moran и A. Newell в 1983 году.

Идея метода заключается в том, что все действия пользователя можно представить, как набор типовых составляющих. Для этих типовых составляющих можно провести измерения времени их выполнения и получить статистические оценки времени выполнения того или иного элементарного действия. Оценка качества интерфейса заключается в разложении выполняемой задачи на типовые составляющие, и вычислении времени, которое будет в среднем затрачиваться пользователем на выполнение этой задачи. Обычно тот интерфейс лучше, при котором время выполнения задачи меньше [14].

Анализ навигации с помощью меню сайта (см. табл. 6).

Таблица 6 - Анализ навигации с помощью меню

|  |  |  |  |
| --- | --- | --- | --- |
| № | Тип | Время | Комментарий |
| 1 | M | 1.2 сек. | Принятие решения |
|  | P | 1.1 сек. | Наведение курсора мыши на ссылку |
|  | BB | 0.1 сек. | Щелчок по левой кнопке мыши |
|  | W | 0.1 сек. | Ожидание открытия страницы |
| Сумма | | 1.5 сек. | |

Анализ авторизации пользователе (см. табл. 7).

Таблица 7 - Анализ авторизации пользователей

|  |  |  |  |
| --- | --- | --- | --- |
| № | Тип | Время | Комментарий |
| 1 | P | 1.1 сек. | Наведение указателя маши на поле для ввода логина |
| 2 | BВ | 0.1 сек. | Щелчок по левой кнопке мыши |
| 3 | H | 0.4 сек. | Перемещение руки с мыши |
| 4 | T(10) | 2.8 сек | Ввод логина |
| 5 | H | 0.4 сек. | Перемещение руки на мышь |
| 6 | P | 1.1 сек. | Наведение указателя мыши на поле для ввода пароля |
| 7 | BВ | 0.1 сек. | Щелчок по левой кнопке мыши |
| 8 | H | 0.4 сек. | Перемещение руки с мыши |
| 9 | T(10) | 2.8 сек. | Ввод пароля |
| 10 | H | 0.4 сек. | Перемещение руки на мышь |
| 11 | P | 1.1 сек. | Наведение указателя мыши на копку подтверждения |

Продолжение таблицы 7

|  |  |  |  |
| --- | --- | --- | --- |
| № | Тип | Время | Комментарий |
| 12 | BВ | 0.1 сек. | Щелчок по левой кнопке мыши |
| 13 | W | 0.1сек. | Ожидание ответа от сервера |
| Сумма | | 10.9 сек. | |

Анализ времени на регистрацию пользователя (см. табл. 8)

Таблица 8 - Анализ времени на регистрацию

|  |  |  |  |
| --- | --- | --- | --- |
| № | Тип | Время | Комментарий |
| 1 | P | 1.1 сек. | Наведение указателя маши на поле для ввода логина |
| 2 | BВ | 0.1 сек. | Щелчок для захода в поле |
| 3 | H | 0.4 сек. | Перемещение руки с мыши |
| 4 | T(10) | 2.8 сек | Ввод логина |
| 5 | H | 0.4 сек. | Перемещение руки на мышь |
| 6 | P | 1.1 сек. | Наведение указателя мыши на поле для ввода пароля |
| 7 | BВ | 0.1 сек. | Щелчок для захода в поле |
| 8 | H | 0.4 сек. | Перемещение руки с мыши |
| 9 | T(10) | 2.8 сек. | Ввод пароля |
| 10 | H | 0.4 сек. | Перемещение руки на мышь |
| 11 | P | 1.1 сек. | Наведение указателя мыши на селектор пола |
| 12 | BВ | 0.1 сек. | Щелчок по левой кнопке мыши для открытия списка |
| 13 | P | 1.1сек. | Выбор половой принадлежности |
| 14 | BB | 0.1 сек. | Щелчок для подтверждения выбора |
| 15 | P | 1.1 сек. | Наведение на поле для указания страны |
| 16 | BB | 0.1 сек | Щелчок по полю для ввода |
| 17 | H | 0.4 сек. | Перемещение руки с мыши |
| 18 | T(20) | 5.6 сек. | Ввод названия страны |
| 19 | H | 0.4 сек. | Перемещение руки на мышь |
| 20 | P | 1.1 сек. | Наведение курсора на поле для указания города |
| 21 | BB | 0.1 сек. | Щелчок для захода в поле |
| 22 | H | 0.4 сек. | Перемещение руки с мыши |
| 23 | T(20) | 5.6 сек. | Ввод названия города |
| 24 | H | 0.4 сек | Перемещение руки на мышь |
| 25 | P | 1.1 сек. | Наведение курсора на кнопку подтверждения |
| 26 | BB | 0.1 сек. | Щелчок по кнопке подтверждения |
| 27 | W | 0.1 сек. | Ожидание ответа от сервера |
| Сумма | | 28.6 сек. | |

Анализ времени на добавление фенологического явления (см. табл. 9).

Таблица 9 - Анализ времени на добавление явлений

|  |  |  |  |
| --- | --- | --- | --- |
| № | Тип | Время | Комментарий |
| 1 | P | 1.1 сек. | Наведение указателя на поле для выбора типа явления |
| 2 | BB | 0.1 сек | Щелчок правой кнопкой мыши по полю |
| 3 | P | 1.1 сек. | Наведение указателя на нужную строку |
| 4 | BB | 0.1 сек. | Щелчок для выбора типа |
| 5 | P | 1.1 сек | Наведение указателя на поля для выбора явления |
| 6 | BB | 0.1 сек. | Щелчок левой кнопкой по полю |
| 7 | P | 1.1 сек. | Наведение указателя на нужную строку |
| 8 | BB | 0.1 сек | Щелчок для выбора явления |
| 9 | P | 1.1 сек | Наведение указателя на карту |
| 10 | M | 2 сек. | Выбор места на карте |
| 11 | BB | 0.1 сек. | Щелчок по выбранному месту |
| 12 | P | 1.1 сек. | Наведение указателя на поле для выбора доты |
| 13 | BB | 0.1 сек. | Щелчок левой кнопкой мыши по полю для открытия календаря |
| 14 | M | 2 сек. | Выбор нужной даты |
| 15 | P | 1.1 сек. | Наведение курсора на кнопку подтверждения |
| 16 | BB | 0.1 сек. | Щелчок по левой кнопке мыши |
| 17 | W | 0.1 сек. | Ожидание ответа от сервера |
| Сумма | | 12.5 сек. | |

Анализ времени на выбор явлений для отображения на карте (см. табл. 10).

Таблица 10 - Анализ времени для вывода меток на карту

|  |  |  |  |
| --- | --- | --- | --- |
| № | Тип | Время | Комментарий |
| 1 | P | 1.1 сек. | Наведение указателя на поле для выбора типа явления |
| 2 | BB | 0.1 сек. | Щелчок левой кнопкой мыши |
| 3 | P | 1.1 сек. | Наведение на нужный тип явления |
| 4 | BB | 0.1 сек. | Щелчок левой кнопкой мыши |
| 5 | P | 1.1 сек. | Наведение на кнопку подтверждения |
| 6 | BB | 0.1 сек. | Щелчок левой кнопкой мыши |
| 7 | W | 0.1 сек | Ожидание ответа сервера |
| Сумма | | 3.7 сек. | |

Анализ времени на написание комментария (см. табл. 11).

Таблица 11 - Анализ времени на добавление комментария

|  |  |  |  |
| --- | --- | --- | --- |
| № | Тип | Время | Комментарий |
| 1 | P | 1.1 сек. | Наведение указателя на поле для ввода комментария |
|  | BB | 0.1 сек. | Щелчок левой кнопкой мыши |
|  | T(100) | 28 сек. | Ввод сообщения |
|  | P | 1.1 сек. | Наведение курсора на кнопку подтверждения |
|  | BB | 0.1 сек. | Щелчок левой кнопкой мыши |
|  | W | 0.1 сек | Ожидание ответа от сервера |
| Сумма | | 30.5 сек. | |

# ЗАКЛЮЧЕНИЕ

Сегодня с помощью спутников ученые могут наблюдать обширные пространства Земли с суровыми условиями, такие как арктические тундры и смещение границ криолитозоны. Собранные данные могут определить решения, связанные с такими явлениями, как пожары, наводнения, а также оптимальные сроки борьбы с сельскохозяйственными вредителями, возбудителями болезней, в том числе аллергий.

В соответствии с техническим заданием в данной дипломной работе произведена разработка системы сбора фенологических данных на основе веб-технологий.

В теоретической части были произведены исследования актуальности данной работы и состояние фенологии в России.

В аналитической части был произведен обзор современных систем сбора фенологических данных и их недостатков, а также анализ разработки системы сбора на основе современных веб-технологий.

В исследовательской части был произведён выбор средств для разработки системы и обоснование выбора, выбран язык для написания движка сайта, описано создание базы данных для сбора и хранения информации, проведена разработка пользовательского интерфейса сайта и тестирование.

# СПИСОК ЛИТЕРАТУРЫ

1. <https://ru.wikipedia.org/wiki/Фенология>
2. <http://dic.academic.ru/dic.nsf/dic_biology/6047/ФЕНОЛОГИЯ>
3. <http://biofile.ru/bio/15527.html>
4. http://webcodingsolutions.blogspot.ru/p/yandex-maps-api.html
5. http://www.sir35.ru/Fenologiya-v-Rossii-vchera-i-segodnya.html
6. <http://wladm.narod.ru/C_Sharp/webbegin.html>
7. <https://ru.wikipedia.org/wiki/Реляционная_модель_данных>
8. <https://ru.wikipedia.org/wiki/XAMPP>
9. <https://ru.wikipedia.org/wiki/Денвер_(программа)>
10. <http://www.databases.com.ua/article/populyarnyye-subd>
11. <http://php.net/manual/ru/language.basic-syntax.phpmode.php>
12. <http://www.umade.ru/log/2004/07/use-php-in-html-pages/>
13. <http://works.doklad.ru/view/gJ3G3KD9Qbs.html>
14. <http://pandia.ru/text/77/497/6204.php>
15. <http://33testers.blogspot.ru/2012/02/goms.html>
16. <http://www.tigir.com/javascript_select.htm>
17. http://e-javascript.ru/select.php?pp=10

# ПРИЛОЖЕНИЕ А

Техническое задание

1.Техническое задание на программу по ГОСТ 19.201-78

1.1 Введение

1.1.1. Наименование программного продукта

Наименование – «Сайт для сбора фенологических данных»

1.1.2. Краткая характеристика области применения

Данный сайт предназначен для сбора фенологических данных.

1.2. Основание для разработки

1.2.1 Основание для проведения разработки

Основанием для проведения разработки данного программного продукта было дипломное проектирование.

1.3. Назначение разработки

1.3.1. Функциональное назначение

Функциональным назначением данного сайта является предоставление информации о происходящих фенологических событиях для наблюдения и дальнейшего изучения.

1.3.2. Эксплуатационное назначение

Данный сайт является общедоступным в сети интернет. Конечными пользователя данного сайта являются все пользователи сети интернет.

1.4. Требование к программе или программному изделию

1.4.1. Требования к функциональным характеристикам

1.4.1.1. Требования к составу выполняемых функций

Сайт должен обеспечивать возможность выполнения перечисленных ниже функций:

1. Регистрация пользователей.
2. Авторизация пользователей.
3. Добавление информации о фенологических явлениях в базу данных.
4. Отображение информации о фенологических явлениях с указанием места их наблюдения на карте.
5. Оставление комментариев на сайте.

1.4.3.Условия эксплуатации

1.4.3.1. Требования к численности и квалификации персонала

Минимальное количество персонала для обслуживания данного сайта составляет не менее 2 штатных единиц – системный администратор. Количество конечных пользователей сайта не ограничено.

Системный администратор должен иметь высшее профессиональное образование. В перечень задач, выполняемых системным администратором должно входить:

1. Поддержание работоспособности технических средств;
2. Поддержание работоспособности системных программных средств.

1.4.4. Требования к информационной и программной совместимости

1.4.4.1. Требования к исходным кодам и языкам программирования

Исходные коды должны быть реализованы на языках: HTML5, PHP, JavaScript. В качестве среды разработки программы должна быть использована среда MicrosoftVisualStudio 2015.

1.5. Стадии и этапы разработки

1.5.1. Стадии разработки

Разработка должна быть проведена в 3 стадии:

1. Разработка технического задания;
2. Рабочее проектирование;
3. Внедрение.

1.5.2. Этапы разработки

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии рабочего проектирования должны быть выполнены перечисленные ниже этапы работ:

1. Разработка макета сайта;
2. Разработка страниц сайта;
3. Разработка программной документации;
4. Тестирование сайта.

1.6. Порядок контроля и приемки

1.6.1. Виды испытаний

Приемно-сдаточные испытания сайта представляют собой проверку его работоспособности на разных браузерах в том числе и мобильных.

# ПРИЛОЖЕНИЕ Б

Руководство программиста

Ввод информации в базу данных

Для ввода информации необходимо установить соединение с базой данных. За это отвечает следующий скрипт:

<?php

$db = mysql\_connect("localhost","ash","qwerty") ordie ("НевозможноподключениекMySQL".mysql\_error());

mysql\_select\_db("phenology", $db);

mysql\_query("SET names cp1251");

?>

Функция mesql\_connectпредназначена для подключения к серверу. Далее следует функция для выбора базы данных с которой предстоит работать. Функция mysql\_querryустанавливает кодировку для корректного отображения символов.

Для получения значений из полей на странице сайта используется функция $\_REQUEST[]. После получения значений всех необходимых полей и запись их в переменные с помощью функции mysql\_query() посылается sql запрос в базу данных на вставку:

INSERTINTOназвание таблицы (списокполей для заполнения) VALUES(список переменных значения которых требуется внести)

Во избежание повторной отправки данных на сервер используется функция exit() завершающая выполнение скрипта.

Вывод информации из базы данных

Для вывода информации необходимо установить соединение с базой данных. За это отвечает следующий скрипт:

<?php

$db = mysql\_connect("localhost","ash","qwerty") ordie ("НевозможноподключениекMySQL".mysql\_error());

mysql\_select\_db("phenology", $db);

mysql\_query("SET names cp1251");

?>

Функция mesql\_connectпредназначена для подключения к серверу. Далее следует функция для выбора базы данных с которой предстоит работать. Функция mysql\_querryустанавливает кодировку для корректного отображения символов.

После установки соединения необходимо выбрать информацию из базы данных. За это отвечает mysql\_query(). Данный скрипт посылает sqlзапрос для выборки:

SELECT список полей FROMназвание таблицы.

Функция mysql\_num\_rows() возвращает количество рядов результата запроса.

Функция mysql\_fetch\_array() возвращает массив с обработанным рядом результата запроса, или FALSE, если рядов больше нет.

Для вывода информации на страницу используется функция echo.

# ПРИЛОЖЕНИЕ В

Руководство пользователя

Для регистрации на сайте необходимо перейти по ссылке в окне авторизации (см. рис. 11).

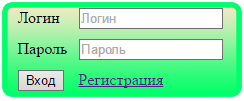


Рисунок 11 - Форма авторизации

После перехода по ссылке пользователь будет перенаправлен на страницу регистрации (см. рис. 12).

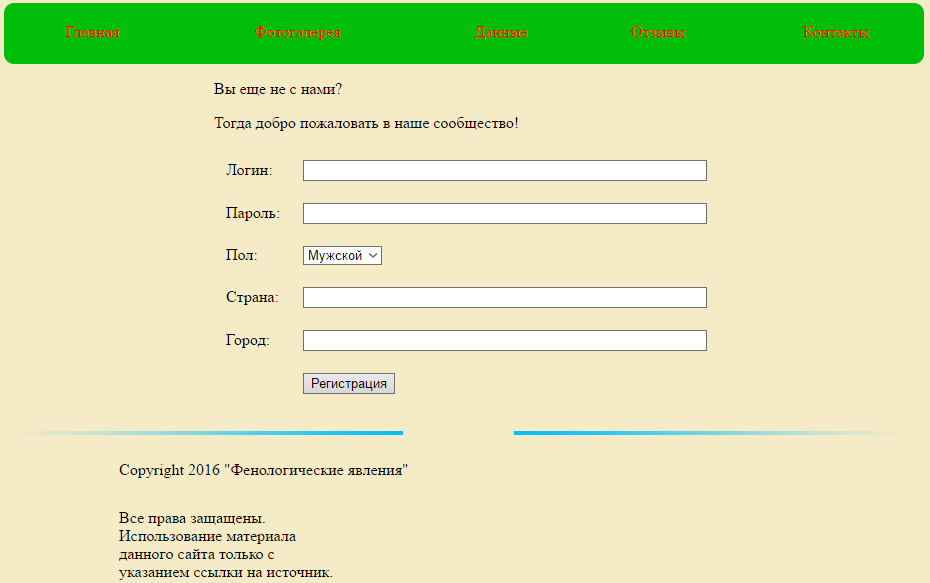


Рисунок 12 - Страница для регистрации

Для того что бы зарегистрироваться необходимо заполнить все поля на форме регистрации и нажать кнопку «Регистрация». Если оставить поле незаполненным и нажать кнопку, то появится незаполненного поля появится предупреждающая надпись с требование заполнить указанное поле. Аналогичные сообщения появляются и на других страницах где необходимо ввести какую-либо информацию (см. рис. 13).

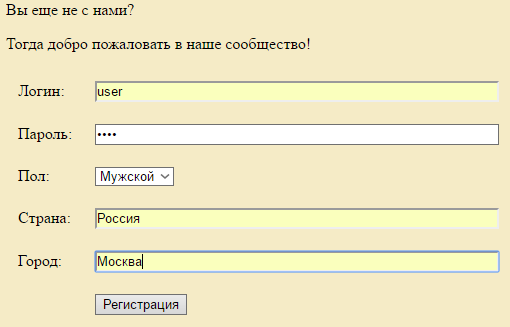


Рисунок 13 - Форма регистрации

После успешной регистрации пользователь будет перенаправлен на титульную страницу сайта (см. рис. 14).



Рисунок 14 - Титульная страница сайта

Если пользователь уже зарегистрирован на сайте, то для авторизации необходимо ввести свой логин и пароль в форме авторизации и нажать кнопку «Вход» (см. рис. 15).

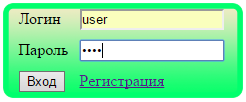


Рисунок 15 - Заполненная форма авторизации

После успешной авторизации вместо формы авторизации пользователю будет показана форма приветствия в которой отображается ник авторизованного пользователя и кнопка для того что бы разлогиниться (см. рис. 16).

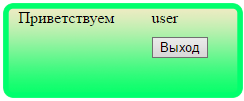


Рисунок 16 - Форма приветствия

Для перехода между страницами в шапке сайта расположено меню (см. рис. 17).



Рисунок 17 - Меню сайта

Для добавления или просмотра фенологических явлений необходимо перейти по ссылке «Данные» в меню сайта. После чего откроется страница для просмотра и добавления фенологических явлений (см. рис. 18).

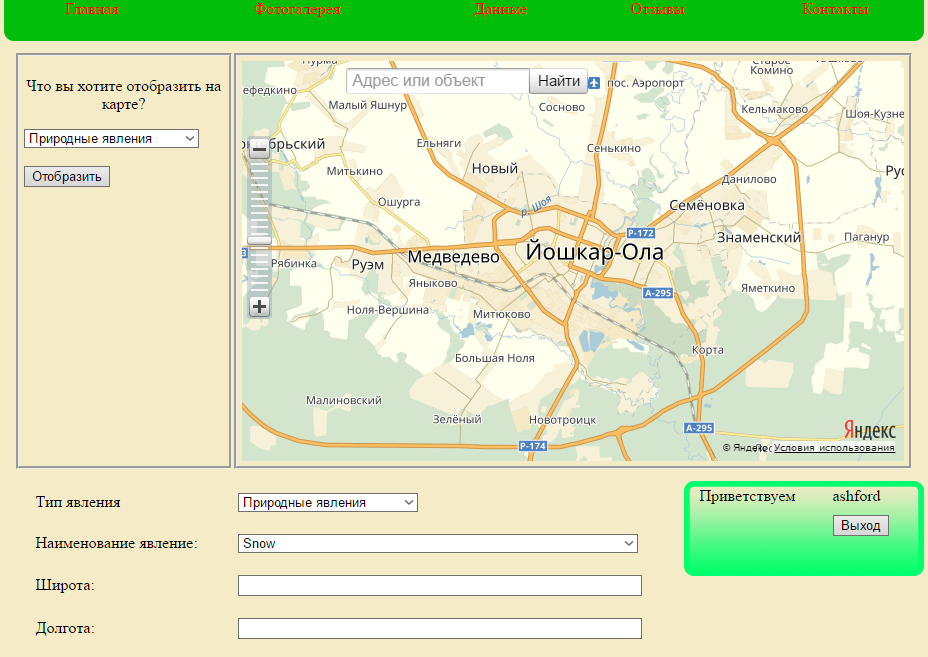


Рисунок 18 - Страница для добавления феноменов

Что бы добавить информацию о фенологическом явлении необходимо заполнить все поля в форма добавления и нажать кнопку «Добавить запись». Для указания координат Широты и Долготы необходимо перейти к карте, расположенной выше и щелкнуть по тому месту где данное явление наблюдается, после чего координаты места будут переданы в соответствующие поля (см. рис. 19).

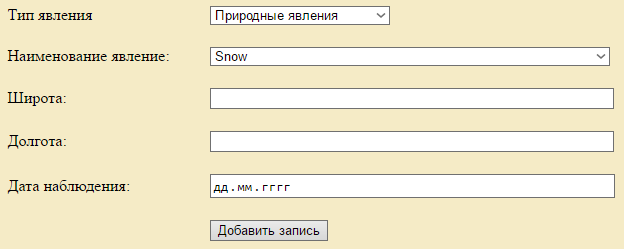


Рисунок 19 - Форма добавления феноменов

После успешного добавления записи информация о явлении будет отображена в таблице ниже (см. рис. 20).

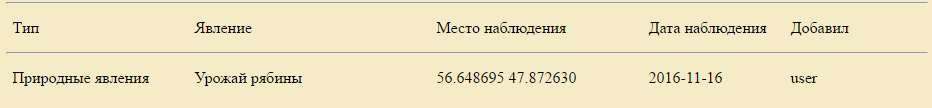


Рисунок 20 - Форма отображения феноменов

Для того что бы оставить отзыв необходимо в меню сайта перейти по ссылке «Отзывы» (см. рис. 21).

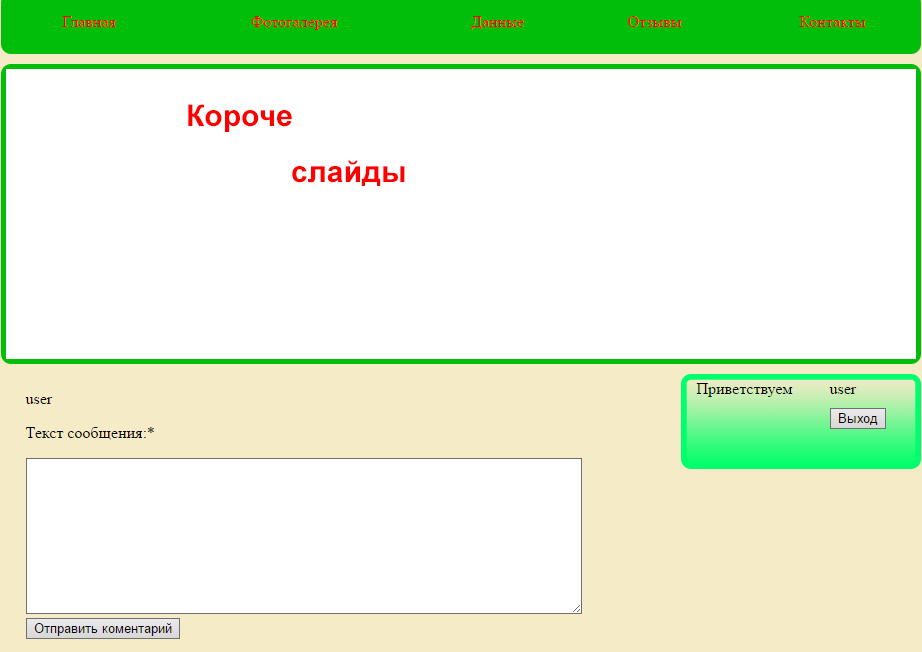


Рисунок 21 - Страница отзывов

Что бы оставить отзыв необходимо заполнить текстовое поле и нажать кнопку «Отправить отзыв» (см. рис. 22).

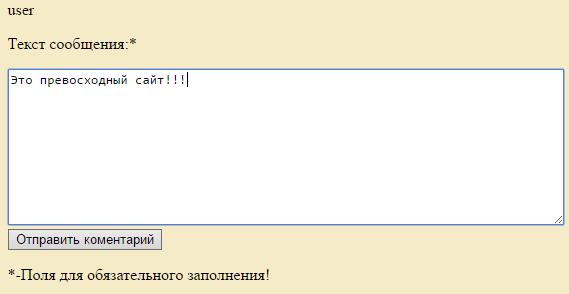


Рисунок 22 - Форма добавления отзыва

После успешного добавления отзывы пользователей будут отображены ниже на странице (см. рис.23).

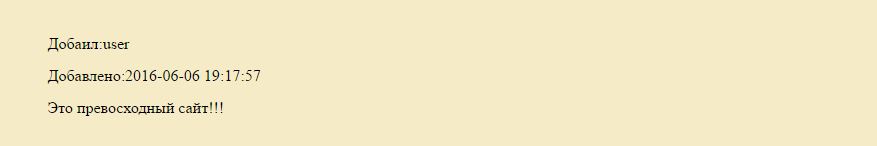


Рисунок 14 - Форма отображения отзывов

Последний лист дипломной работы.

Дипломная работа выполнена мной самостоятельно. Использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

Отпечатано в \_\_\_\_\_\_\_\_экземплярах.

Библиография\_\_\_\_\_\_\_\_ наименований.

Один экземпляр сдан на отделение.

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_20 \_\_\_ г.

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

Ф.И.О.