МИНОБРНАУКИ РОССИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ОБРАЗОВАНИЯ**

**“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”**

Факультет *компьютерных наук*

Кафедра *технологий обработки и защиты информации, информационных технологий управления*

*Web-приложение учета подарков “Golden fish”*

*Курсовой проект*

*09.03.02 Информационные системы и технологии*

Допущен к защите

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*С.А. Никонова, 3 курс, д/о*

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*Е.М. Скворцова, 3 курс, д/о*

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*А.А. Шурыгина, 3 курс, д/о*

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*В.С. Тарасов, преподаватель*

Воронеж 2020

**Содержание**

1. Введение
2. Постановка задачи
3. Анализ предметной области
4. Проектирование
5. Реализация
6. Тестирование
7. Заключение

Введение

В современном мире человеку предъявляются высокие требования: он должен быть успешен во всех сферах, успевать всё и везде. За последнее столетие темп жизни серьезно возрос, и тенденции к его уменьшению не предвидится. В суете мы часто забываем о действительно важных вещах: порой люди не помнят о предстоящих праздниках или у них попросту не хватает времени к ним подготовится. Бывают ситуации, когда человек за день до события заходит в магазин и покупает первый попавшийся под руку подарок. Конечно, это не способствует гармоничным взаимоотношениям и часто заставляет человека испытывать чувство вины.

Данный курсовой проект посвящен созданию приложения, которое поможет человеку организовать процесс выбора подарков, позволит составлять список желаний, просматривать списки желаний друзей и решать, кто подарит тот или иной подарок.

При наличии такого приложения у дарителя отпадет необходимость мучительного выбора, а получатель всегда будет доволен полезным и нужным подарком, о котором он давно мечтал. Система поможет лучше узнать человека, а в некоторых моментах даже развлечься. Стоит отметить, что наличие данного приложения не означает полного отказа от спонтанных и неожиданных подарков, а его использование носит исключительно вспомогательный характер.

Постановка задачи

Целью данной курсовой работы является реализация веб-приложения, позволяющего зарегистрированным пользователям создавать свои списки желаний (подарков, которые они хотели бы получить), а также просматривать списки желаний других пользователей - друзей, отмечать те желания, которые они могли бы осуществить, планируя таким образом список подарков для своих друзей.

Приложение должно выполнять перечисленные функции и решать основную задачу — управление списками желаний. Для этого к приложению выдвинуты следующие требования:

* удобный, интуитивно понятный пользовательский интерфейс;
* минималистичный дизайн;
* стабильная работа в браузере Google Chrome v.80.0.3987.

Архитектура разрабатываемого приложения должна отвечать требованиям шаблона проектирования MVC, иметь front-end и back-end части. Выбор данной архитектуры обусловлен стремлением отделить бизнес-логику и интерфейс, чтобы реализовать независимые друг от друга компоненты, позволяющие легко поддерживать проект в дальнейшем. В соответствии с данной архитектурой должны быть реализованы следующие компоненты приложения:

* база данных;
* пользовательский интерфейс, клиентская часть приложения;
* бизнес-логика проекта, обрабатывающая запросы к базе данных на удаленном сервере;
* взаимодействие между клиентской и серверной частью (RESP API).

Анализ предметной области

**3.1. Глоссарий**

*Желание* - стремление к осуществлению чего-либо или обладанию чем-либо.

*Подарок* - осуществление желания другого человека.

**3.2 Анализ существующих решений**

3.2.1. Lester Wish

[LesterWish](http://lesterwish.com/) — это портал организации подарков. Первый в списке запросов через поисковую систему Google и Yandex. Позволяет создать событие и составить список желаний для него, отправить ссылку друзьям через социальные сети, SMS или по электронной почте, и позволить им забронировать подарки, чтобы не было повторений.

Достоинства:

- создание событий;

- много полей для описания события;

- возможность добавления желания прямо из онлайн-каталога известных магазинов;

- возможность поделиться списком любым способом.

Недостатки:

- нельзя добавить свое желание (только выбрать товар из имеющихся каталогов);

*Рис. 1. Главная страница сайта LesterWish*

3.2.2. MyWishList

[MyWishList](http://mywishlist.ru/) - самый крупный российский сервис для составления списков желаний. Полезен пользователям, желающим подготовить список ожидаемых подарков, а также создать список для личного использования, для планирования собственных приобретений.

Достоинства:

- создание поздравлений;

- добавление пользователей в друзья;

- возможность оставлять комментарии;

- создание скрытых желаний (не видны никому, видны только друзьям, видны всем).

Недостатки:

- непрезентабельный интерфейс.



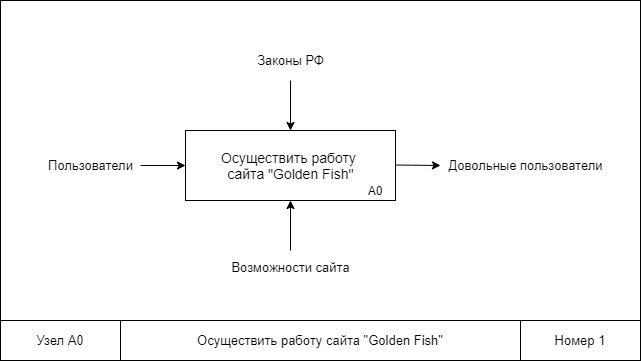
*Рис. 2. Страница авторизованного пользователя сайта* MyWishList

Проектирование

**4.1. IDEF0**

IDEF0 диаграмма предназначенная для формализации и описания бизнес-процессов. В IDEF0 рассматриваются логические отношения между функциями системы, а не их временная последовательность.

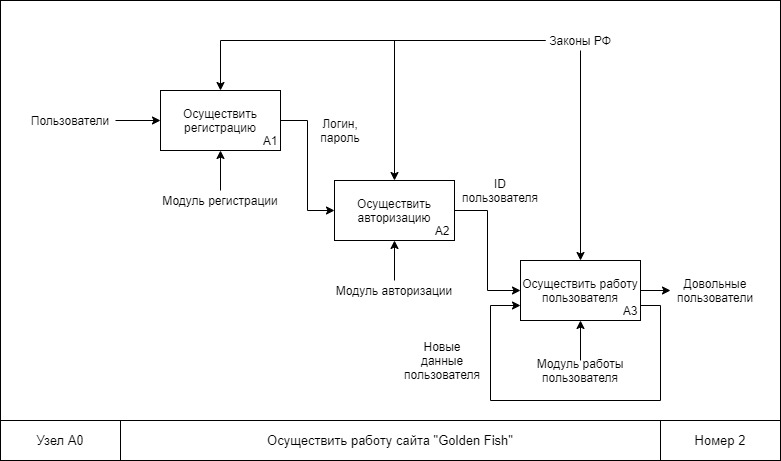
На рисунке 3 представлена контекстная диаграмма разрабатываемого приложения.



*Рис. 3. Контекстная диаграмма разрабатываемого приложения*

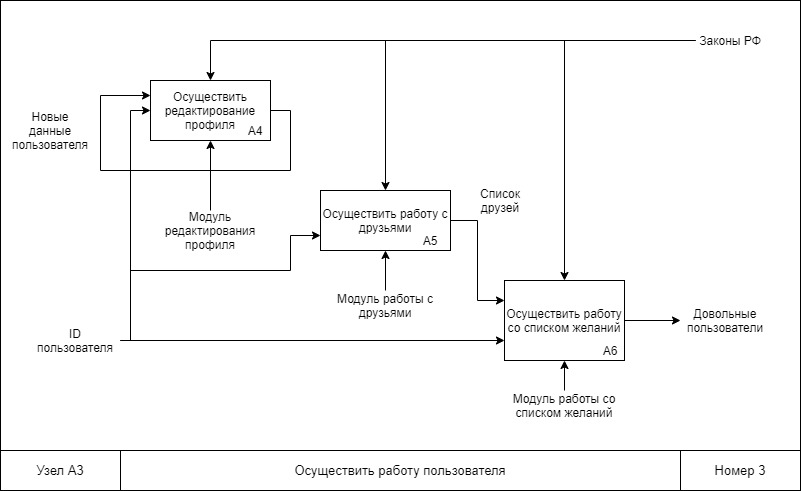
Работу сайта регулируют законы РФ, а именно законы о деятельности сайтов. На вход в Систему поступают пользователи, которые используют возможности сайта и получают от этого удовлетворение. На выходе Система получает довольных пользователей, которые снова захотели бы посетить сайт.

Детальное представление диаграммы верхнего уровня представлено на рисунке 4.



*Рис. 4. Детальное представление диаграммы верхнего*

Блок А3 нуждается в более глубокой детализации. Дочерняя диаграмма изображена на рисунке 5.



*Рис. 5. Диаграмма блока А3*

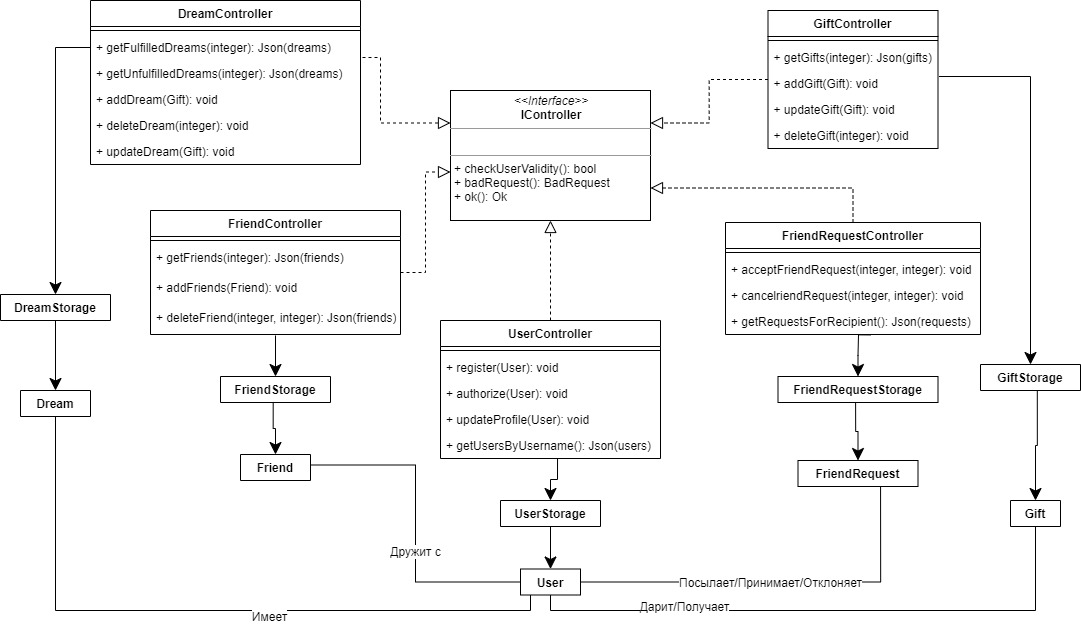
**4.2. Диаграмма прецедентов**

**4.3. Диаграмма классов**

Диаграмма классов описывает систему с точки зрения ее проектирования, показывая ее структуру: строятся классы, интерфейсы и отношения между ними.

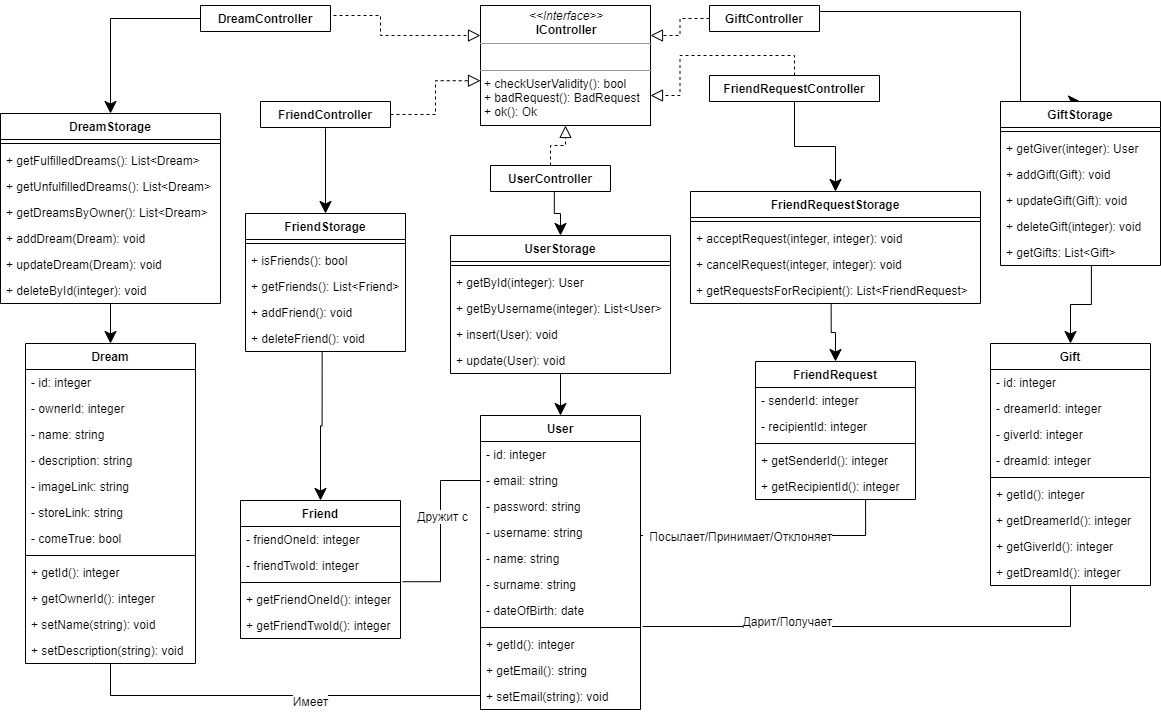
Диаграмма классов разрабатываемой Системы с подробным устройством классов-контроллеров расположена ниже. Стоит отметить, что данный проект строится на шаблоне проектирования MVC. Подробнее об этом написано в пункте «Обоснование архитектуры проекта».

**­­­­**



*Рис. 6. Диаграмма классов. Классы-контроллеры*

Та же диаграмма, но с детальным устройством классов, относящихся к модели, находится на рисунке 6.

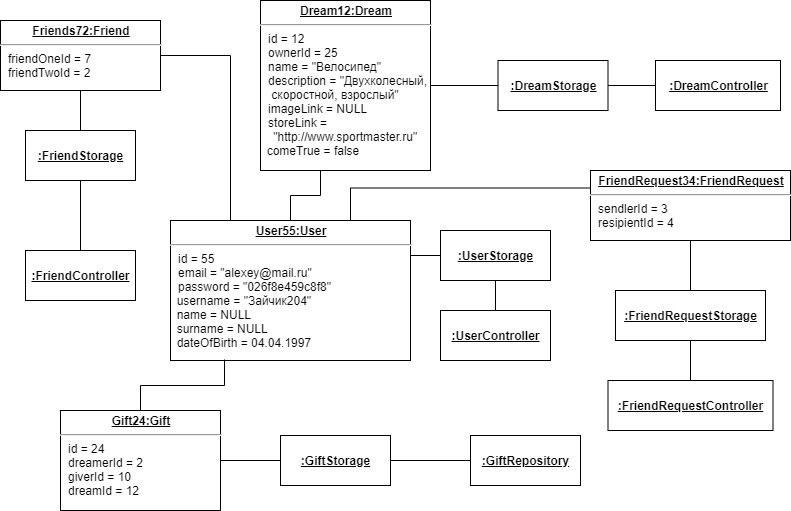


*Рис. 7. Диаграмма классов. Классы, относящиеся к модели*

**4.4. Диаграмма объектов**

На диаграмме объектов отображаются экземпляры классов системы с указанием текущих значений их атрибутов и связей между объектами.

В разрабатываемой системе есть много классов, реализующих только методы. Они соответственно представлены прямоугольниками без атрибутов.



*Рис. 8. Диаграмма объектов*

**4.5. Диаграммы последовательности**

**4.6. Диаграммы коммуникации**

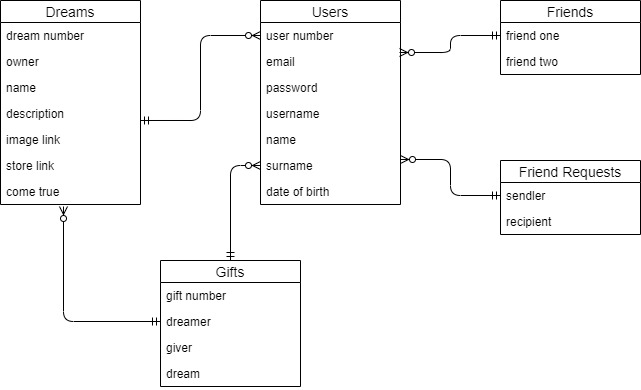
**4.7. Диаграмма состояний**

**4.8. Диаграмма активностей**

**4.9. Диаграмма развертывания**

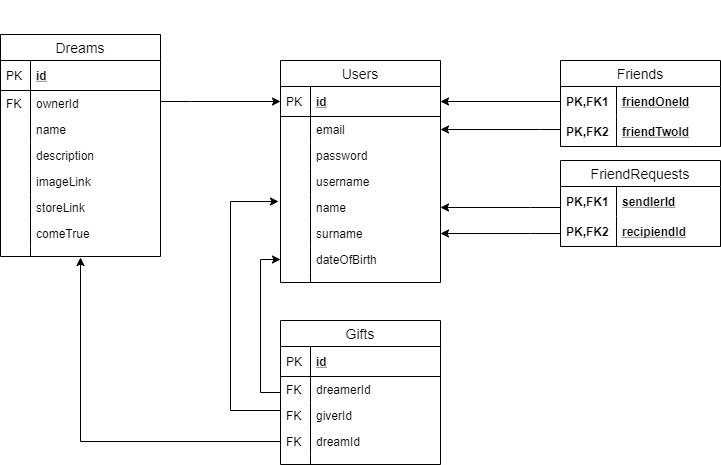
**4.10. ER-диаграмма и схема базы данных**

ER-диаграмма показывает, как разные «сущности» (люди, объекты, концепции и так далее) связаны между собой внутри системы. На рисунке () представлена ER-диаграмма разрабатываемого приложения. Она состоит из пяти сущностей: пользователи, друзья, заявки в друзья, желания, подарки. Все связи на диаграмме в отношении «один-ко-многим».



*Рис. . ER-диаграмма приложения*

Данная ER диаграмма преобразовывается в следующую схему базы, представленную на рисунке ().



*Рис. . Схема базы данных приложения*

База данных состоит из пяти таблиц:

1. Users

* Уникальный идентификатор пользователя в Системе (первичный ключ)
* Электронная почта
* Пароль
* Псевдоним пользователя в Системе
* Имя (необязательно поле)
* Фамилия (необязательное поле)
* Дата рождения (необязательное поле)

1. Friends

* Идентификатор первого друга (ссылается на уникальный идентификатор в таблице Users)
* Идентификатор второго друга (ссылается на уникальный идентификатор в таблице Users)

(friendOneId, friendTwoId) – составной первичный ключ.

1. FriendsRequests

* Уникальный идентификатор пользователя, от которого исходит заявка (ссылается на уникальный идентификатор в таблице Users)
* Уникальный идентификатор пользователя, которому приходит заявка (ссылается на уникальный идентификатор в таблице Users)

(sendlerId, recipientId) – составной первичный ключ.

1. Dreams

* Уникальный идентификатор мечты
* Идентификатор пользователя, имеющего эту мечту (ссылается на уникальный идентификатор в таблице Users)
* Название мечты
* Описание (необязательно поле)
* Ссылка на картинку/фотографию желания (необязательное поле)
* Ссылка на интернет-магазин, в котором можно приобрести подарок (необязательное поле)
* Флаг, указывающий сбылась ли мечта

1. Gifts

* Уникальный идентификатор подарка
* Идентификатор пользователя, мечтающего о данном подарке (ссылается на уникальный идентификатор в таблице Users)
* Идентификатор пользователя, планирующего подарить данный подарок (ссылается на уникальный идентификатор в таблице Users)
* Идентификатор мечты (ссылается на уникальный идентификатор в таблице Dream)

**4.11. Обоснование архитектуры проекта**