# **ВВЕДЕНИЕ**

С каждым годом  не многие могут правильно назвать даты начала и окончания Великой Отечественной войны, ее причины, участников и основные сражения. Молодежь все меньше ходит в библиотеку и все меньше интересуется Великой Отечественной войной. Большинство людей привыкли узнавать новости через интернет. В интернете не так много ресурсов где можно узнать о Великой Отечественной войне.

Для того что бы ни забывать о ветеранах Великой Отечественной войны

поможет сервис «OurMemory.by». Данный сервис прост в использовании и предлагает узнать информации о ветеранах, также можно самому добавлять своих бабушек и дедушек которые прошли Великую Отечественную войну, добавлять статьи о Великой Отечественной войне, создавать альбомы фотографии.

Целью проектирования и программной реализации является разработка приложения, которое станет в Беларуси информационным источником о ветеранах Великой Отечественной войны.

В данном дипломном проекте была принята попытка разработать соответствующее программное обеспечение, которое представляла бы из себя Web Api приложение для хранения и обработки данных о ветеранах.

В настоящей пояснительной записке описывается процесс проектирования и создания программного обеспечения. Она состоит из следующих глав:

* постановка задачи;
* моделирование и инструментарий – выбор инструментов для разработки приложения и моделирование структуры базы данных и архитектуры приложения;
* реализация программного обеспечения – описание структуры приложения;
* руководство пользователя;
* тестирование программного обеспечения – описание тестов, проводимых над приложением;
* определение экономической эффективности разработки программного обеспечения;
* охрана труда.

Дипломный проект выполнен с учётом указаний и требований к выполнению дипломного проекта [15].

# ПРЕДМЕТ РАЗРАБОТКИ В КОНТЕКСТЕ AS-IS И TO-BE

## Обзор состояния вопроса

Для жителей нашей страны слово «Победа» исполнено глубочайшего смысла. В нем смешались память о выживших и погибших, слезы радости и скорби. Проходят годы, десятилетия, сменяются поколения, но разве можно предать забвению подвиг воинов, защитивших не только нашу жизнь, но и само звание человека, которое хотел растоптать фашизм.

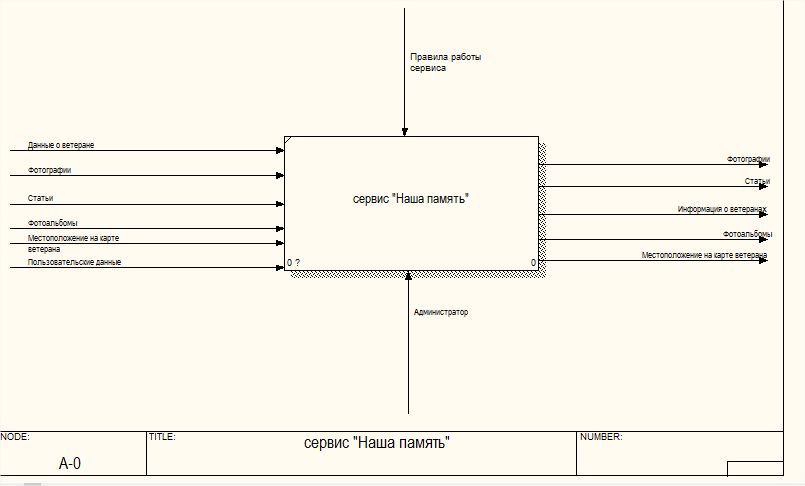
В течение нескольких десятилетий был собран богатый материал ветеранах Великой Отечественной войны. Мы считаем, что нам необходимо продолжать начатую работу, которая, по нашему мнению, позволит:

* лучше узнать историю Великой Отечественной войны от непосредственных участников тех событий;
* обратить внимание людей на основные этапы Великой Отечественной войны, историческую роль Советского Союза в разгроме фашистской Германии;
* воспитывать гражданина-патриота своей Родины на примере истории своей семьи и родного края;
* развивать познавательный интерес людей, формировать навыки исследовательской деятельности;
* развивать коммуникативные качества людей при поиске и сборе информации;

На сегодняшний день уже существуют системы о ветеранах. Наиболее популярный из всех «pamyatnaroda.mil.ru».Данный сервис дает возможность просматривать информацию о ветеранах, боевых операциях, воинских захоронениях. Однако этот ресурс не обладает возможность добавлять информацию и медиа контент о ветеранах самостоятельно. Поэтому имеет смысл создание Web-приложения которое смогло бы улучшить имеющиеся интернет ресурсы о ветеранах и дать возможность пользователям самостоятельно добавлять информацию.

## Модель AS-IS

Изучение состояния проблемы является обязательной частью любого проекта создания или развития системы. Построение функциональной модели «КАК ЕСТЬ» позволяет четко зафиксировать, какие процессы осуществляются в системе, какие информационные объекты используются при выполнении процессов и отдельных операций. Контекстная диаграмма модели «КАК ЕСТЬ» представлена на рисунке 1.1.



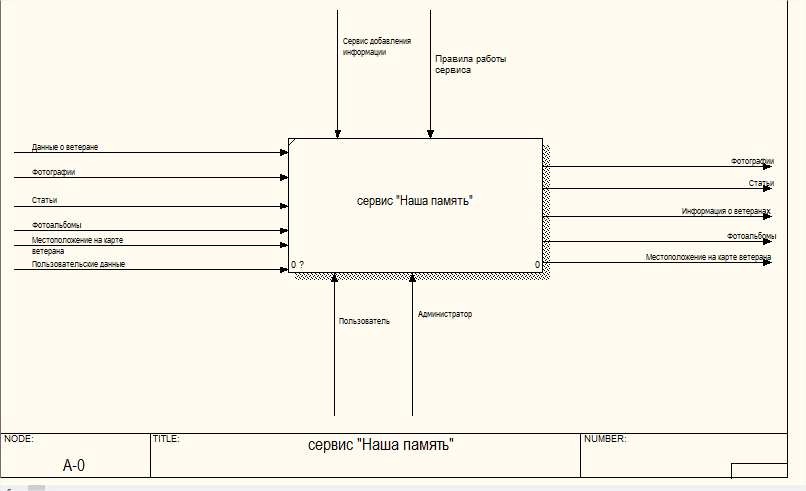
* + - 1. Модель AS-IS

## Модель TO-BE

Существенным недостатком модели AS-IS является то, что процесс коммуникации пользователя с системой происходит через посредника (администратора системы). На наш взгляд для улучшения существующей ситуации следует отказаться от посредников. В создаваемой коммуникация между пользователем и системой происходит без посредников.

Найденные в модели AS-IS недостатки исправляются путем создания модели TO-BE, то есть модели новой организации процессов компании. Функциональная модель TO-BE позволяет уже на стадии проектирования будущей системы определить эти изменения. Применение функциональной модели TO-BE позволяет не только сократить сроки внедрения информационной системы, но также снизить риски, связанные с невосприимчивостью персонала к информационным технологиям.

С учетом сказанного была построена модель TO-BE, показанная на рисунках 1.2 - 1.3.



* + - 1. Контекстная диаграмма модели TO-BE

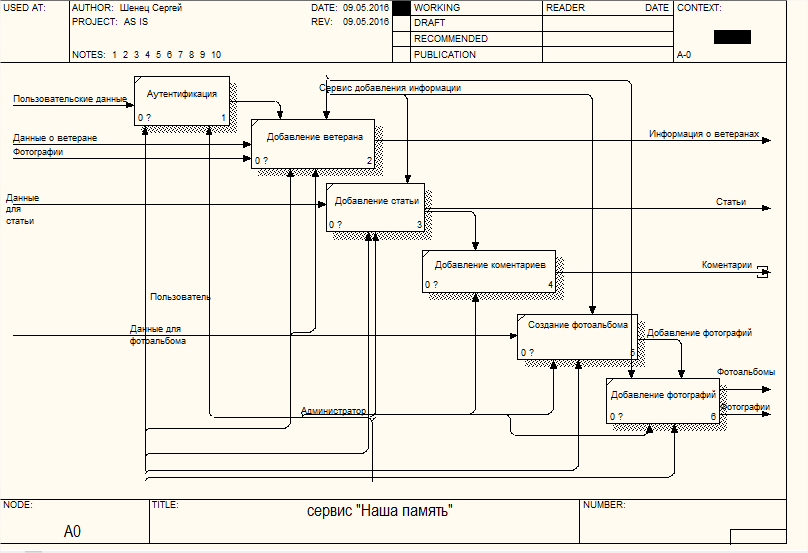


Рисунок 1.3 – Декомпозиция контекстной диаграммы модели TO-BE

# ЦЕЛЬ И ЗАДАЧИ ПРОЕКТА

Задачей дипломного проекта является проектирование и реализация серверной части сервиса «Наша память», а также осуществления его внедрения и развёртывания на целевом Интернет-сервере, настройка и оптимизация параметров нормальной работы системы.

Цель сервиса «Наша память» заключается в том, чтобы связать вместе тех, кто ищет информацию о ветеранах Великой Отечественной войны. Информация может быть разнообразной: отображение на карте, биография ветеранов, статьи, комментарии к статьям, изображения.

Таким образом, основное назначение сервиса – это почтить память о ветеранах а также быстрый и удобный способ найти своих родственников, которые участвовали в Великой Отечественной войне.

Сервис должен обеспечивать разделение прав доступа и быть реализована для трех категорий пользователей:

1. незарегистрированный пользователь;
2. зарегистрированный пользователь;
3. администратор.

В общем случае сервис должен для всех пользователей предоставлять следующие возможности:

* получение данных о ветеранах;
* получение изображений связанных с Великой Отечественной войной.
* получение статей;
* получение комментариев к статьям;
* получение количества просмотров категорий;
* просмотр дополнительных сведений о Великой Отечественной войне.

Для пользователя, который уже был зарегистрирован в системе, должны предоставляться следующие возможности:

* публиковать данные о ветеранах, а также возможность его последующего редактирования и удаления;
* делать метку на карте о местоположении ветерана;
* редактировать метку на карте, которая была сделана конкретным пользователе;
* добавление статей;
* создание фотоальбомов, а также возможность его последующего редактирования и удаления;
* удаление фотографий, которые были добавлены конкретным пользователем;
* удаление статей, которые были добавлены конкретным пользователем;
* редактирование статей, которые были добавлены конкретным пользователем;
* изменения фотографии профиля;
* изменения контактных данных;
* изменение пароля;
* выхода из сервиса.

Во всех случаях добавления или редактирования соответствующих данных система должна контролировать правильность вводимой информации пользователем и по возможности исключать ситуации, которые могут привести к ошибочным действиям со стороны пользователей системы.

Сервис должен дополнительно включать в себя:

1. фирменный стиль (логотип, цвет, стиль);
2. фильтрация данных;
3. поиск данных.

С целью облегчения поддержки, сопровождения, дальнейшего развития (расширения) системы, добавления нового функционала или исправление старого, программный продукт должен быть спроектирован и создан на основе слоёной архитектуры построения программных систем (в частности, для проектирования и построения использовать шаблон проектирования WEB API).

Реализация шаблона WEB API должна быть осуществлена на базе фреймворка ASP.NET WEB API.

Серверная часть системы должна отвечать за доступ к данным и содержать бизнес-логику приложения.

В качестве уровня доступа к данным можно использовать одну из следующих систем управления базами данных (СУБД): MySQL или Microsoft SQL Server Express. Данные СУБД обеспечивают централизованное структурированное хранение всех данных системы, гарантируя их целостность и непротиворечивость, а также предоставляют множество сервисов низкого уровня для чтения данных из хранилища, сохранения данных, изменения их структуры и прочее. Реализация команд выборки данных, контроль целостности и непротиворечивости данных может осуществляться с помощью соответствующих хранимых процедур, триггеров и других объектов, предоставляемых сервером баз данных. В качестве языка структурированных запросов рекомендуется использовать SQL, а не привязываться к конкретным диалектам данного языка, используемых в выбранных СУБД.

В любом случае, система должна быть спроектирована таким образом, чтобы компоненты системы не завесили от уровня данных, и в последующем, можно было без лишних затрат поменять целевую СУБД или тип хранилища данных и при это остальные компоненты не пришлось адаптировать под новое хранилище.

Уровень бизнес-логики будет разворачиваться на сервере приложений и представлять собой ядро системы. На этом уровне должна быть сосредоточена большая часть бизнес-логики системы:

* алгоритмы авторизации пользователя системы, система проверки прав доступа;
* правила обработки данных, такие как: проверка правильности заполнения данных пользователем, проверка и организация взаимосвязей данных, правила движения информации внутри системы;
* классы для подключения к базе данных и выполнения транзакций;
* классы и алгоритмы для работы с таблицами базы данных.

На уровень представления необходимо вынести простейшую бизнес-логику: интерфейс авторизации, алгоритмы шифрования, функции ввода и отображения данных, первичную проверку вводимых значений на допустимость и соответствие формату, несложные операции (сортировка, группировка, подсчет значений) с данными, уже загруженными на рабочие станции клиента.

Система должна быть разработана под семейство операционных систем Windows и языке программирования C# с использованием интегрированной среды разработки Web-приложений Microsoft Visual Studio 2015.

Web-система должна быть развёрнута под управлением IIS версии 10.0.

# ЛОГИЧЕСКОЕ МОДЕЛИРОВАНИЕ

## Выбор методологий моделирования и инструментария

Для визуального моделирования проблемной области было отдано предпочтение Rasional Rose компании Rational Software. Данное средство является простым и полностью интегрированным решением для разработки ПО, включая Интернет-решения.

Rational Rose является стандартом дефакто среди инструментов проектирования приложений. Ни одно другое CASE-средство не предлагает такую широту и глубину решений как платформа Rational. С помощью Rational Rose можно визуализировать, изменять и тестировать модель.

Одно из неоспоримых преимуществ Rational Rose – обратное проектирование, поскольку разработчику и проектировщику важно увидеть перед изменениями уже работающую систему в нормальном графическом представлении. Как правило визуально-графический ряд оказывает куда большее воздействие нежели пролистывание технических заданий и программных текстов. Тем более что, проект, подвергшийся обратному проектированию может быть доработан и вновь сгенерирован (а впоследствии и скомпилирован). Rational Rose предоставляет для этого все необходимые средства.

Rational Rose является лидирующим инструментом визуального моделирования, поскольку он имеет все необходимые возможности - поддержку UML, многоязыковую поддержку итерационной разработки, полную поддержку командной разработки, компонентно-базированную разработку с поддержкой ведущих архитектур и таких компонентных моделей, как WinDNA и J2EE/SE/ME, легкость применения, оптимизированную интеграцию и многое другое. [15]

Для проектирования и моделирования данных был использован инструментарий AllFusion ERwin Data Modeler (ERwin) компании Computer Associates. ERwin позволяет проектировать, документировать и сопровождать базы данных, хранилища данных и витрины данных (data marts). Основные аргументы и факты для разработчиков ПО в пользу использования данного инструментария:

* поддерживается прямое (создание БД на основе модели) и обратное (генерация модели по имеющейся базе данных) проектирование для 20 типов СУБД;
* увеличивает производительность труда благодаря удобному интерфейсу и автоматизации рутинных процедур;
* ERwin является стандартом де-факто;
* позволяет повторно использовать компоненты созданных ранее моделей, а также использовать наработки других разработчиков. Повышается эффективность;
* позволяет переносить структуру БД из СУБД одного типа в СУБД;
* позволяет документировать структуру БД;
* продукт можно использовать на всех стадиях жизненного цикла баз;
* позволяет получить точную и наглядную информацию, где хранятся данные и как получить к ним доступ;
* позволяет, используя визуальные средства, описать структуру БД, а затем автоматически сгенерировать файлы данных для любого типа СУБД. [16]

## Разработка диаграмм вариантов использования

Диаграмм вариантов использования описывает функциональное назначение системы, т.е. то, что система будет делать в процессе своего функционирования, и является исходной концептуальной моделью системы в процессе ее проектирования и разработки.

Разработка диаграммы вариантов использования преследует следующие цели:

* определить общие границы и контекст моделируемой предметной области на начальных этапах проектирования системы, а также сформулировать общие требования к функциональному поведению проектируемой системы;
* разработать исходную концептуальную модель системы для ее последующей детализации в форме логических и физических моделей;
* подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями.

Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью так называемых вариантов использования. При этом актером (actor) называется любая сущность, взаимодействующая с системой извне.

В свою очередь, вариант использования (use case) служит для описания сервисов, которые система предоставляет актеру, т.е. каждый вариант использования определяет некоторый набор действий, совершаемый системой при диалоге с актером. При этом ничего не говорится о том, каким образом будет реализовано взаимодействие актеров с системой.

Техническое задание реализуемое обычно в виде модели вариантов использования, включает в себя список действующих лиц с указанием их ролей, список вариантов использования, диаграммы вариантов использования и их описание.

Язык моделирования может быть представлен нотациями UML, Буча, OMT и др. В данном случае нами была выбрана нотация UML, а в качестве инструментального средства – Rational Rose [16].