1. **ПОСТАНОВКА ЗАДАЧИ**

## 2.1 Определение требований к программной системе

Постановка задачи - один из наиболее важных этапов при создании программы. От того, насколько полно, точно и ясно определены требования к разрабатываемому ПО, его функции и предполагаемые возможности, во многом зависит качество и стоимость разработки [5].

Одной из самых важных функций любого языка программирования является предоставление возможностей для управления программой вручную. Это подразумевает создание пользовательского интерфейса.

Согласно общим требованиям, графический интерфейс разрабатываемого приложения должен:

* ориентироваться на пользователя, который общается с программой на внешнем уровне взаимодействия;
* сохранять стандартизированное назначение и местоположение на экране графических объектов, работающих в среде Windows.

Большинство программных приложений работают в диалоговом режиме.

Интерфейс программного приложения будет разрабатываться с учетом общих требований к пользовательскому интерфейсу. Цветовая гамма и оформление должно быть выдержано в стиле.

Доступ к информации, содержащейся в разрабатываемой базе данных, будет защищен. Информация, хранящаяся в базе, будет доступна лишь тем, кто имеет наивысший приоритет доступа. Все действия по добавлению и удалению информации будет осуществляться через пользовательские окна и только пользователями с наивысшим приоритетом.

Информационная система должна обладать следующими характеристиками:

* простота навигации;
* стабильность информационных ресурсов;
* оперативность обновления информации;
* доступность для пользователей;
* удобный интерфейс;
* единство дизайна всех разделов.

Требования к информационной системе со стороны администрирования:

* возможность добавлять данные;
* возможность редактирования и удаления данных;
* возможность просматривать записи нагрузки по выбранной сущности;
* загрузка и парсинг Excel-файла нагрузки

Минимальными и достаточными требованиями по конфигурации оборудования являются:

* MySQL8;
* Java 11.0.8;
* Maven 3.6.0;
* объем оперативной памяти не мене 2 Гб;
* свободное дисковое пространство не менее 200 Mб.

## 2.2 Описание аналогов системы

Основными требованиями к информационной системе учёта нагрузки преподавателей является быстродействие, простота восприятия приложения пользователем, возможность хранить и обрабатывать данные напрямую из приложения и возможность парсить локально сделанный excel-файлы по шаблону. На данный момент функционал на рынке программного обеспечения представляют программы для учёта, обзор программ приведён ниже.

Программа «БИТ:Учет нагрузки преподавателей» предназначена для автоматизации планирования и учёта выполнения нагрузки профессорско-преподавательского состава. Реализует процессы планирования и учёта нагрузки по преподавателям и кафедрам, формирования учебных планов по утверждённым государством стандартам, а также всей необходимой отчетности [6].

Преимущества:

* Формирование учебных планов в соответствии со стандартом Gosinsp;
* Ведение учебных планов по стандартам 3-го поколения;
* Гибкая настройка параметров расчета нагрузки в соответствии с требованиями вуза;
* Программа может быть использована при создании информационной системы персональных данных любого класса, дополнительная сертификация решения не требуется.

Программа решает задачи

* Автоматизация составления и ведения академических и рабочих учебных планов с учётом обязательных и факультативных дисциплин, дисциплин по выбору, дисциплин специализации;
* Импорт-экспорт учебных планов в формат GosInsp для прохождения аккредитации учебных планов по государственным стандартам;
* Планирование нагрузки по кафедрам;
* Распределение нагрузки кафедры по преподавателям;
* Учёт выполнения нагрузки профессорско-преподавательским составом;
* Предоставление актуальной отчетности.

Сопровождение: Предусмотрено бесплатное сопровождение в течение трех месяцев после покупки. Далее вы можете воспользоваться услугой консультационного сопровождения.

Автоматизированная система «Нагрузка ВУЗа»

АС «Нагрузка ВУЗа» обеспечивает комплексный подход к формированию и распределению учебной нагрузки учреждений ВПО [7]. Система рассчитана для работы в локальной сети и имеет три уровня доступа, которые определяют функционал доступный пользователям.

Администратору системы доступны следующие функции:

- проверка учебных планов на преемственность;

- формирование сведений об ожидаемом контингенте студентов;

- создание списка учебных групп на основе контингента студентов;

- задание норм на прием зачетов и экзаменов, руководство дипломными, курсовыми, диссертационными и другими видами работ;

- централизованное переименование дисциплин и закрепление их за кафедрами;

- определение параметров формирования потоков и учебной нагрузки;

- формирование учебной нагрузки кафедр на базе учебных планов и списка групп;

- расчет штатного расписания кафедр.

Заведующим кафедрами доступны следующие функции:

- закреплять учебную нагрузку за преподавателем или передавать ее другой кафедре;

- объединять группы в потоки и разбивать группу на подгруппы по различным видам занятий, например, лабораторным, практическим или курсовым работам;

- учитывать данные о распределении учебной нагрузки предыдущего учебного года;

- уточнять нагрузку в течение учебного года;

- создавать поручения кафедр, экспортируемые в программу автоматического составления расписания занятий;

- выполнять проверку корректности и соответствия распределенной нагрузки данным учебного отдела.

Преподавателям доступны следующие функции

- получать развёрнутую информацию по учебной нагрузке;

- заполнять необходимое количество часов на подготовку к занятиям;

- формировать свои индивидуальные планы, в которых можно конкретизировать содержание работ, выполняемых во «второй половине дня» (научная, организационно-методическая, учебно-методическая, воспитательная работа);

- вести учет пожеланий преподавателей по графику работы в течение недели;

Качественный результат достигается перекрестным контролем содержимого РУП всеми заинтересованными сторонами: кафедрами, деканатами, учебным управлением. Карты дисциплин кафедр позволяют осуществлять предварительное ознакомление кафедр с перечнем закрепленных за ними дисциплин и распределением аудиторных часов и форм контроля по семестрам (курсам, сессиям).

## 2.3 Обзор и обоснование выбора инструментальных средств

Для создания приложения язык программирования высокого уровня Java версии 11.0.8 и среда разработки IDEA.

IntelliJ IDEA анализирует код в поисках связей между символами во всех файлах и на всех языках, используемых в проекте. На основе этого анализа IntelliJ IDEA предоставляет помощь при написании кода, удобную навигацию, проверку ошибок в коде и, конечно, рефакторинги.

Предлагая варианты автодополнения, IntelliJ IDEA анализирует поток данных, чтобы предугадать возможный тип символа во время выполнения, и на основе этой информации уточняет варианты, автоматически добавляя приведение классов.

Зная все об использовании символов, IntelliJ IDEA предлагает эффективные и точные рефакторинги. Например, при переименовании класса в JPA-выражении, IntelliJ IDEA обновит все, что необходимо: от класса сущности до каждого выражения JPA, в котором он используется.

За основу проекта был взят Spring Boot. Он упрощает использование Spring Framework. Spring - один из наиболее широко используемых фреймворков для разработки интерпрайз-приложений, обеспечивающий продуманную модель программирования и конфигурации.

Целью создания данного фреймворка способствовало желание упростить разработку приложений на популярном в то время (да и сейчас) Java EE стеке технологий от компании Oracle, который на тот момент был очень сложен и не всегда удобен в использовании. В отличие от других платформ, Spring фокусируется на нескольких областях функционирования приложений и предоставляет для них широкий спектр дополнительных функций [8].

Одной из основных особенностей Spring Framework является использование паттерна Dependency Injection (DI, внедрение зависимости). DI помогает намного проще реализовывать необходимую приложениям функциональность, а также позволяет разрабатывать слабо связанные классы, делая их более универсальными. Spring Boot упрощает разработку проектов на Spring Framework, т.к.:

* Не требует развертывания war-файлов;
* Создает автономные приложения;
* Помогает напрямую встроить в приложение Tomcat, Jetty или Undertow;
* Не требует XML-конфигурации;
* Направлен на уменьшение объема исходного кода;
* Имеет дополнительную функциональность "из коробки";
* Простота запуска;
* Простая настройка и управление.

Благодаря таким функциям, как автоконфигурация, Spring Boot избавляет вас от написания лишнего кода и помогает избежать ненужной настройки

Для создания и работы с базой данных были использованы:

* Spring Data -реализация JPA для Java EE, предоставляет возможность сохранять в удобном виде Java-объекты в базе данных
* Liquibase - независимая от базы данных библиотека с открытым исходным кодом для отслеживания, управления и применения изменений схемы базы данных.

MySQL как база данных. Одной из самых популярных СУБД на сегодняшний день является MySQL, распространяемая свободно (с некоторыми ограничениями).

Эта серверная система способна эффективно функционировать во взаимодействии с интернет-сайтами и веб-приложениями. При этом она проста в освоении, что лишь увеличивает ее популярность, особенно среди «айтишников»-любителей [9].

Примечательно, что результатом огромной популярности MySQL стало появление в интернете множества руководств по освоению системы, а также огромного количества всевозможных плагинов и расширений, упрощающих работу с этой системой. Это в свою очередь лишь придало системе еще большей популярности.

Несмотря на отсутствие некоторого функционала, имеющегося у других СУБД, MySQL обладает достаточно обширным разнообразием доступных инструментов для создания приложений.

Преимущества MySQL

Помимо универсальности и распространенности СУБД MySQL обладает целым комплексом важных преимуществ перед другими системами.

В частности, следует отметить такие качества как:

* Простота в использовании. MySQL достаточно легко инсталлируется, а наличие множества плагинов и вспомогательных приложений упрощает работу с базами данных.
* Обширный функционал. Система MySQL обладает практически всем необходимым инструментарием, который может понадобиться в реализации практически любого проекта.
* Безопасность. Система изначально создана таким образом, что множество встроенных функций безопасности в ней работают по умолчанию.
* Масштабируемость. Являясь весьма универсальной СУБД, MySQL в равной степени легко может быть использована для работы и с малыми, и с большими объемами данных.
* Скорость. Высокая производительность системы обеспечивается за счет упрощения некоторых используемых в ней стандартов.