# 

# Содержание

Содержание 1

История изменений 2

1 Введение 3

1.1 Цели 3

1.2 Границы применения 3

1.3 Термины, аббревиатуры, сокращения 3

1.4 Ссылки 3

1.5 Краткий обзор 3

2 Общее описание 3

2.1 Описание изделия 3

2.1.1 Интерфейсы системы 3

2.1.2 Интерфейсы пользователя 3

2.1.3 Интерфейсы аппаратных средств ЭВМ 3

2.1.4 Интерфейсы программного обеспечения 3

2.1.5 Интерфейсы коммуникаций 3

2.1.6 Ограничения памяти 4

2.1.7 Действия 4

2.1.8 Требования настройки рабочих мест 4

2.2 Функции изделия 4

2.3 Характеристики пользователей 4

2.4 Ограничения 4

2.5 Предположения и зависимости 4

2.6 Распределение требований 4

3 Детальные требования 4

3.1 Функциональные требования 4

3.1.1 <Functional Requirement One> 5

3.2 Надежность 5

3.2.1 <Reliability Requirement One> 5

3.3 Производительность 5

3.3.1 <Performance Requirement One> 5

3.4 Ремонтопригодность 5

3.4.1 <Maintainability Requirement One> 5

3.5 Ограничения проекта 5

3.5.1 <Design Constraint One> 5

3.6 Требования к пользовательской документации 5

3.7 Используемые приобретаемые компоненты 5

3.8 Интерфейсы 5

3.8.1 Интерфейс пользователя 5

3.8.2 Аппаратные интерфейсы 5

3.8.3 Программные интерфейсы 5

3.8.4 Интерфейсы коммуникаций 5

3.9 Требования лицензирования 5

3.10 Применимые стандарты 5

Индекс 5

# История изменений

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Версия** | **Описание** | **Автор(ы)** |
| 2022-10-14 | 0.1 | Начальная ревизия | Антонова А.К, Терентьев А.А. |
| 2022-10-24 | 0.2 | Дополнение информации в некоторые разделы | Антонова А.К |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Введение

[The introduction of the **Software Requirements Specification (SRS)** should provide an overview of the entire **SRS**. It should include the purpose, scope, definitions, acronyms, abbreviations, references, and overview of the **SRS**.]

[Note: The Software Requirements Specification (**SRS**) captures the complete software requirements for the system, or a portion of the system. This document describes a typical **SRS** outline for a project using only traditional natural-language style requirements – with **no use-case modelling.**.]

[Many different arrangements of an **SRS** are possible. Refer to [IEEE830-1998] for further elaboration of these explanations, as well as other options for organizing an **SRS**.]

## Цели

В условиях пандемии приоритетным направлением учебных заведений становится безопасность студентов и преподавателей. В связи с этим институт переходит на дистанционную форму обучения. Студенты продолжают получать и сдавать домашние задания, но их проверка преподавателями уже не представляется возможной из-за ограниченного графика. Появляется необходимость в системе дистанционной самопроверки домашних заданий. Предполагается, что студенты будут проверять работы друг друга и выставлять оценки.

[Specify the purpose of this **SRS**. The **SRS** should fully describe the external behaviour of the application or subsystem identified. It also describes non-functional requirements, design constraints and other factors necessary to provide a complete and comprehensive description of the requirements for the software.]

## Границы применения

Система применима в рамках ВУЗа, ее используют студенты и преподаватели

[A brief description of the software application that the **SRS** applies to; the feature or other subsystem grouping; what Use-Case model(s) it is associated with; and anything else that is affected or influenced by this document.]

## Термины, аббревиатуры, сокращения

|  |  |
| --- | --- |
| Студент | Непосредственный участник учебного процесса высшего учебного заведения или техникума |
| Преподаватель | Сотрудник, преподающий какую-либо учебную дисциплину студентам в рамках конкретного учебного заведения |
| Работа (домашнее задание) | Задание, задаваемое преподавателем студенту для самостоятельного выполнения, загружаемое впоследствии студентом в систему в виде файла |
| Оценка | Мера оценивания работы, выраженная в баллах от 1 до 5 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

[This subsection should provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the **SRS**. This information may be provided by reference to the project Glossary.]

## Ссылки

|  |  |
| --- | --- |
| **Обозначение** | **Расшифровка** |
| [IEEE-830] | IEEE Std 830-1998 |

[This subsection should provide a complete list of all documents referenced elsewhere in the **SRS**. Each document should be identified by title, documentation number (if applicable), date, and publishing organization. Specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document.]

## Краткий обзор

Данный документ структурирован согласно [IEEE-830].

Раздел 2 содержит описание поставляемой системы и схему её использования в Организации. Раздел 3 содержит функциональные и нефункциональные требования, предъявляемые к системе и необходимые для её проектирования.

[This subsection should describe what the rest of the **SRS** contains and explain how the document is organized.]

# Общее описание

[This section of the **SRS** should describe the general factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides a background for those requirements, which are defined in detail in Section 3, and makes them easier to understand. Include such items as:

• product perspective

• product functions

• user characteristics

• constraints

• assumptions and dependencies

• requirements subsets]

## Описание изделия

Наименование разработки: “Система дистанционного обучения (далее "система"). Система, помогающая преподавателям в проверке работ студентов. Работа, загруженная студентом в систему, отправляется на проверку некоторым другим студентам, которые должны выставить ей оценку по пятибалльной системе. N –параметр системы, показывающий количество студентов, проверивших домашнее задание. На основе параметра N рассчитывается оценка, которую получит студент за свою работу. Оценка = среднее арифметическое из оценок, выставленных всеми студентами за эту работу. В случае большой разницы в выставленных оценках, работа должна быть отправлена на проверку преподавателю.

### Интерфейсы системы

Система должна иметь понятный и удобный интерфейс. Интерфейсы каждого типа пользователей описаны в пункте 3.8

### Интерфейсы пользователя

#### Класс пользователя 1 (“Преподаватель”)

#### Класс пользователя 2 (“Студент”)

### Интерфейсы аппаратных средств ЭВМ

Нет требований

### Интерфейсы программного обеспечения

Нет требований

### Интерфейсы коммуникаций

Нет требований

### Ограничения памяти

Нет требований

### Действия

#### Создание и выгрузка списка студентов в систему

#### Автоматическое создание учетных записей студентов в соответствии с выгруженным списком

#### Добавление студентом своей работы

#### Оценка студентом чужих работ

#### Оценка преподавателем работы студента

### Требования настройки рабочих мест

Нет требований

## Функции изделия

### Создать условия для дистанционной проверки домашних заданий

### Предоставить возможность загрузки домашних работ в систему

### Предоставить возможность распределить домашнее задание студента между другими студентами

### Предоставить возможность преподавателю и студенту выставлять оценки

### Предоставить возможность считать среднее арифметическое выставленных оценок

### Предоставить возможность перенаправлять работу преподавателю в случае большого разброса в баллах

## Характеристики пользователей

Пользователь (студент) будет иметь

-личный номер

-список выгруженных работ и оценка за каждую (если она уже выставлена)

## Ограничения

### Если разброс выставленных оценок в баллах слишком велик, работа считается спорной и отправляется на проверку преподавателю

### Студент не видит имя автора проверяемой работы

### Студент не может изменить выставленную оценку после закрытия проверяемого файла

## Предположения и зависимости

Хранит все данные (домашние задания и оценки, пользователи) в централизованном хранилище данных. Нет требований к конкретной системе хранения данных.

## Распределение требований

# Детальные требования

This section of the **SRS** should contain all the software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements. When using use-case modelling, these requirements are captured in the Use-Cases and the applicable supplementary specifications.]

## Функциональные требования

[This section describes the functional requirements of the system for those requirements which are expressed in the natural language style. For many applications, this may constitute the bulk of the **SRS** Package and thought should be given to the structure of this section. This section is typically structured by feature, but alternative structures may also be appropriate, for example, structure by user or by subsystem. Functional requirements may include feature sets, capabilities, and security.

Where application development tools, such as requirements tools, modelling tools, etc., are employed to capture the functionality, this section will refer to the availability of that data, indicating the location and name of the tool that is used to capture the data.]

### Класс пользователя 1 (“Преподаватель”)

#### Функциональное требование 1.1

#### Пользователь может видеть имя автора проверяемой работы

#### Функциональное требование 1.2

#### Оценка этого пользователя является приоритетной относительно других оценок и конечной

### Класс пользователя 2 (“Студент”)

#### Функциональное требование 2.1

#### Пользователь не может видеть имя автора проверяемой работы

#### Функциональное требование 2.2

Пользователь не имеет возможность выставить оценку, пока не будет прочитан весь файл

#### Функциональное требование 2.3

Пользователь имеет возможность поменять оценку до того, как он подтвердит свой выбор оценки и закроет файл

[The requirement description.]

## Надежность

[Requirements for reliability of the system should be specified here. Some suggestions follow:

• Availability—specify the percentage of time available ( xx.xx%), hours of use, maintenance access, degraded mode operations, etc.

• Mean Time Between Failures (MTBF) — this is usually specified in hours, but it could also be specified in terms of days, months or years.

• Mean Time To Repair (MTTR)—how long is the system allowed to be out of operation after it has failed?

• Accuracy—specify precision (resolution) and accuracy (by some known standard) that is required in the system’s output.

• Maximum Bugs or Defect Rate—usually expressed in terms of bugs per thousand of lines of code (bugs/KLOC) or bugs per function-point( bugs/function-point).

• Bugs or Defect Rate—categorized in terms of minor, significant, and critical bugs: the requirement(s) must define what is meant by a “critical” bug; for example, complete loss of data or a complete inability to use certain parts of the system’s functionality.]

### <Reliability Requirement One>

[The requirement description.]

## Производительность

[The system’s performance characteristics should be outlined in this section. Include specific response times. Where applicable, reference related Use Cases by name.

• response time for a transaction (average, maximum)

• throughput, for example, transactions per second

• capacity, for example, the number of customers or transactions the system can accommodate

• degradation modes (what is the acceptable mode of operation when the system has been degraded in some manner)

• resource utilization, such as memory, disk, communications, etc.

### <Performance Requirement One>

[The requirement description goes here.]

## Ремонтопригодность

[This section indicates any requirements that will enhance the maintainability of the system being built, including coding standards, naming conventions, class libraries, maintenance access, maintenance utilities.]

### <Maintainability Requirement One>

[The requirement description goes here.]

## Ограничения проекта

[This section should indicate any design constraints on the system being built. Design constraints represent design decisions that have been mandated and must be adhered to. Examples include software languages, software process requirements, prescribed use of developmental tools, architectural and design constraints, purchased components, class libraries, etc.]

### <Design Constraint One>

[The requirement description goes here.]

## Требования к пользовательской документации

[Describes the requirements, if any, for on-line user documentation, help systems, help about notices, etc.]

## Используемые приобретаемые компоненты

[This section describes any purchased components to be used with the system, any applicable licensing or usage restrictions, and any associated compatibility and interoperability or interface standards.]

## Интерфейсы

[This section defines the interfaces that must be supported by the application. It should contain adequate specificity, protocols, ports and logical addresses, etc. so that the software can be developed and verified against the interface requirements.]

### Интерфейс пользователя

Для студентов должна быть реализована возможность видеть в своем личном кабинете, в рамках конкретного курса, список лабораторных работ (домашних заданий). Студент имеет возможность войти в конкретную лабораторную работу и в рамках данной работы загрузить свое решение. Спустя некоторое время после загрузки своей работы студенту становится доступен дополнительный раздел, в котором находятся работы других студентов, ожидающие проверки именно этим студентом.

Интерфейс преподавателя должен поддерживать возможность загрузки списка студентов, выставления оценки студентам за их работу, а также возможность просматривания списка работ каждого студента и его оценки

[Describe the user interfaces that are to be implemented by the software.]

### Аппаратные интерфейсы

[This section defines any hardware interfaces that are to be supported by the software, including logical structure, physical addresses, expected behaviour, etc. ]

### Программные интерфейсы

[This section describes software interfaces to other components of the software system. These may be purchased components, components reused from another application or components being developed for subsystems outside of the scope of this **SRS** but with which this software application must interact.]

### Интерфейсы коммуникаций

[Describe any communications interfaces to other systems or devices such as local area networks, remote serial devices, etc.]

## Требования лицензирования

[Defines any licensing enforcement requirements or other usage restriction requirements that are to be exhibited by the software.]

## Применимые стандарты

[This section describes by reference any applicable standard and the specific sections of any such standards which apply to the system being described. For example, this could include legal, quality and regulatory standards, industry standards for usability, interoperability, internationalization, operating system compliance, safety, security, etc.]

# Индекс