**Software Requirements Specification**

**(Спецификация требований к продукту)**

#### **1.** **Introduction (Введение)**

##### **1.1** **Purpose**

Данный документ описывает спецификации требования к продукту http://tvigle.ru/ – сайт для онлайн просмотра кино бесплатно.

## 1.2 Scope (Область применения)

Проект относится к сфере сайтов, распространяющ

## 1.3 Definitions, Acronyms and Abbreviations (Определения и аббревиатуры)

## 1.4 References (Ссылки)

## 1.5 Overview (Обзор документа)

# **2.** **Overall Description (Общее описание)**

##### **2.1** **Product functions (Функционал продукта)**

## 2.2 User characteristics (Описание пользователей)

Основной контингент пользователей будет состоять из молодого поколения, возраст которого будет в промежутке 6-16 лет. В основном пользователи будут заходить на сайт через браузер по закладке в браузере или через поиск и сразу выбирать необходимую игру. После загрузки средняя сессия игрока будет составлять от 5 до 45 минут, а дальше в целях удержания внимания у пользователя всегда есть возможность подключиться к форуму или чату, где пользователь сможет выразить свое мнение об игре или пообщаться на другую тему с другими пользователями.

## 2.3 Assumptions and dependencies (Влияющие факторы и зависимости)

База данных. Расположение сервера, так как это влияет на правила распространения игр.

## 2.4 Сonstraints(Ограничения)

Четкое направление на бесплатные игры ограничивает функционал тем, что не будет системы оплаты, так как игры бесплатные.

# **3.** **Specific Requirements (Спецификация требований)**

## 3.1 Functionality (Функциональные требования)

### *3.1.1* *<Требование 1>*

Форум. У авторизованного пользователя должен быть доступ к форуму, где создаются отдельные темы и пользователи могут общаться. Приоритет задачи 8/10. Необходимая стабильность – Средняя. Затраты человеко-час – 5.

### *3.1.2* *<Требование 2>*

Чат. Дополнительный метод общения пользователей. Представляет собой отдельную страницу, где все пользователи в режиме реального времени могут общаться на разные темы. Приоритет задачи 5/10. Необходимая стабильность – Низкая. Затраты человеко-час – 6.

### *3.1.3* *<Требование 3>*

Регистрация / Авторизация. Метод регистрации и авторизации необходим для доступа к форуму, играм и чату, а также к системе оценок. Приоритет задачи 10/10. Необходимая стабильность – Высокая. Затраты человеко-час – 10.

### *3.1.4* *<Требование 4>*

Запуск и скачивание flash игр с сайта. Основная часть данного сервиса. Дает доступ к различным flash играм. Приоритет задачи 10/10. Необходимая стабильность – Высокая. Затраты человеко-час – 14.

### *3.1.5* *<Требование 5>*

Система оценок игр – Дополнительная функциональность призванная дать пользователям возможность оценивать игры по 5 звездочной системе. Приоритет задачи 2/10. Необходимая стабильность – Низкая. Затраты человека-час – 4.

## 3.2 Usability (Требования к удобству использования)

### *3.2.1* *<Требование 1>*

Должен быть доступен авторизованным пользователям и в случае того, что пользователь не авторизован происходит переадресация на страницу авторизации. Представляет из себя страницу со списком тем, где каждая тема состоит из названия и описания умещенного в 2 строки. Открывая тему страница состоит из подобия переписки, где каждый блок это логотип пользователя, псевдоним, ссылка на сообщение (если это ответ на сообщение) и текст самого сообщения, а также дата создания текста.

### *3.2.2* *<Требование 2>*

Должен быть доступен авторизованным пользователям и в случае того, что пользователь не авторизован происходит переадресация на страницу авторизации. Представляет из себя страницу в формате переписки, где отображаются логотипы пользователей, их псевдонимы и сообщения с датой отправки.

### *3.2.3* *<Требование 3>*

Авторизация состоит из вводных полей логина, пароля и кнопки авторизация, а регистрация добавляет к этим полям (доп поля для сайта).

### *3.2.4* *<Требование 4>*

Должен быть доступен авторизованным пользователям и в случае того, что пользователь не авторизован происходит переадресация на страницу авторизации. Страница состоит из сетки с играми, а именно сетка из логотипов. При наведение логотип затемняется и появляется название игры. При нажатии случается переход на саму игру и внизу под окном игры есть раздел с оценкой.

### *3.2.5* *<Требование 5>*

Должен быть доступен авторизованным пользователям и в случае того, что пользователь не авторизован происходит переадресация на страницу авторизации. Находится под окном с игрой и представляет из себя блок с 5 звездами. При нажатии на звезду запоминает выбранный рейтинг пользователя и изменяет общий рейтинг.

## 3.3 Reliability (Требования к надежности)

### *3.3.1* *<Требование 1>*

Ожидаемый процент ошибок при загрузке сайта должен быть не более 0.1% от всех заходов, однако сайт не рассчитан на большой единоразовый поток пользователей и ожидаемое время восстановление допускается до 10 минут.

### *3.3.2* *<Требование 2>*

Количество критических ошибок должно быть сведено к минимуму, однако допустимы в течении некоторого времени некоторые ошибки, которые не нарушают возможность запуска игр.

## 3.4 Performance (Требования к производительности) [нефункциональные]

### *3.4.1* *<Требование 1>*

Так как ожидаемая нагрузка от пользователей небольшая желаемая скорость загрузки игр в пределах 1.5-2 секунд, а максимальное не более 10 секунд для каждой игры.

### *3.4.2* *<Требование 2>*

Ожидается, что максимальное количество одновременно играющих пользователей не превысит 300 и поэтому ожидаемое использование ресурсов: процессор – 8 ядер, память – 16гб, дисковое пространство – 400гб. Этого должно хватить для критических случаев, когда резко количество одновременно играющих пользователей увеличится в 5 раз.

### *3.4.2* *<Требование 3 +>*

Из-за возможного увеличения активных пользователей система должна быть разработана таким образом, чтобы она могла спокойно масштабироваться горизонтально.

## 3.5 Design Constraints (Ограничения разработки)

### *3.5.1* *<Требование 1>*

В качестве языка программирования будет использоваться JavaScript в Frontend и он же в Backend (NodeJS). Выбор обуславливается возможностью использовать Frontend разработчиков в Backend, что снизит затраты на разработку.

### *3.5.2* *<Требование 2>*

Из-за простоты проекта и его незамысловатости будет использована V-model.

### *3.5.3* *<Требование 3>*

Инструменты разработки – Visual Studio Code, как редактор кода, так как его инструменты достаточны для разработки проекта и он бесплатный.

### *3.5.4* *<Требование 4>*

Сторонние библиотеки – (тут написаны библиотеки для frontend для работы с flash и возможно React, так как для удобства расширения он будет предпочтителен)

## 3.6 Interfaces (Интерфейсы)

### *3.6.1* *User Interfaces (Пользовательские интерфейсы)*

Основные страницы интерфейса: Главная страница, Страница форума, Страница регистрации, Страница авторизации, Страница со списком игр, Страница общего чата, Страницы для каждой темы, созданной в форуме, Страница игры, Страница чата для отдельной игры.

### *3.6.2* *Hardware Interfaces (Аппаратные интерфейсы)*

Один сервер с характеристиками: процессор – 8 ядер, оперативная память от 16гб, Постоянная память от 400гб. Наличие как минимум одного белого IP адреса.

### *3.6.3* *Software Interfaces (Программные интерфейсы)*

Сервисы будут обернуты в Docker контейнеры и будут общаться через открытые порты. В докер контейнерах будет сам сервис, база данных PostgreSQL и другие сервисы, такие как (Kafka?)

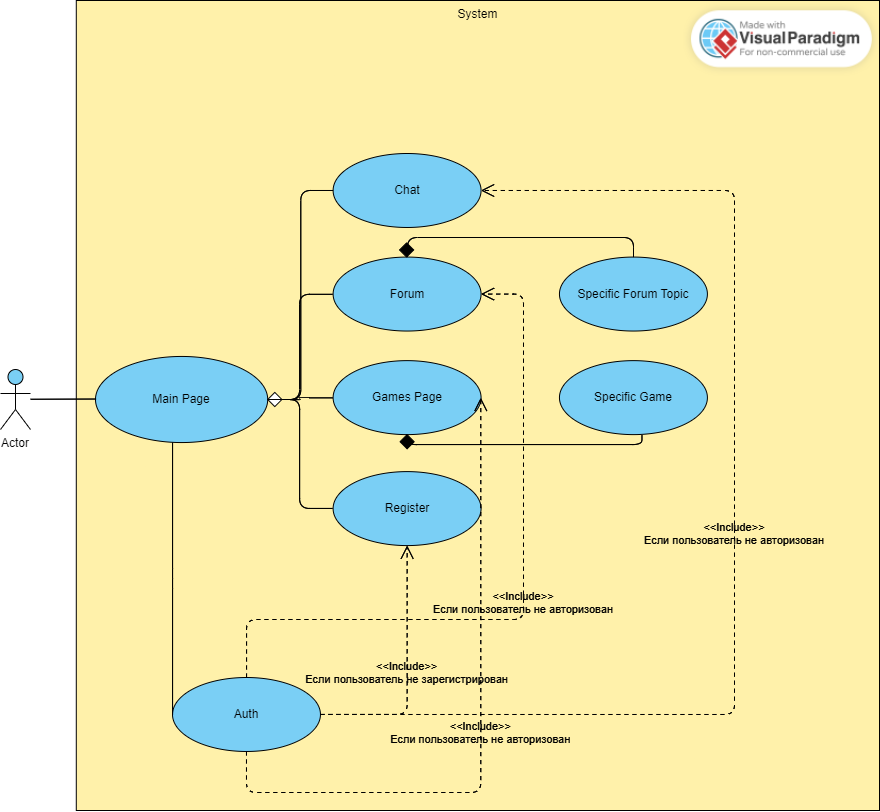
### *3.6.4* *Communications Interfaces (Сетевые интерфейсы)*

На сервере будут работать Docker контейнеры и основной трафик сетевых коммуникаций будет проходить внутри машины через открытые порты. Примером может быть обращение к базе данных PostgreSQL, которая будет открыта на порте 5432 и формат будет следующим: Request to localhost:5432 …

## 3.7 Licensing Requirements (Требования к лицензированию)

Данный продукт является коммерческим и не планируется раскрываться и поэтому лицензия будет Commercial.

## 4.0 Use-Case Диаграмма



https://mermaid.ink/img/pako:eNqdVV1r2zAU_StCUNiYExrHaWw9FNqN7WWFsdKXYSiardgitmQkeVka8t-nj8yxUiWs05N1dc65V_dc2ztY8JJABK-uwCxnRYOl_ERxJXCbM6CXjYAnSQTYuYhZH0CvIwy3BCDwqARl1fiw05wNF-XJ4X4s-YApe_6GK-Lryppv3r3XxF-clgHaZy761qcUNSnWd72quaAvWFHOTgQcjHeEXVL-TioqlbACfgK-WhHhJbjvlTqTRpCSClIoD2_ueQZtkhJxqTBPya8MH47IxVJMqwMVeEm-aC8Ddvxjb41rRkF-1bcJIroGbw3iUg0fa6z-21pDDovb4Z1MbkcjhwBluuu4UBJsqKod8Hhu0G7SkK6Bc0ncNoTz5maAj6Mhlm_qQPPCId7Rp4FjQsCEQnjb0wFqdg7lbheqhK4A4-o4WuUp4W9jTNfdmdeC4O0aWqwHSRf9ad-hMwJjqwRRvWBAcRsd3dTP8prmpj9APDbxDQ3wSWMfzPSDyow_aPQ1AGalnXcbc2RrwxuSDfiDf6bX2kTjHoxgS0SLaak_2_ZtyaGqic4EkX4ssVjnMGd7jdOq_HHLCoiU6EkE-67Eihy-7xCtcCN1tMPsB-feHqId_A3RJIun8-wmXSTJ3K0IbiGaLbPpfHGdpml2s0yyeLGP4ItVmE3jRRYnyySO41k6j-M0gtoGxcWD-83Yv83-DzUfBXE?type=png

На этой диаграмме у нас есть семь классов, представляющих сущности в системе: User, Main\_Page, Forum, Registration, Authorization, Game\_Page и Chat. Стрелки указывают на взаимодействие между классами.

Пользователь взаимодействует с Main\_Page, которая отображает параметры для выбора пользователем. Классы Forum, Registration, Authorization, Game\_Page и Chat представляют различные варианты, из которых может выбирать пользователь. Классы Forum, Registration, Game\_Page и Chat перед выполнением своих задач проверяют, авторизован ли пользователь. Если пользователь не авторизован, он перенаправляется в класс Authorization. Класс Registration предлагает пользователю кнопку авторизации и перенаправляет его в класс Authorization, если он ее нажимает. Класс Authorization авторизует пользователя и перенаправляет его обратно на страницу Main\_Page.

## 5W.0 Вывод

В ходе работы мы ознакомились с методологией RUP и структурой SRS документа, была создана UseCase-диаграмма и составлен список требований к сайту, оформленный в виде документа SRS.