Система автоматизированной генерации расписания

План разработки программного обеспечения

Версия 1.0

**История Ревизий**

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Версия** | **Описание** | **Автор** |
| 20.04.2020 | 1.0 | Первоначальная версия | Допира Валерия (Руководитель проекта) |

**1 Введение**

* 1. **Цель**

Цель документа – определить действия по разработке с точки зрения этапов и итераций, необходимых для реализации системы доставки оборудования на мероприятия.

* 1. **Объём**

Этот План разработки программного обеспечения описывает общий план, который будет использоваться командой для разработки системы.

* 1. **Определения, сокращения и аббревиатуры**

*См. Глоссарий*

**2 Обзор проекта**

**2.1 Цели проекта**

В этом проекте будет реализована система для составления расписания в образовательных учреждениях.

Данное приложение создается со следующими целями:

* Обеспечение минимальных усилий пользователя при составлении расписания.
* Создание удобного интерфейса.
* Обеспечение быстрого доступа к данным, хранящимся в программе.
* Обеспечение возможности добавления, изменения и удаления данных в программе.
* Отображение расписание в виде многодольного графа.
* Обеспечение сериализации данных при работе с программой.

**2.2 Допущения и ограничения**

Система должна быть закончена к 31 мая 2020 года.

**2.3 Эволюция плана проекта.**

Этот план будет обновляться по ходу разработки продукта. Целевые даты окончания каждого этапа показаны ниже (см. табл.1).

Таблица 1 - Даты окончания каждого этапа

|  |  |
| --- | --- |
| ***Название этапа*** | ***Дата окончания*** |
| Начало (Inception) | 03.05.2020 |
| Уточнение (Elaboration) | 15.05.2020 |
| Построение (Construction) | 23.05.2020 |
| Внедрение (Transition) | 31.05.2020 |

**3.Роли и обязанности**

Распределение ролей и обязанностей исполнителям представлено ниже (см. табл.2).

Таблица 2 – Роли и обязанности

|  |  |  |
| --- | --- | --- |
| ***ФИО студента*** | ***Роль*** | ***Обязанность*** |
| Допира Валерия,  Бочкарев Иван,  Ильянов Вячеслав | Аналитик | Устанавливает требования к разрабатываемой системе путем определения необходимой функциональности |
| Разработчик требований | Формирует детальные требования к разрабатываемой системе |
| Допира Валерия | Проектировщик | Проектирование компонентов системы с учётом требований, заданных проектом |
| Допира Валерия | Разработчик архитектуры | Управление разработкой программной архитектуры системы, что включает продвижение и поддержку ключевых технических решений, задающих рамки реализации проекта. |
| Бочкарев Иван, Вячеслав Ильянов,  Павлов Данила | Реализатор | Разработка программных компонентов и проверка их функциональности после интеграции в подсистемы |
| Ильянов Вячеслав,  Допира Валерия, | Испытатель | Выполнение тестирования продукта и описание исхода тестирования. |
| Допира Валерия,  Бочкарев Иван,  Ильянов Вячеслав | Разработчик тестов | Определение общей стратегии тестирования и контроль успешности ее реализации. |
| Ильянов Вячеслав,  Павлов Данила | Руководитель тестирования | Защита качества тестирования, планирование ресурсов и управление ими, а также решение проблем, препятствующих процессу тестирования. |
| Допира Валерия | Руководитель проекта | Планирование затрат ресурсов, распределение ресурсов, выделение приоритетов, направление усилий коллектива. |
| Спицын Александр Валентинович | Проверяющий | Оценивание планирования проекта и ценности рабочих продуктов на границах важнейших этапов жизненного цикла проекта. |

**4. Процесс управления**

**4.1 Фазовый план**

Разработка системы будет производиться с использованием поэтапного подхода. Фазы и относительная временная шкала показаны ниже (см. табл.3)

Таблица 3 – Фазы и относительная временная шкала

|  |  |  |
| --- | --- | --- |
| ***Фаза*** | ***Дата начала*** | ***Дата окончания*** |
| Начало | 1 неделя | 6 неделя |
| Уточнение | 7 неделя | 10 неделя |
| Построение | 11 неделя | 12 неделя |
| Внедрение | 13 неделя | 14 неделя |

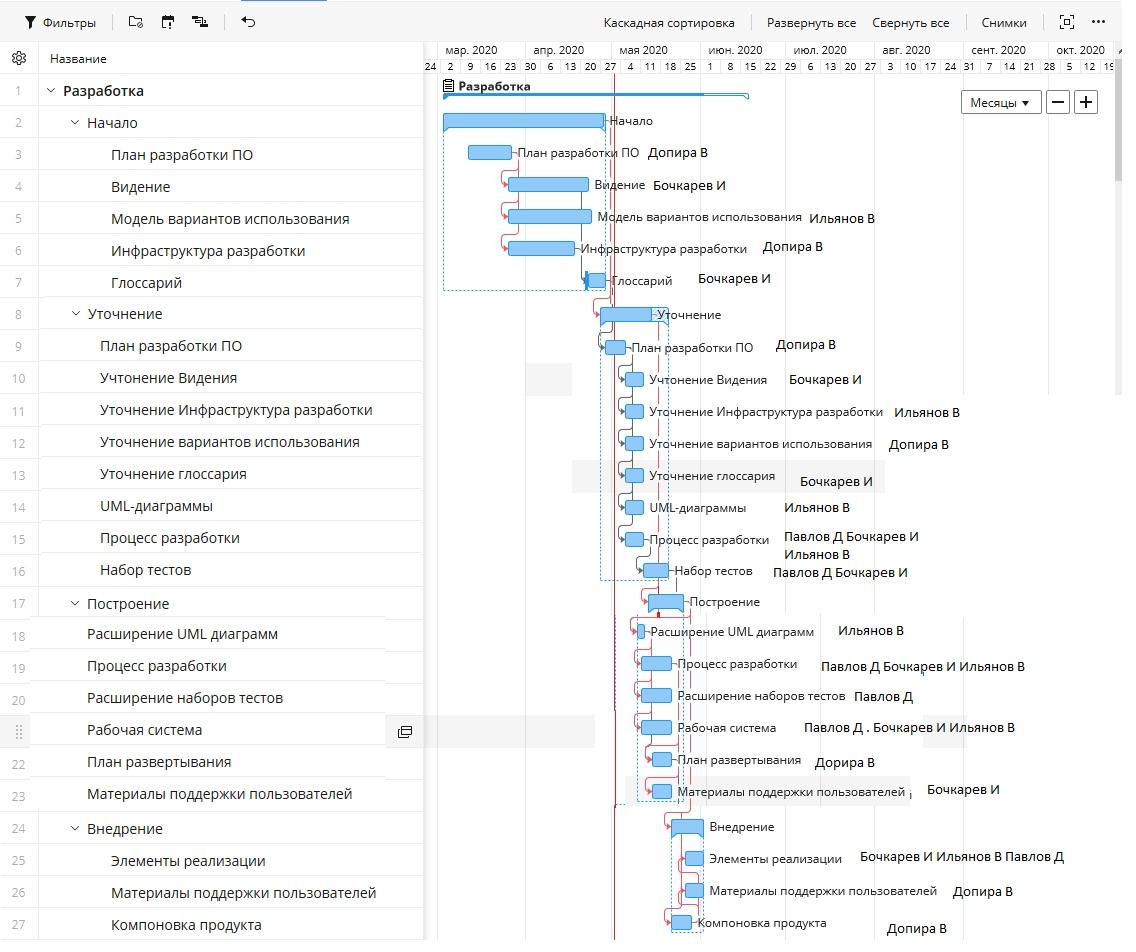
Рабочие продукты получаемые после завершения определенного этапа представлены ниже (см. табл.4)

Таблица 4 – Рабочие продукты каждого этапа

|  |  |  |
| --- | --- | --- |
| ***Фаза*** | ***Описание*** | ***Рабочие продукты*** |
| Начало | Определение содержания проекта, разработка основных сценариев | План разработки ПО  Видение  Глоссарий  Модель вариантов использования  Инфраструктура разработки |
| Уточнение | Создание стабильной формальной архитектуры для выполнения разработки ПО | План разработки ПО  Видение  Глоссарий  Модель вариантов использования  Инфраструктура разработки  Процесс разработки  Диаграмма вариантов использования, Диаграмма классов, Диаграмма пакетов, Диаграмма компонентов, Диаграмма последовательностей, Диаграмма состояний  Тестовый набор |
| Построение | Завершение разработки системы в соответствии с базовой архитектурой | Рабочая система  Расширены диаграммы:  Диаграмма вариантов использования, Диаграмма классов, Диаграмма пакетов, Диаграмма компонентов, Диаграмма последовательностей, Диаграмма состояний  Процесс разработки  Материалы поддержки пользователей  План развертывания  Тестовый набор |
| Внедрение | Обеспечение готовности программного обеспечения к представлению пользователям | Компоновка продукта  Материалы поддержки пользователей  Элементы реализации |

**4.2 Диаграмма Ганта**

Для иллюстрации плана, графика работ и занятости членов команды, работающих над проектом, удобно использовать диаграмму Ганта. Диаграмма Ганта со списком рабочих продуктов и исполнителями представлена ниже (см. рис.1).



*Рис. 1 Диаграмма Ганта*

**4.3 График проекта**

График сдачи каждой задачи проекта представлен ниже (см. табл.5).

Таблица 5 – График проекта

|  |  |  |  |
| --- | --- | --- | --- |
| ***Фаза*** | ***Задача*** | ***Окончание работы*** | ***Исполнители*** |
| Начало | План разработки ПО | 26.03.2020 | Допира Валерия |
| Видение | 22.04.2020 | Бочкарев Иван |
| Модель вариантов использования | 22.04.2020 | Павлов Данила |
| Инфраструктура разработки | 22.04.2020 | Бочкарев Иван |
| Глоссарий | 27.04.2020 | Ильянов Вячеслав |
| Уточнение | План разработки ПО | 30.04.2020 | Допира Валерия |
| Видение | 04.05.2020 | Бочкарев Иван |
| Модель вариантов использования | 04.05.2020 | Павлов Данила |
| Инфраструктура разработки | 04.05.2020 | Бочкарев Иван |
| Глоссарий | 04.05.2020 | Ильянов Вячеслав |
| UML диаграммы | 04.05.2020 | Допира Валерия, Бочкарев Иван, Павлов Данила, Ильянов Вячеслав |
| Набор тестов | 14.05.2020 | Допира Валерия, Бочкарев Иван |
| Построение | UML диаграммы | 15.05.2020 | Допира Валерия, Бочкарев Иван, Павлов Данила, Ильянов Вячеслав |
| Набор тестов | 22.05.2020 | Допира Валерия, Бочкарев Иван |
| Материалы поддержки пользователей | 22.05.2020 | Ильянов Вячеслав |
| Рабочая система | 22.05.2020 | Павлов Данила |
| План развёртывания | 22.05.2020 | Допира Валерия |
| Внедрение | Элементы реализации | 27.05.2020 | Бочкарев Иван |
| Компоновка продукта | 29.05.2020 | Павлов Данила |
| Материалы поддержки пользователей | 29.05.2020 | Ильянов Вячеслав, Допира Валерия |

**4.4 Загрузка ресурсов по дням**

Количество дней выделенных на каждый ресурс представлено ниже (см. табл. 6).

Таблица 6 – Загрузка ресурсов по дням

|  |  |
| --- | --- |
| ***Ресурсы*** | ***Количество дней*** |
| План разработки ПО | 14 |
| Видение | 35 |
| Модель вариантов использования | 35 |
| Инфраструктура разработки | 12 |
| Глоссарий | 6 |
| UML диаграммы | 6 |
| Набор тестов | 3 |
| Материалы поддержки пользователей | 7 |
| Разработка продукта | 25 |

**4.5 Нагрузка исполнителей**

Количество часов в неделю, которые исполнители тратят на проект описано ниже (см. табл. 7). Нагрузка распределена равномерно на исполнителей на протяжении всего проекта. Роли распределены в соответствии с планом разработки программного обеспечения. Исполнитель может распределять время в течение недели, как ему удобно: работая каждый день по несколько часов или полноценный рабочий день. Синхронизация работы проходит раз в неделю или чаще, если это необходимо.

Таблица 7 – Нагрузка на исполнителей

|  |  |
| --- | --- |
| ***Исполнитель*** | ***Количество часов в неделю*** |
| Допира Валерия | 8 |
| Бочкарев Иван | 8 |
| Павлов Данила | 8 |
| Ильянов Вячеслав | 8 |

Система автоматизированной генерации расписания

Видение

Версия 1.1

**История Ревизий**

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Версия** | **Описание** | **Автор** |
| 18.04.2020 | 1.0 | Первоначальная версия | Допира  Валерия  (Аналитик) |
| 01.05.2020 | 1.1 | Дополненная версия | Бочкарев Иван  (Аналитик) |

**1 Введение**

**1.1 Цель**

Цель документа – определить требования к разрабатываемому продукту с точки зрения потребностей пользователей.

**1.2 Область применения**

Настоящий документ о видении применяется к приложению, которое предназначено для составления расписания образовательного процесса в учебном заведении.

Одна из основных составляющих учебного процесса – это расписание занятий. Очевидно, что качество составленного расписания сильно влияет на такие факторы как успеваемость студентов и школьников, их мотивацию, а в некоторых случаях и на творческую составляющую. Таким же образом расписание касается и преподавателей, так как их распорядок дня, а соответственно и продуктивность, также зависит от грамотно составленного расписания.

Из вышесказанного можно сделать вывод о том, что расписание имеет большое значение в эффективности учебного процесса, а, следовательно, в качестве образования Высших учебных заведения и Школах.

Представляемый программный продукт призван:

1. Облегчить ручной и умственный труд сотрудников при составлении учебного расписания. От сотрудника требуется только ввести необходимые данные и проверить корректность итогового расписания.
2. Сгенерировать максимально удобное и оптимальное расписание как для студентов и школьников, так и для преподавателей.
3. Удобный интерфейс для взаимодействия программы с пользователем.

Конечно, не только учебные заведения заинтересованы в составлении оптимального расписания, но и крупные компании и предприятия, где нужно составить эффективный распорядок дня как для определенных людей, так и для групп людей.

В наши дни это все более актуально, так как динамика жизни становится все интенсивнее и интенсивнее.

**1.3 Определения, сокращения и аббревиатуры**

*См. Глоссарий*

**2 Позиционирование**

**2.1 Необходимость в продукте**

На данный момент в образовательных учреждениях остро стоит вопрос о составлении учебных расписании. Это связано с тем, что итоговое расписание зачастую является не эффективными, а кроме этого требует больших трудозатрат пропорциональных масштабам учебного учреждения.

В данной области лучше всего подходит создание программного обеспечения по автоматизированному составлению оптимального расписания.

**2.2 Постановка проблемы**

|  |  |
| --- | --- |
| Проблема | Повышение сложности составления эффективного учебного расписания |
| Кого затрагивает проблема | Составители расписаний в школах/Вузах, преподаватели, школьники и студенты. |
| Как проблема появилась | Увеличение временных затрат на составление расписания. Нерациональное использование времени учащихся. Высокая стоимость внесения изменений в расписание. |
| Как поможет успешное решение | Увеличится эффективность учебного процесса, увеличится продуктивность преподавателей, студентов и школьников, появится хорошо распланированный порядок дня. |

**3. Описание заинтересованных сторон и пользователей**

В данном разделе описываются пользователи системы составления учебного расписания. Система подразумевает только один тип пользователей – это составители расписания.

**3.1 Состояние рынка**

Предоставляемый продукт является системой автоматизированного составления расписания.

Аналоги данного ПО на данный момент либо слабо развиты, либо предоставляют только косвенный функционал, к примеру, такой как простое представление расписания в электронном виде.

В сравнении с аналогами наш продукт имеет много схожих возможностей, а именно возможность постоянно пополнять данные по учебным дисциплинам, учебным группам, учебным помещениям, учебным неделям, времени начала лекций.

Но отличительной чертой, а также особенностью нашего продукта, является наличие уникального функционала по генерации оптимального расписания. В этом аспекте продукт сильно выигрывает.

**3.2 Описание заинтересованных лиц**

|  |  |
| --- | --- |
| ***Заинтересованное лицо*** | ***Роль*** |
| IT Executive | Следит за ходом разработки проекта |
| Пользователи системы | В качестве пользователя системой выступает человек отвечающий за составление расписания. Пользователь имеет возможность добавлять и удалять элементы расписания (дисциплина, время, помещение, группы учащихся) |
| Заказчик | Представляет интересы образовательной организации, заинтересованной нашем продукте. |

**3.3 Описание пользователей**

|  |  |
| --- | --- |
| ***Имя*** | ***Описание*** |
| Пользователи системы | Редактирование и просмотр данных: учебные дисциплины, учебные группы, учебные помещения, учебные недели, время начала лекций, на основе которых генерируется расписание, которое можно просматривать в виде таблиц или многодольного графа |

**3.4 Окружение пользователя**  
 Для нормальной работы с ПО пользователю достаточно иметь обычный стационарный компьютер с установленной системой по генерации расписания.

Отличительной чертой является то, что системе не требуется выход в интернет, что означает программой можно пользоваться когда и где угодно.

**3.5 Ключевые потребности заинтересованных лиц**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Потребность*** | ***Приоритет*** | ***Текущее решение*** | ***Предлагаемое решение*** |
| Автоматизированное  составление расписания | Высокий | Многие пользователи вынуждены составлять расписание вручную | Система предлагает на основе загружаемых данных генерировать учебное расписание |

**3.6 Альтернативы и конкуренты**

Конкурентами являются всевозможные приложения для составления расписания доступные в сети интернет. Но как говорилось выше функционал таких приложение ограничен визуализацией расписания.

Альтернативным решением будет передать задачу составления расписания другой организации, что является нецелесообразным.

**4 Обзор продукта**

В данном разделе представлен общий обзор возможностей системы автоматизированного составления учебного расписания.

**4.1 Перспектива продукта**

Система будет построена по принципу монолитной архитектуры, которое означает, что данное приложение будет представлять большой связанный модуль, где все компоненты спроектированы так, чтобы работать вместе с друг с другом, общая память и ресурсы.

Приложение будет работать в автономном режиме без подключения к интернету, что облегчит работу пользователям в местах без доступа к интернету.  
 Система генерирования расписания может стать конкурентоспособной за счет функциональности и расчета оптимальных вариантов расписаний.

Конечно, система не ограничится текущим функционалом и в будущем планируется дополнить систему клиент-серверной технологией для обеспечения многопользовательского режима работы системы, а также централизации хранения данных. Это позволит скачать актуальное расписание на локальный компьютер, после редактирования отправить копию на сервер.

Также клиент-серверная архитектура позволяет генерировать расписание используя вычислительные мощности серверного компьютера, который как правило на порядок мощнее. Это ускорит процесс генерации для больших учебных заведений, большим объемом данных.

В следующих поколениях системы возможно добавление дополнительного функционала. Такого как составление расписания не на неделю, а на семестр, с учетом всех праздничных дней и учебной сессии.

Потенциально возможно создание мобильного приложения для студентов и школьников с расписание на неделю.

**4.2 Набор основных возможностей**

В таблице в этом разделе указаны основные возможности системы генерирования расписания с точки зрения преимуществ и возможностей.

|  |  |
| --- | --- |
| ***Преимущество для пользователей*** | ***Поддерживающие функции*** |
| Представление расписания в виде электронной таблицы | Система выводит составленное расписание в виде электронной таблицы |
| Представление расписания в виде многодольного графа | Система выводит составленное расписание в виде многодольного графа |
| Фильтрация расписания, представленное в виде многодольного графа | Система позволяет отфильтровать многодольный граф по: учебной дисциплине, времени начала занятий, учебным помещениям, учебным группам |
| Сохранение расписания в файл и загрузка расписания из файла | Система позволяет сохранять исходные и выходные данные в файл, который потом можно будет загрузить при новом сеансе работы с системой. |
| Легкость освоения продукта | Отсутствие авторизации и интуитивно понятный интерфейс |

**4.3 Предположения и зависимости**

Следующие предположения и зависимости относятся к возможностям системы генерации расписания, изложенным в данном документе:

* Система зависит от корректных исходных данных, загружаемых пользователем;
* Предполагается, что сгенерированное расписание системой полностью удовлетворит потребности пользователя;
* Успешные продажи системы автоматизированной генерации расписания зависят от правильной рекламы и раскрутки, а также качества самого продукта.

**4.4 Стоимость и цена продукта**

Затраты на разработку системы не должны превышать 200 000 рублей. Ожидается, что конечный продукт будет предложен заказчикам с условием оплаты – 10% от одного заказа.

**5 Характеристики продукта**

Этот раздел определяет и описывает функции системы автоматизированной генерации расписания.

Функциональные возможности – это высокоуровневые возможности системы, которые необходимы для предоставления преимуществ пользователям.

**5.1 Вход в систему**

Пользователь системы вход в системы без ввода логина и пароля.

**5.2 Редактирование данных**

В системе должны иметься отдельные таблицы для добавления, удаления, изменения следующих данных: учебные дисциплины, учебные группы, учебные помещения, время начала лекций, чтобы упростить процесс ввода данных.

**5.3 Поиск расписания**

Система должна позволять просматривать расписание в виде электронной таблицы и многодольного графа.

**5.3 Сериализация состояния системы**

Система должна позволять пользователям сохранять загруженные данные, сгенерированное расписание, для повторного использования.

**5.4 Фильтрация многодольного графа**

Система должна позволять пользователю фильтровать многодольный граф по: учебной дисциплине, дню, преподавателю и помещению

**6 Ограничения для продукта**

**6.1 Требования к качеству**

В этом разделе определяются диапазоны качества для доступности, надежности, удобства использования и аналогичных характеристик для системы доставки оборудования.

*Доступность.* Доступность ограничивается только работоспособностью компьютера, на котором она запущена.

*Удобство использования.* Система должна быть простой в использовании, с интуитивно понятным интерфейсом.

*Удобство обслуживания.* Система должна быть разработана для простоты обслуживания. Входные данные должны быть модифицируемыми.

*Совместимость с устройствами.* Приложение должно работать на платформах (Windows, Linux).

**6.2 Системные требования**

Система должна взаимодействовать с файлами формата json при сериализации.

Система не должна занимать более 50 мб на жестком диске.

Клиентский компонент системы не должен требовать более 128 МБ ОЗУ.

**6.3 Требования к производительности**

Время ответа не должно превышать 0,1 секунды, т.е. пользователь должен воспринимать работу системы как «мгновенную».

Пользователь не должен ощущать задержки при работе, каждый его запрос должен быть обработан в кратчайшие сроки.

Система автоматизированной генерации расписания

Модель вариантов использования

Версия 1.1

**История Ревизий**

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Версия** | **Описание** | **Автор** |
| 12.03.20 | 1.0 | Первоначальная версия | Ильянов В |
| 29.04.20 | 1.1 | Дополненная версия | Ильянов В |

1. **Введение**

**1.1 Цель**

Этот артефакт представляет собой модель намеченных функций системы и ее среды. Применяется в качестве входных данных для дальнейшей деятельности: анализа, проектирования, реализации и тестирования. Модель вариантов использования состоит из вариантов использования и субъектов. Каждый вариант использования в модели описан, шаг за шагом показывая, как система взаимодействует с субъектами, и что система выполняет в данном варианте использования. Модель вариантов использования служит связующей средой по вопросам функциональности системы. Эта модель позволяет:

* Заказчикам и пользователям – убедиться, что система будет такой, как они ожидают.
* Разработчикам системы – создать то, что ожидается.

Потенциальные пользователи используют модель вариантов использования для лучшего понимания системы.

Данная модель разработана для составления расписания, удовлетворяющий заданным требованиям.

**Определения, аббревиатуры и сокращения**

*См. Глоссарий.*

**Рекомендации**

Нет.

**Каталог Акторов (действующих лиц)**

|  |  |
| --- | --- |
| ***Имя*** | ***Описание*** |
| Пользователь | Физическое лицо, составляющее расписание, используя уже готовые данные или заполняющих их сам |
| Приложение | Приложение, позволяющее по введенным данным составить расписание |

1. **Варианты использования**

В данном разделе предложены различные варианты взаимодействия с приложением. Данный набор сценариев описывает наиболее важные функции и охватывает значительное число элементов архитектуры.

**2.1 Основной поток: составление расписания**

1. Пользователь загружает данные с файла, содержащий информацию по учебным дисциплинам, группам, помещениям, времени начале занятий
2. После загрузки данных пользователь нажимает на кнопку “Составить расписание”
3. Приложение по полученным данным выстраивает расписание в виде электронной таблице и многодольного графа
4. Далее пользователь может посмотреть на полученную электронную таблицу и граф с возможностью фильтрации, переключая вкладки “Расписание” и “Граф” соответственно

Альтернативные потоки:

1. Добавление в таблицу данных
   1. 1.Пользователь открывает таблицу, где расположена таблица с данными по учебным дисциплинам, учебным группам, учебным помещениям, время начала занятий.
   2. 2. Пользователь нажимает кнопку “добавить” с одной из колонок
   3. 3. Приложение вводит данные в таблицу
2. Удаление в таблице данных
   1. 1. Пользователь открывает таблицу, где расположена таблица с данными по учебным дисциплинам, учебным группам, учебным помещениям, время начала занятий.
   2. 2. Пользователь выбирает в таблице ячейку и нажимает кнопку “Удалить”
   3. 3. Приложение удаляет с таблицы данные, выбранные пользователем
3. Изменение в таблице данных
   1. 1. Пользователь открывает таблицу, где расположена таблица с данными по учебным дисциплинам, учебным группам, учебным помещениям, время начала занятий.
   2. 2. Пользователь находит нужную ячейку и нажимает на кнопку “Изменить”
   3. 3. Пользователь переписывает информацию в окне, которое вывело приложение
   4. 4. Приложение меняет информацию в ячейке
4. Некорректные данные в таблице данных
   1. 1.Пользователь вводит данные по учебным дисциплинам, учебным группам, учебным помещениям, времени начала занятий.
   2. 2. Программа выдает сообщение об ошибке и не дает занести данные
   3. 3. Пользователь продолжает работу

**3. Особые требования**

Будут определены в ходе следующих итераций.

**3.1. Предварительные условия:**

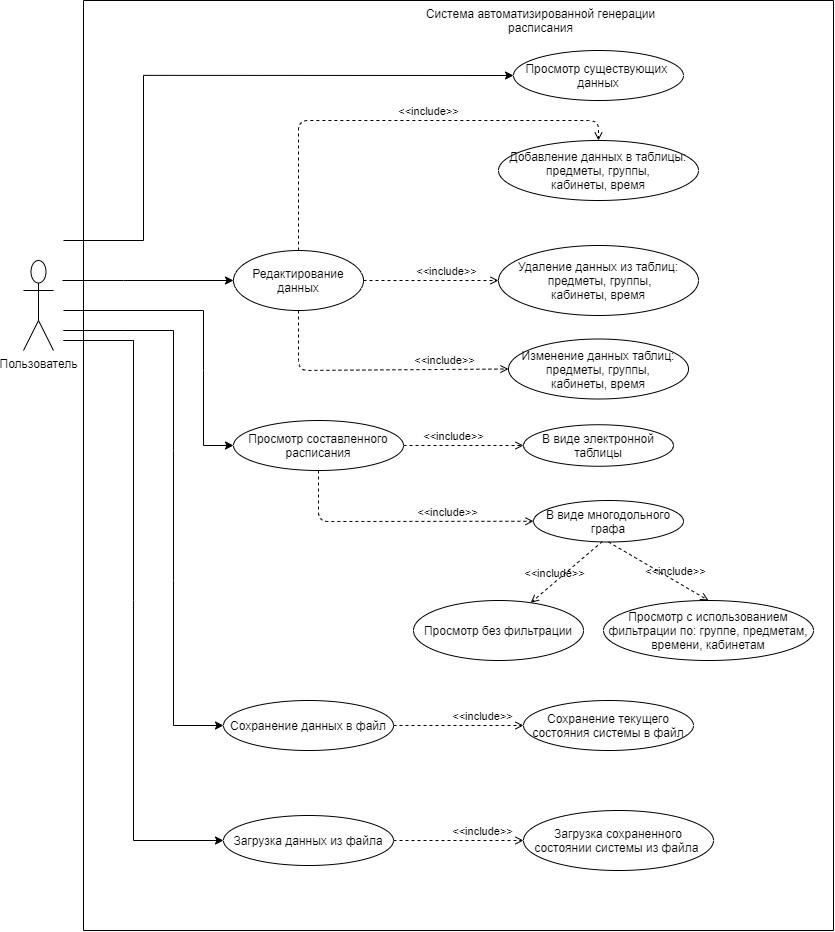
У пользователя должны иметься исходные данные по: учебным дисциплинам, учебным группам, учебным помещениям, времени начала занятий для загрузки их в систему, для составления учебного расписания.

* 1. **Постусловия**

Будут определены в ходе следующих итераций.

* 1. **Диаграмма варианта использования**

Пользователь: при первой работе с системой, необходимо добавить данные в таблицы: предметы, группы, кабинеты, время на основе которых система генерирует учебное расписание. Составленное расписание представляется в виде электронной таблицы и многодольного графа. Представление многодольного графа можно отфильтровать по: группе, предметам, времени, кабинетам. После окончания работы с системой, пользователь может сохранить полученные данные в файл, который можно использовать при следующем входе в систему (рис. 1).



*Рис. 1 Диаграмма вариантов использования*

Система автоматизированной генерации расписания

Инфраструктура разработки

Версия 1.0

**История Ревизий**

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Версия** | **Описание** | **Автор** |
| 25.04.2020 | 1.0 | Первоначальная версия | Ильянов Вячеслав  (Аналитик) |

**Введение**

Инфраструктура разработки – этот рабочий документ описывает аппаратное и программное обеспечение, такое как компьютеры и операционные системы, на которых реализованы различные инструменты. Стандартная инфраструктура разработки существует для того, чтобы определить условия для разработки.

Налаживание необходимой инфраструктуры разработки можно представить в следующих этапах:

1. **Выбор и получение инструментов**

Цель этой задачи:

* Выбор инструментов, наилучшим образом подходящих для разработки проекта.
* Получение необходимых инструментов.

Для эффективного выполнения многих этапов процесса требуется поддержка соответствующих инструментов. Инструменты следует выбирать с учетом конкретных потребностей. Особое внимание уделяется отдельным задачам и рабочим продуктам, необходимым для процесса.

1. **Определение требований и ограничений для инструментов**

На это влияют следующие факторы:

**Процесс разработки**. В первую очередь определяются сферы, которые будут затронуты в процессе разработки. В частности, могут понадобятся инструменты для планирования рабочего процесса, инструменты для оценки эффективности процесса разработки, инструменты для визуализации процесса разработки.

В качестве инструмента планирования и оценки эффективности рабочего процесса был выбран веб сервис компании Яндекс «Трекер».

В качестве инструмента моделирования был выбран веб сервис Draw.io.

* **Платформы разработки**. Разработка ведется на операционных системах Windows и Linux. Целевой платформой также являются данные операционные системы.
* **Применяемые языки программирования**. Разработка системы ведется на объектно-ориентированном языке «с++».
* **Библиотеки и Фреймворки.** Для более эффективной разработки система написано с использованием Фреймворка Qt. Данный Фреймворк имеет большой набор виджете, который облегчает разработку программного интерфейса.
* **Среда разработки.**  В качестве среды разработки была взята IDE Qt Creator 5.13. Это кроссплатформенная свободная среда для разработки на С++, включающая в себя графический интерфейс отладчика и визуальные средства разработки интерфейса.
* **Средство контроля версий.** Был выбран доступный веб-инструмент GitHub, предоставляющий систему управления репозиториями кода, системой отслеживания ошибок, отслеживания заданий, системой уведомлений по группам, ссылки и доступ к конференциям из задач.

1. **Настройка инструментов**

Цель этой задачи:

* Установка инструментов.
* Настройка инструментов.

Большинство инструментов разработки программного обеспечения поддерживают совместную работу пользователей с возможностью одновременного обращения к информации из общего хранилища, такие как Github, draw.io.

Ряд инструментов: компиляторы, отладчики, редакторы и графические средства можно просто установить на компьютеры. Как было сделано с инструментов «Qt». Единственное что потребуется это активировать их. Для этого потребуется создать учетную запись в соответствующем портале и получить ключ активации или бесплатную версию инструмента.

Большинство инструментов можно не конфигурировать, так как начальные настройки уже достаточны для нормальной работы.

1. **Проверка результатов установки и конфигурации инструментов**

Цель этой задачи – убедиться, что инструменты подходят для разработки системы.

Данная процедура необходима, так как иначе процесс разработки может быть застопорен из-за непроверенного функциональности инструмента, оказавшегося неподходящим для решения поставленной на него проблемы.

Для каждого инструмента выполнили при необходимости регистрацию, запуск и тестовый сценарий, проверив тем самым правильность установки и настройки.

Тестовый сценарий для каждого инструмента является уникальным и как правило предоставляется компанией, разработавшей данный инструмент.

Система автоматизированной генерации расписания

Глоссарий

Версия 1.0

**История Ревизий**

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Версия** | **Описание** | **Автор** |
| 20.04.2020 | 1.0 | Первоначальная версия | Бочкарев Иван (Аналитик) |

**1 Введение**

* 1. **Цель**

Цель документа – глоссарий содержит все рабочие определения для всех классов системы. Этот глоссарий будет расширяться на протяжении всей жизни проекта.

**1.2 Объем**

В этом глоссарии рассматриваются все термины, которые имеют конкретное значение для этого проекта. Акторы (действующие лица) здесь не перечислены, так как они более полно описаны в модели вариантов использования.

**2. Определения, сокращения и аббревиатуры**

*Артефакт (синоним. Рабочий продукт, Ресурс) –* некоторые продукты проекта, порождаемые или используемые в нем при работе над окончательным продуктом.

*Интерфейс* – граница между двумя функциональными объектами, требования к которой определяются стандартом; совокупность средств, методов и правил [взаимодействия](https://ru.wikipedia.org/wiki/%D0%92%D0%B7%D0%B0%D0%B8%D0%BC%D0%BE%D0%B4%D0%B5%D0%B9%D1%81%D1%82%D0%B2%D0%B8%D0%B5) (управления, контроля и т. д.) между элементами [системы](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0).

*Персональный компьютер* – настольная микро-ЭВМ, имеющая эксплуатационные характеристики бытового прибора и универсальные функциональные возможности.

*Прикладная программа, или приложение* – программа, предназначенная для выполнения определенных задач и рассчитанная на непосредственное взаимодействие с пользователем.

*Прецедент –* последовательность действий, выполняемых системой для получения наблюдаемого результата.

*Расписание занятий* - документ, определяющий педагогически целесообразную последовательность учебных занятий в образовательном учреждении на каждый день учебной недели и конкретизирующий таким образом учебный план.

*Учебная дисциплина* - система знаний, умений и навыков, отобранных из определенной отрасли науки, техники, искусства, производственной деятельности для изучения в образовательном учреждении.

*Преподаватель* - работник высших, средних специальных и профессионально-технических учебных заведений, ведущий какой-либо предмет и воспитательную работу.

*Учебная группа* - определенное число лиц с примерно одинаковым

уровнем подготовки, изучающих одно и то же в одно и то же время под руководством одних и тех же преподавателей на протяжении одинакового для всех периода; обособленная часть контингента образовательного учреждения, являющаяся для ее членов первичным коллективом.

*Учебная неделя* - дни, по которым проходят занятия по определенным дисциплинам.

*Учебное помещение* - специально оборудованное помещение, для обучения студентов.

Время начала лекций - время в которое проходят занятия по учебным дисциплинам.

*Семестр* - в высших учебных заведениях: полугодие учебного года: различают осеннее и весеннее.

*Многодольный граф* - граф, множество вершин которого можно разбить на k независимых множеств (доль).

*Qt* - кроссплатформенный фреймворк для разработки программного обеспечения на языке программирования C++.

*С++* - компилируемый, статически типизированный язык программирования общего назначения.