|  |  |
| --- | --- |
|  | ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - ВАРНА  ФАКУЛТЕТ ПО ИЗЧИСЛИТЕЛНА ТЕХНИКА И АВТОМАТИЗАЦИЯ |

**КУРСОВ ПРОЕКТ**

по

**Управление на софтуерни проекти**

Тема

**Система за избор на песен**

Изготвили:

Виктор Бисеров Петков - фак. №19621649

Елица Иванова Тодорова - фак. №19621649

Кенан Сейханов Кемалов - фак. №19621649

спец.: Софтуерни и интернет технологии

III курс, 2-ра група

В А Р Н А

2 0 2 2 год

Съдържание

[Техническо задание на проекта 3](#_Toc104219855)

[Предмет на техническото задание 3](#_Toc104219856)

[Екрани на системата 3](#_Toc104219857)

[Обхват на разработка 6](#_Toc104219858)

[Изисквания към програмния продукт 6](#_Toc104219859)

[Структура на проекта 7](#_Toc104219860)

[Бюджет 7](#_Toc104219861)

[Обосновка на избора за система за контрол на версиите 8](#_Toc104219862)

[Причина за избора 8](#_Toc104219863)

[Трудности при използването 8](#_Toc104219864)

[Настройки, нужни за работа с него 9](#_Toc104219865)

[Съпоставка на избрания софтуер с други 9](#_Toc104219866)

[Обосновка на избора за система за управление на проекти 9](#_Toc104219867)

[Причина за избора 9](#_Toc104219868)

[Трудности при използването 9](#_Toc104219869)

[Настройки, нужни за работата с него 9](#_Toc104219870)

[Мобилна/desktop/web версия 9](#_Toc104219871)

[Използвани функционалности. Отчети 10](#_Toc104219872)

[Обосновка и разяснения към избраната методология за разработка на софтуера 12](#_Toc104219873)

[Причина за избора 12](#_Toc104219874)

[Предимства на подхода 12](#_Toc104219875)

[Недостатъци на подхода, открити при изпълнението на задачата 12](#_Toc104219876)

[Съпоставка на подхода с други 12](#_Toc104219877)

[Описание на стъпките по реализацията 13](#_Toc104219878)

[Определяне на роли на членовете от екипа 13](#_Toc104219879)

[Разделяне на задачи. Определяне на срокове за проектиране, кодиране, интегриране, тестване, документиране 14](#_Toc104219880)

[Избор на програмни средства за реализацията 14](#_Toc104219881)

[Стъпки, зависещи от конкретния подход за разработка 14](#_Toc104219882)

[Жалонни точки. Timeline 15](#_Toc104219883)

[Microsoft Project. Диаграма на Гант 16](#_Toc104219884)

[Профил на риска 16](#_Toc104219885)

[Изводи 16](#_Toc104219886)

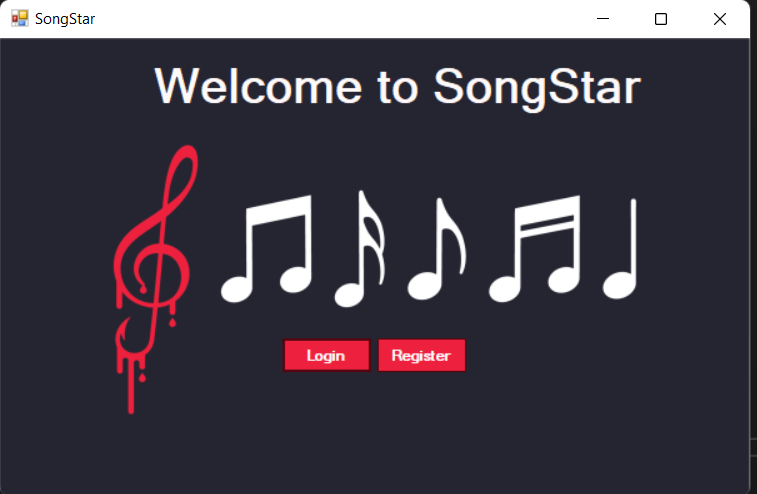
# Техническо задание на проекта

## **Предмет на техническото задание**

Техническото задание представя изискванията на възложителя по отношение на разработването на програмния продукт – Система за избор на песен. Системата трябва да предоставя възможност за добавяне на информация за песен, като изпълнител (певец/група), жанр, времетраене, година на издаване. Системата трябва да предоставя и възможност за търсене на песен по различни зададени критерии (филтриране), като изпълнител (певец/група), жанр, времетраене, година на издаване.

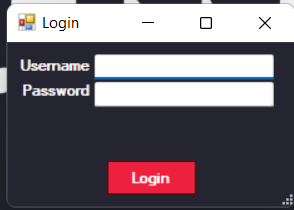
## **Екрани на системата**

Главен екран



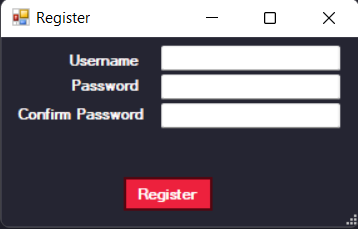
Съдържа избор за вход или регистрация в системата.

Вход



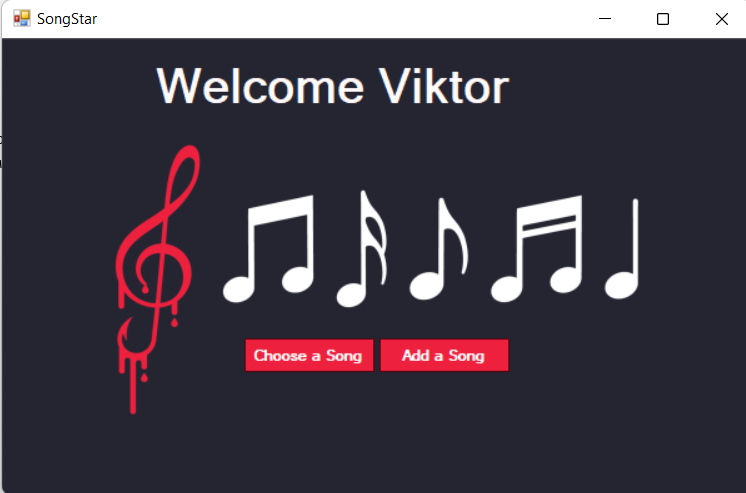
Проверява дали съществува потребителското име, хешира паролата и я сравнява с вече хешираната парола съотвестваща на потребителското име в базата данни. Допуска вход при съвпадение, в противен случай отвхърля вход.

Регистрация



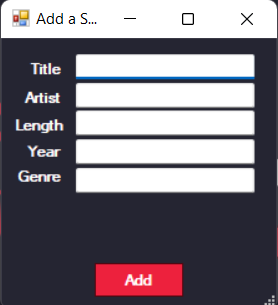
Създаване на потребителски профил. Извършват се проверки за валидност на въведеният текст (дължина, съвпадение). При успешна валидация хешира паролата и въвежда новият профил в базата данни.

Главен екран след вход в системата.



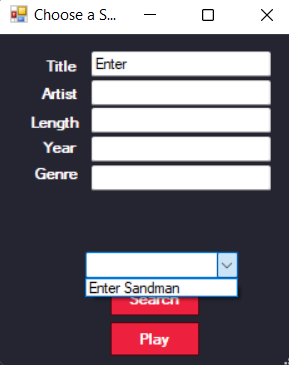
Извежда се съобщение за посрещане на потребителя. Има две нови опции – избор на песен и добавяне на песен.

Добавяне на песен



Чрез този екран могат да бъдат добавяни песни в базата данни. Извършват се верификации за правилно въвеждане на полетата (размер, тип на данни, проверки за отрицателни стойности). След натискането на бутон „Add“ се добавят данните в базата данни при успешна верификация.

Избор на песен



При този екран може да се избере песен. Извършват се верификации за правилно въвеждане на полетата (размер, тип на данни, проверки за отрицателни стойности).  
Полетата не е нужна да бъдат запълнени всички, филтрира се само по запълнените полета, търси се най-близкия резултат. При празни полета се извеждат всички песни от БД.  
При натискане на бутона „Search“ се извършва търсенето и се попълват резултатите в Dropdown лист.  
При натискане на бутона „Play“ се избира песента.

## **Обхват на разработка**

Софтуерният продукт трябва да удовлетворява всички функционални и нефункционални изисквания, поставени от възложителя на проекта. Приложението трябва да предоставя лесен и удобен за работа интерфейс, който да изпълнява зададените функционалности за въвеждане на информация за песен и търсене на песни по зададени критерии. Софтуерът следва да осигурява лесна навигация между отделните модули и компоненти на проекта.

## **Изисквания към програмния продукт**

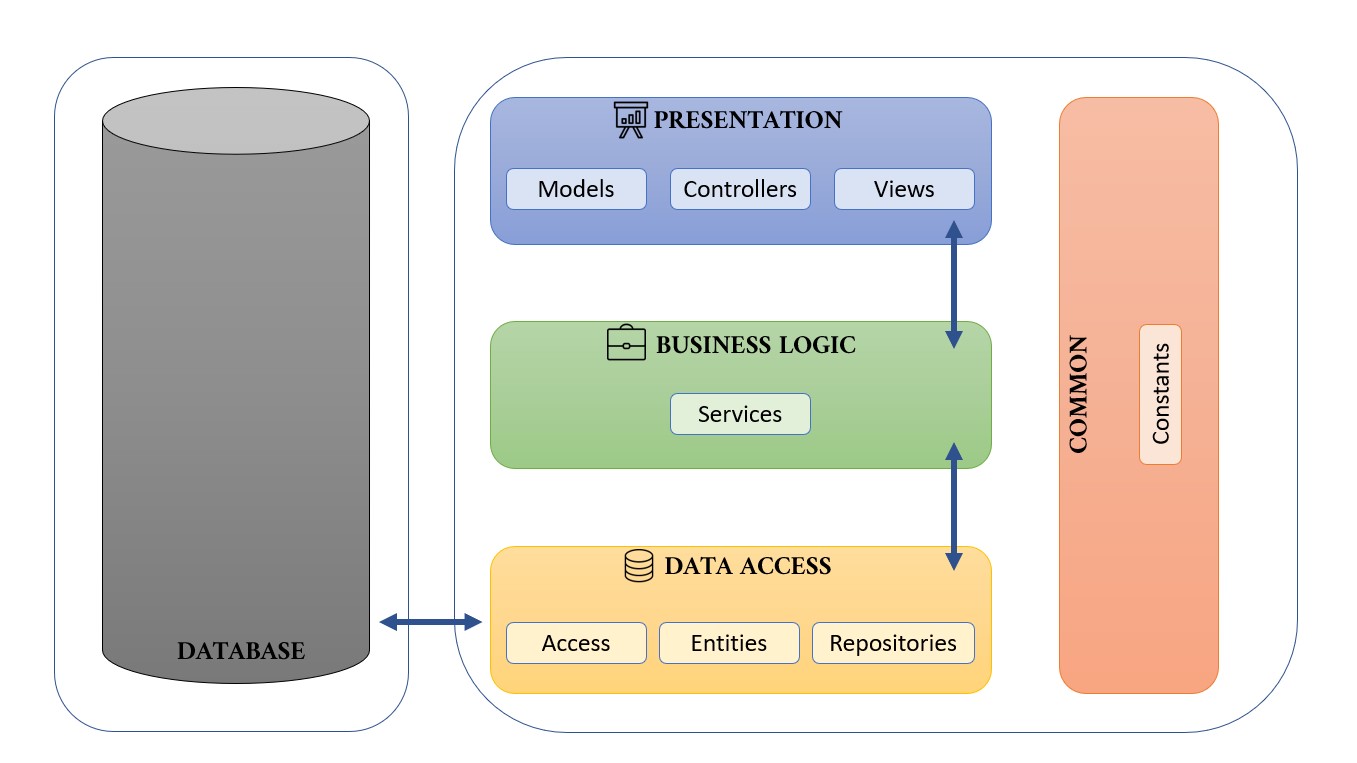
Програмният продукт трябва напълно да удовлетворява изискванията на възложителя, представени в техническото задание. Всички доставени компоненти на софтуерния продукт трябва да бъдат напълно съвместими един с друг и да функционират като цялостна система, а не като самостоятелни елементи. Приложението има минимални изисквания към хардуера, на който трябва да бъде инсталирано.

За реализацията на базата данни на проекта е използвана сървърната система за управление на релационни бази от данни (RDBMS) **Microsoft SQL Server**, защото включва професионален софтуер за управление на бази данни, който е лесен за използване и предоставя повече въможности от конкурентни продукти. MS SQL е подходящ за управление на големи бази от данни и разполага с редица опции за възстановяване на базата данни при случайна повреда, например загуба на мощност или неправилно изключване.

За реализация на интерфейса на проекта е използван шаблона **Windows Forms App (.NET Framework)** във Visual Studio на езика **C#**.

Windows Forms е рамка за изграждане на потребителски интерфейс на **Windows десктоп приложения** към компонентите на .NET Framework. Тя предоставя един от най-продуктивните и удобни начини за създаване на приложения във Visual Studio. Windows формите поддържат широк набор от функции за разработка на приложения, включително контроли, графики, data binding и въвеждане от потребителя. Разработените приложения са лесни за внедряване, актуализиране и работа. Те имат достъп до локалния хардуер и файловата система на компютъра, докато се изпълняват на него.

## **Структура на проекта**



## **Бюджет**

Бюджета за проекта е съставен въз основа на всички разходи по анализирането, разработването и документирането на проекта. Документирането на всяка итерация по разработката на проекта изисква много време и ресурси. Разработката на бизнес логика и дизайнът на потребителски интерфейс, както анализирането на функционалните и нефункционалните изисквания, и тестването, са основните стъпки в разработването на софтуерен проект и предполагат по-големи разходи, в сравнение с останалите етапи.

# Обосновка на избора за система за контрол на версиите

## **Причина за избора**

За система за контрол на версиите е избран GitHub, тъй като това е безплатна децентализирана система от 3-то поколение. Основен плюс при нея е, че всеки член на екипа може да създаде локално копие на хранилището на своя компютър и може да работи офлайн. Също така, обединяването на версии и записването в хранилището са отделни процеси. Членовете от екипа могат да работят едновременно върху един и същи файл, като след това обединят версиите си. Друго предимство е, че GitHub предоставя възможност за разклонение (създаване на клонове) на проекта, за да може всеки член на екипа да разработва свой вариант на дадена функционалност, а след това да се избере най-оптималната ѝ реализация. Други позитиви на разпределената система GitHub са лесното актуализиране на файлове и връщане към предишна работеща версия на проекта, backup на хранилището и по-малки изисквания към хардуера на централното хранилище.

## **Трудности при използването**

Трудности при разрешаването на възникнал конфликт. При промени на един и същи ред от кода, се изисква ръчно отстраняване на конфликта, което води до затруднения и загуба на време. Трудности при запазване на промени към централното хранилище, когато се е работило върху неактуален файл.

## **Настройки, нужни за работа с него**

За работа с GitHub не са нужни специални настройки. Нужно е да се направи регистрация в сайта <https://github.com/>, след което се създава repository и се канят членове на екипа, които ще разработват конкретния продукт. След това хранилището трябва да се свърже с конкретното IDE. За по-голямо удобство и лесна работа може да се изтели desktop версията на GitHub.

## **Съпоставка на избрания софтуер с други**

|  |  |  |  |
| --- | --- | --- | --- |
|  | GitHub | CVS | RCS |
| Децентализираност | Да | Не | Не |
| Личен workspace | Да | Не | Не |
| Заключване на файл | Не | Да | Да |
| Лесно унищожаване на данни | Не | Да | Да |
| Връщане към предишна версия | Да | Да | Не |
| Прихващане на конфликти | Да | Не | Не |

# Обосновка на избора за система за управление на проекти

## **Причина за избора**

За система за управление на проекта е избран Microsoft Project. Причината за избора му е лесното, мощно и динамично планиране на проекти, базирано на необходимите ресурси, продължителността на проекта и разпределените членове на екипа. В MS Project може да се въвежда информация за задачите по проекта, тяхното начало и продължителност на изпълнение, колко време ще отнеме изпълнението им и кой трябва да извърши работата.

## **Трудности при използването**

Мicrosoft Project не е безплатен софтуерен инструмент, което доведе до трудности при намирането, изтеглянето и инсталирането на неофициална работеща версия на продукта.

## **Настройки, нужни за работата с него**

Първата стъпка е създаване на проект в MS Project, след което му се задават различни свойства като заглавие, предмет на дейност, автор, мениджър, компания и др. метаданни. Определят се началният и крайният срок на изработка на проекта, както и работното време на членовете на екипа. След това се преминава към планирането на задачите за изпълнение. Те могат да се подредят структурно, като за целта се избира задачата от изгледа Ganrr Chart (диаграма на Гант). По този начин те се организират в йерархична структура – обобщаващи задачи (summary tasks) и техните подзадачи (subtasks). Може да се зададе и зависимост между задачите (когато дадена задача не може да започне преди друга да е приключила). В Microsoft Project могат да се зададат и ресурси – работни (членовете на екипа), материални (елементи, инструменти) и ценови (разходи), след което да се добавят към зададените задачите за изпълнение. След приключване на работата по проекта в MS Project, може автоматично да се генерират и издадат отчети по различни критерии.

## **Мобилна/desktop/web версия**

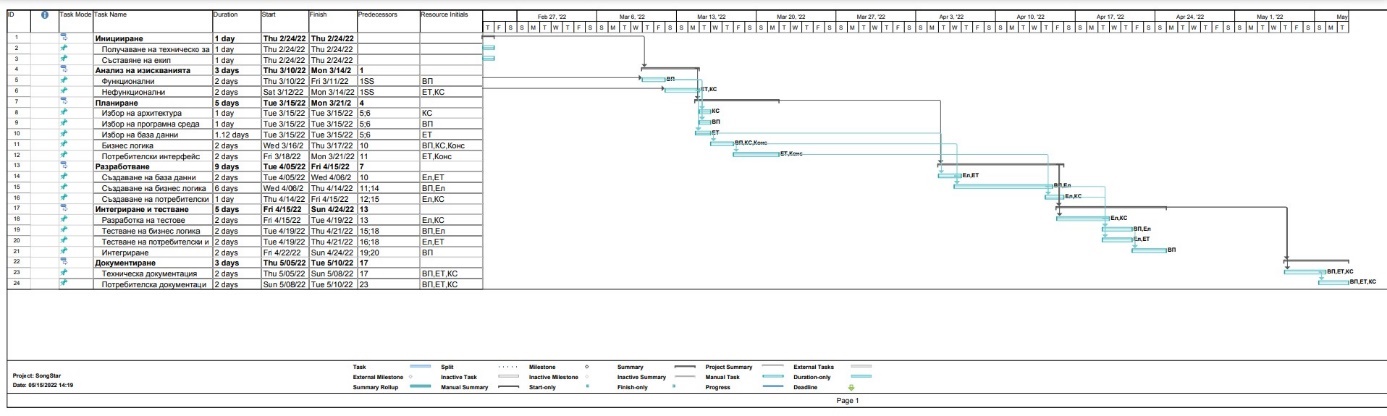
За управление на проекта е използвана desktop версията на Microsoft Project. Продуктът разполага и с мобилна версия за потребители на iOS и Android.

## **Използвани функционалности. Отчети**

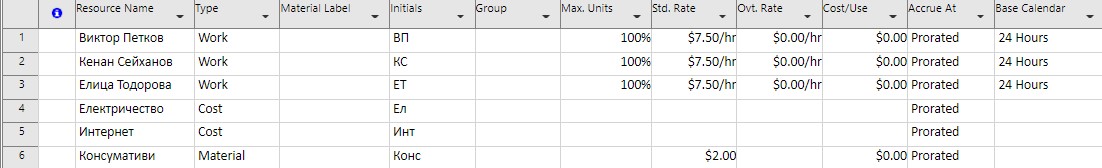
Microsoft Project предоставя редица функционалности за управление на софтуерни проекти:

* създаване на задачи от високо и ниско ниво
* информация за ресурси (работни, материални, ценови)
* задаване на срокове
* разпределяне на задачите между членовете на екипа
* бюджетиране
* изготвяне на отчети (reports)

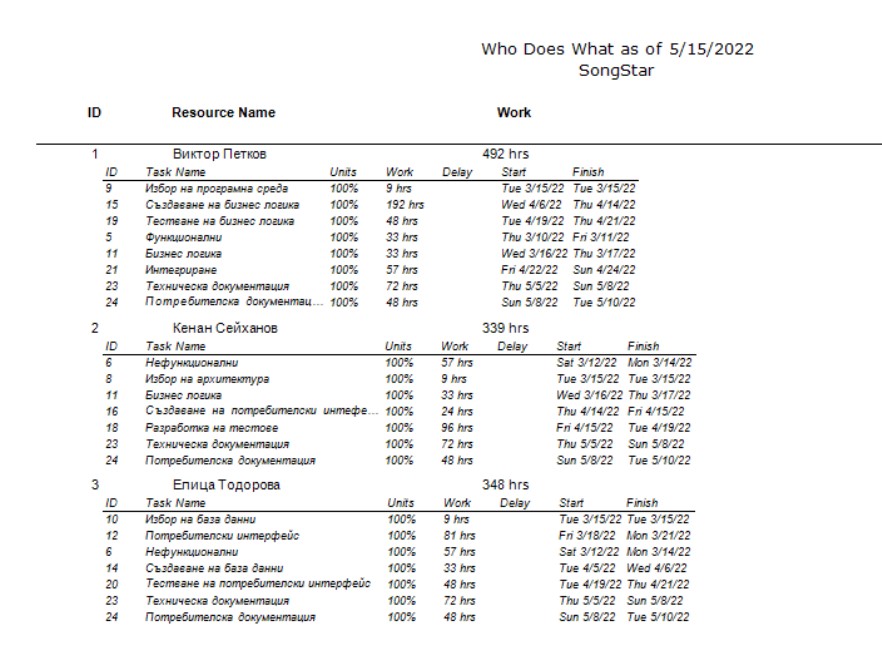
Диаграма на Гант:



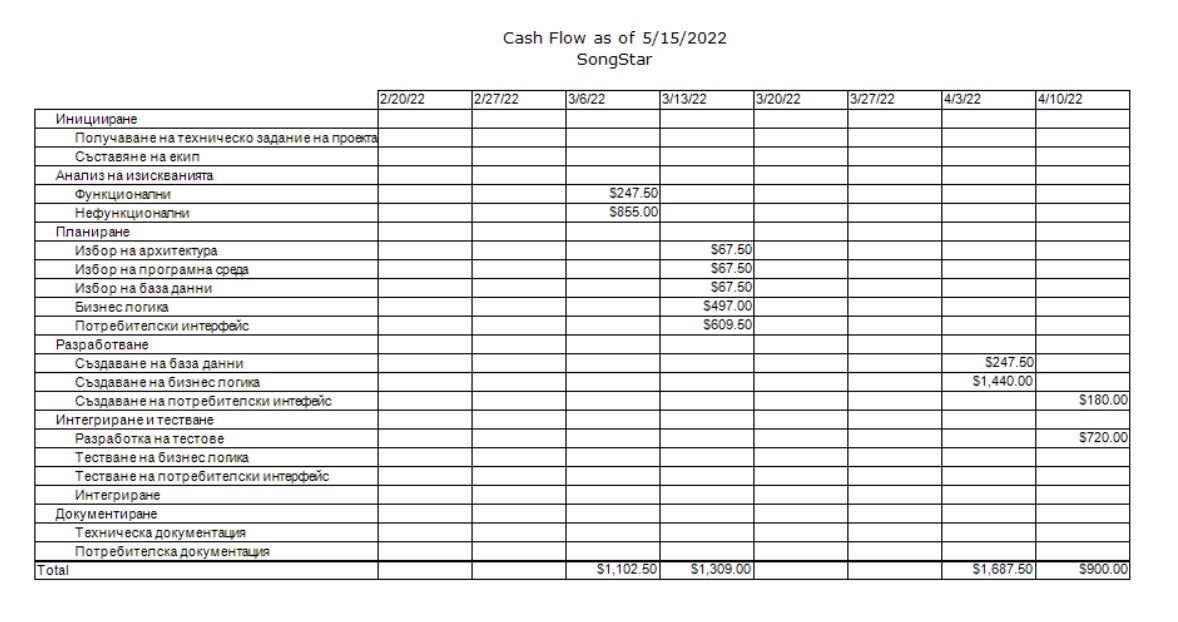
Ресурсен списък:

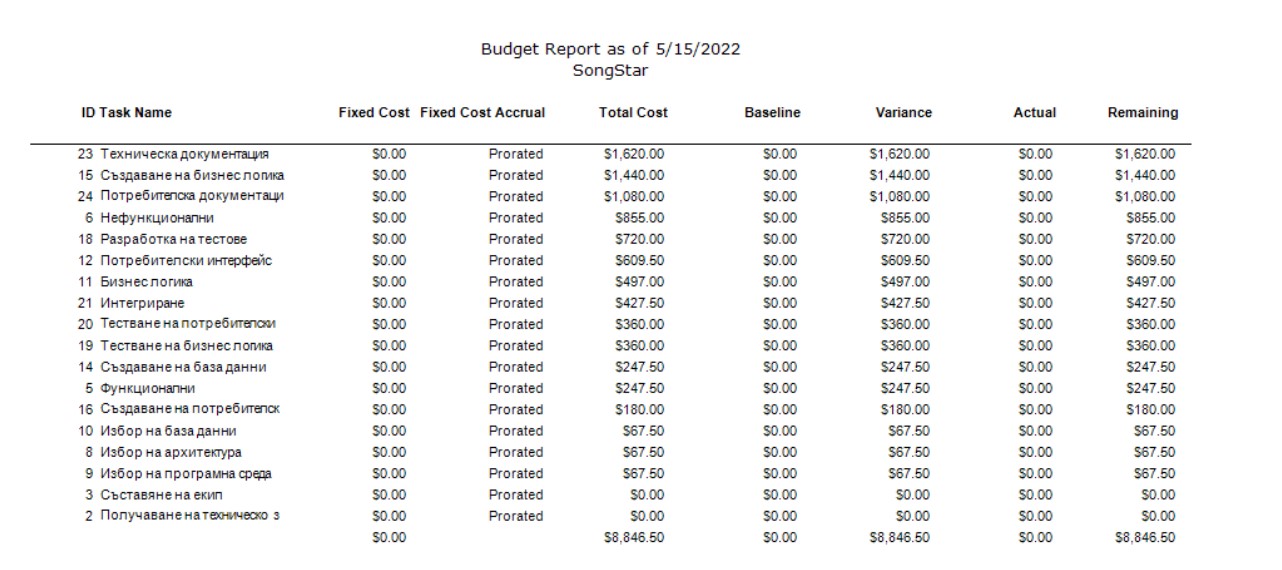


Отчет за задачите и сроковете им за изпълнение на всеки член от екипа:



Отчет за паричния поток:



Отчет на бюджета:

# Обосновка и разяснения към избраната методология за разработка на софтуера

## **Причина за избора**

Избраната методология за разработка на софтуера е Scrum. Scrum e част от Agile методологиите, които насърчават адаптивното планиране, еволюиращата разработка и доставяне на софтуер, времево-разпределения итеративен подход, както и бързото и гъвкаво реагиране на промени. С използването на Scrum методологията се повишава производителността и ефективността на разработка на проекта.

## **Предимства на подхода**

Гъвкавата Scrum методология помага на екипите да постигнат по-бързи и ефективни резултати по разработката на проекта. Гарантира ефективно използване на време и разходи. Осигурява по-лесно управление на по-големи проекти, като ги разделя на спринтове. Разработените части от проекта се тестват по време на всеки спринт. Екипът следи лесно изпълнението и напредъка на проекта, благодарение на спринт срещите. В процеса на изработка на проекта се показват негови прототипи за получаване на обратна връзка от клиентите, като по този начин промените по проекта се осъществяват по-лесно и навреме. Индивидуалната работа на всеки член от екипа може да се проследи лесно по време на ежедневните срещи.

## **Недостатъци на подхода, открити при изпълнението на задачата**

Използването на Scrum методологията за разработка на проекти в големи екипи е рисковано, тъй като трудно се осъществяват чести срещи в многоброен екип. За да бъде успешна, трябва екипът да е високо квалифициран, опитен и добре сработен. Напускането на член от екипа по време на разработката на даден проект може да доведе до пагубен резултат. Честите срещи допълнително могат да изнервят и претоварят членовете на екипа.

## **Съпоставка на подхода с други**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Scrum | Kanban | XP (Extreme programming) |
| Големина на проект | Всички | Всички | Малък |
| Определяне на срокове | Да | Не | Да |
| Обратна връзка от възложителя | Да | Не | Не |
| Стриктен процес (спазване на правила) | Да | Не | Не |
| Проследяване на работата на екипа | Да | Не | Не |
| Разширяване на обхвата на проекта | Да | Да | Да |
| Разпределение на ролите | Product Owner, Scrum Master, Team | Team | По двойки |
| Итерации | Да | Да | Не |
| Тестване | На всяка итерция | Преди внедряването | При внедряването |

# Описание на стъпките по реализацията

## **Определяне на роли на членовете от екипа**

Екип: Виктор Петков, Елица Тодорова, Кенан Кемалов

Мениджър на проекта: Виктор Петков

Разработчик на базата данни: Елица Тодорова

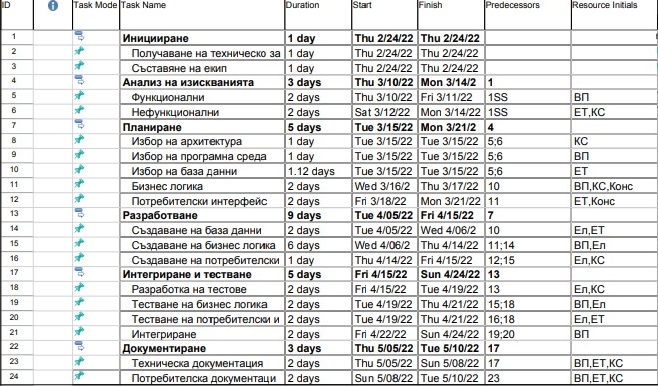
Разработчик на бизнес логика: Виктор Петков

Дизайнер на потребителски интерфейс: Кенан Кемалов

Тестер: Виктор Петков, Елица Тодорова, Кенан Кемалов

Технически писател: Виктор Петков, Елица Тодорова, Кенан Кемалов

## **Разделяне на задачи. Определяне на срокове за проектиране, кодиране, интегриране, тестване, документиране**



ВП – Виктор Петков

ЕТ – Елица Тодорова

КС – Кенан Сейханов

## **Избор на програмни средства за реализацията**

База данни: Microsoft SQL Server

Програмен език: C#

Интерфейс: Windows Forms

IDE: Visual Studio

## **Стъпки, зависещи от конкретния подход за разработка**

Избрания подход за разработка на софтуерния продукт предполага разпределение на ролите и задачите между членовете на екипа. Определят се Product Owner – възложителят на проекта, Scrum Master – ръководителят на проекта, и Team – екипът от разработчици на проекта, между които се разпределят планираните задачи.

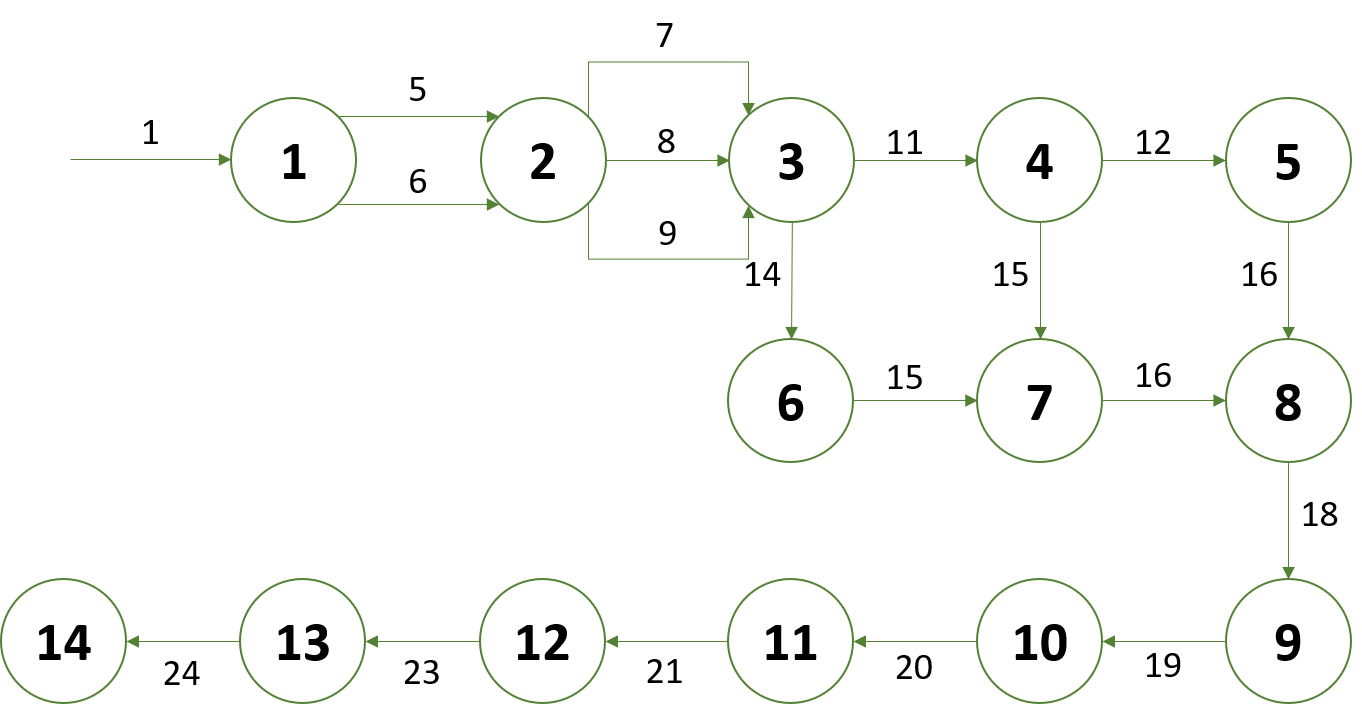
В началото ScrumMaster-ът след среща с Product Owner описва User Stories (изискванията и ограниченията, поставени от клиента), разделя задачите, подрежда ги по приоритет и ги поставя в Product Backlog (списък с неизпълнени задачи). За да бъде една задача част от Product Backlog, трябва да е преминала през етапа планиране.

Дава се начало на първи Sprint (първа итерация по разработката на софтуерния проект). Всеки ден от изпълнението на итерацията се състои среща за планирането ѝ (Sprint planning meeting), където екипът обсъжда свършената работа от предходния ден и се обсъждат задачите, които трябва да се свършат в текущия ден. Задачите, които ще се изпълняват, се местят в Sprint Backlog (списък от задачи, които се изпълняват в текущата итерация).

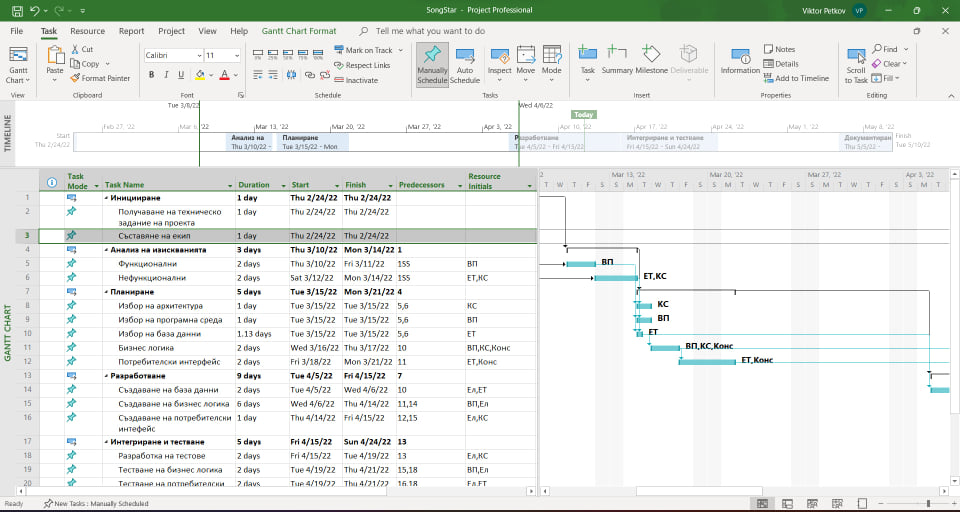
В края на Sprint-a се състои среща Sprint Retrospective, където Team-ът прави обзор на свършената работа по време на изминалата итерация. Възложителят на проекта има възможност да види получените софтуерни компоненти след края на всеки Sprint, който обикновено трае около 30 дни. По този начин екипът получава обратна връзка от възложителя по изпълнието на проекта.

## **Жалонни точки. Timeline**

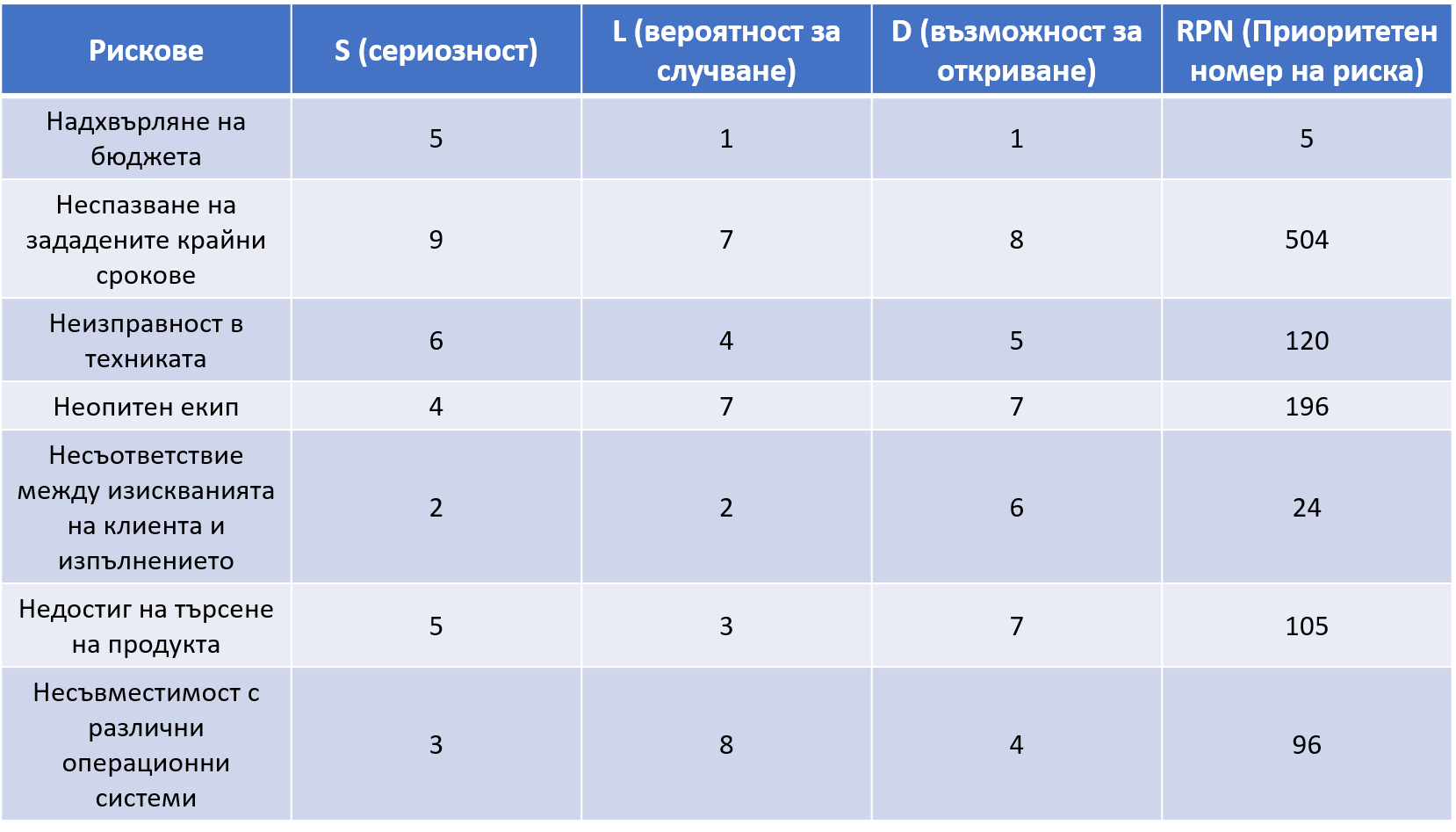
|  |  |
| --- | --- |
| Събитие | Жалонна точка |
| 1 | Проектът е иницииран. |
| 2 | Изискванията са анализирани. |
| 3 | Архитектурата, програмната среда и базата данни са избрани. |
| 4 | Бизнес логиката е планирана. |
| 5 | Потребитекският интерфейс е планиран. |
| 6 | Базата данни е разработена. |
| 7 | Бизнес логиката е разработена. |
| 8 | Потребитекският интерфейс е разработен. |
| 9 | Тестовете са разработени. |
| 10 | Бизнес логиката е тествана. |
| 11 | Потребитекският интерфейс е тестван. |
| 12 | Интегрирането е завършено. |
| 13 | Техническата документация е завършена. |
| 14 | Потребителската документация е завършена. |



## **Microsoft Project. Диаграма на Гант**



## **Профил на риска**



# Изводи

Изработката на софтуерен проект не е никак лесна задача. Изисква се подробно планиране и разпределение на задачите, висококвалифицан и добре сработен екип, бюджет, ресурси, спазване на срокове за изпълнение, анализ на риска. Използването на системи за контрол на версиите и за управление на софтуерни проекти значително улеснява работата на екипа, повишава производителността и ефективността. Изборът на подходяща методология за разработка на проект също е от голямо значение, тъй като в противен случай проектът може да не бъде успешно завършен. За да бъде качествен един софтуерен продукт, трябва напълно да отговаря на функционалните и нефункционалните изисквания, зададени от възложителя на проекта, затова етапът на анализиране на изискванията не трябва да бъде подценяван. Трябва да се отдели достатъчно време на всеки етап от разработката на софтуерния продукт. Бюджетът и профилът на риска трябва да бъдат внимателно разгледани и обсъдени с членовете на екипа, за да се избегнат проблеми по време на разработката на проекта, като надхвърляне на определения бюджет, неспазване на зададените срокове или други рискови събития.