**Курсов проект**

**по**

**Управление на софтуерни проекти**

**Изработили:**

**Кристиан Георгиев 21621632**

**Даниел Гутцов**

**Димитър Сираков**

**Специалност: СИТ**

**Курс: III**

**Група: 4**

Съдържание

[Техническо задание на проекта **Error! Bookmark not defined.**](#_Toc1306728712)

[Бюджет 4](#_Toc2105103721)

[Предмет на техническото задание **Error! Bookmark not defined.**](#_Toc1511278496)

[Начален екран 5](#_Toc495831546)

[Екран за вход 5](#_Toc2010924189)

[Екран за регистрация 6](#_Toc266440217)

[Основен екран за избор на прожекция на филм 7](#_Toc1117205630)

[Екран за закупуване на билети 8](#_Toc850568359)

[Обхват на разработката 8](#_Toc1549937485)

[Изисквания към програмния продукт 9](#_Toc833019046)

[Общи изисквания 9](#_Toc841217278)

[Обосновка на избора на система за контрол на версиите 10](#_Toc915323934)

[Причина 10](#_Toc685957769)

[Трудности при използването 10](#_Toc63319516)

[Настройки 10](#_Toc1599379125)

[Съпоставка 11](#_Toc1322005823)

[Обосновка на избора за система за управление на проекти: 11](#_Toc518347649)

[Причина за избора; 11](#_Toc1834019486)

[Трудности при използването; 11](#_Toc253032260)

[Настройки, нужни за работата с него; 12](#_Toc53958172)

[мобилна/desktop/web версия; 12](#_Toc1507036565)

[Използвани функционалности. 12](#_Toc1966021532)

[Обосновка и разяснения към избраната методология за разработка на софтуера 12](#_Toc1620130668)

[Предимства: 13](#_Toc737952672)

[Недостатъци: 14](#_Toc1874599392)

[Описание на стъпките по реализацията: 15](#_Toc1073443944)

[Екип 15](#_Toc1811984348)

[Определяне на роли 15](#_Toc1541473322)

[Разделяне на задачи 15](#_Toc1606838056)

[Определяне на срокове 17](#_Toc1336128707)

[Програмни средства за реализацията 17](#_Toc627945403)

[Стъпки, зависещи от конкретния подход за разработка 17](#_Toc982934127)

[Жалонни точки 17](#_Toc824199871)

[Timeline 18](#_Toc1972630026)

[Microsoft Project 20](#_Toc1276190377)

[Календар 20](#_Toc859387735)

[Диаграма на Гант 20](#_Toc328654356)

[Ресурсен лист 21](#_Toc108479728)

[Проследяване на завършеност на задачите 21](#_Toc1330231688)

[Диаграма на свършената работа 22](#_Toc486930410)

[Диаграма на бюджет 22](#_Toc1717763263)

[Таблица на бюджет 23](#_Toc1621763149)

[Диаграма на ресурси 23](#_Toc1221479201)

[Бизнес правила: 23](#_Toc1211159881)

[Обекти и техните атрибути 24](#_Toc1909950336)

[Изводи 24](#_Toc2114344712)

Техническо задание на проекта: Сайт за автомобили втора ръка

## Бюджет

Бюджетът за проекта е съставен въз основа на всички разходи по анализирането, проектирането, разработването и документирането на сайта. Анализът на изискванията е ключов етап, включващ както функционални, така и нефункционални изисквания. На този етап са отделени два работни дни, изпълнявани паралелно от двама участници в екипа. За планирането на проекта са предвидени четири работни дни. Този етап отнема повече време, за да се избегнат бъдещи грешки, чието поправяне би било по-скъпо в по-късни етапи на проекта. В този етап се проектира архитектурата на приложението, която трябва да е подходяща за бъдещо развитие и лесно адаптиране на нови членове в екипа. Добрата архитектура също така осигурява добре написан и структуриран код, без повторения.

За разработката са отделени пет работни дни, тъй като след успешното проектиране, реализацията е ясна и не се очаква да бъде времеотнемаща. В зависимост от обхвата на приложението, за тестването е отделен един работен ден, а за изготвянето на документациите – три работни дни. Общата стойност на проекта възлиза на 3.410.25 лв., включвайки разходите за компютри (по 100 лв на човек) и добавени 15% за евентуални бъдещи процеси, свързани с поддръжката на приложението.

## Предмет на техническото задание

Изискванията към сайта включват показване на наличните автомобили втора ръка и възможност за покупка на избран автомобил. Функционалностите на сайта включват:

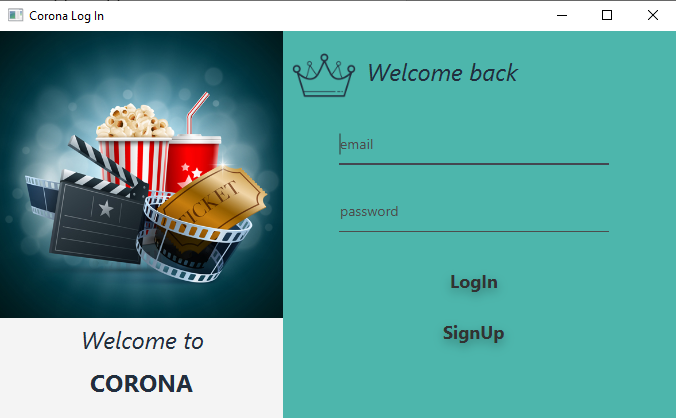
1. **Показване на наличните автомобили:** Списък с автомобили, включващ информация за марка, модел, година на производство, пробег, цена и снимки.
2. **Търсене и филтриране:** Възможност за търсене по марка, модел, година на производство, цена, пробег и други критерии.
3. **Детайли за автомобила:** Подробна информация за избран автомобил, включително технически спецификации, история на обслужване и друга релевантна информация.
4. **Покупка на автомобил:** Форма за заявка за покупка, включваща избор на метод за контакт и плащане.
5. **Популярни автомобили:** Показване на най-популярните автомобили на сайта. Популярността на автомобила се изчислява въз основа на броя на прегледите и заявките за покупка.

Всички екрани и техните функционалности са описани подробно в документацията. Приложението трябва да има също и административен панел за управление на съдържанието, който да позволява добавяне, редактиране и изтриване на обяви за автомобили, както и преглед на статистики за популярността на автомобилите и поведението на потребителите.

### **Регистрация**

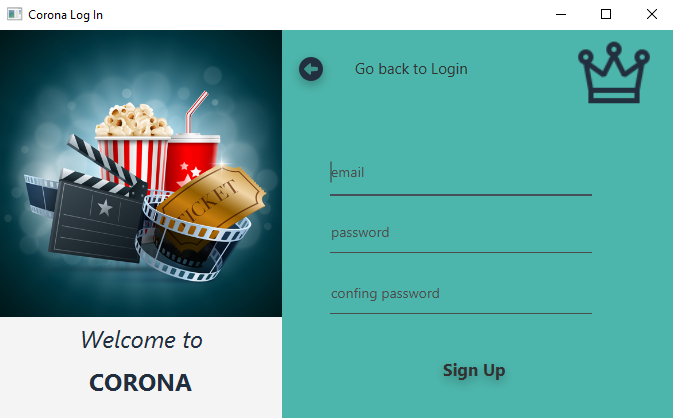
При пускане на приложението се зарежда прозорец с възможност за влизане. Чрез него се стига и до другия прозорец за регистриране.

### Екран за вход



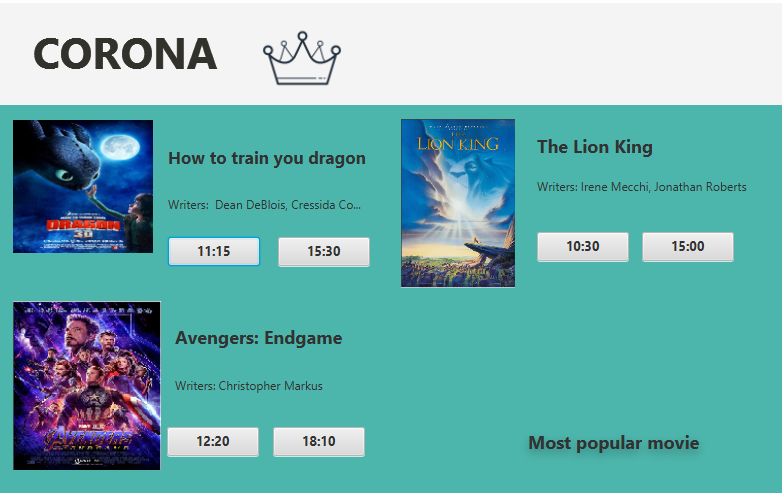
Екранът за вход съдържа две полета. Първото е за въвеждане на имейл адрес на потребителя. Въведеният в това поле текст се валидира за покриване на изискванията на всеки имейл. Другото поле е за парола, която също има валидация. Паролата на потребител трябва да е не по-къса от 8 символа. При грешно въведени данни излиза съобщение за невалидна входна информация.

### Екран за регистрация



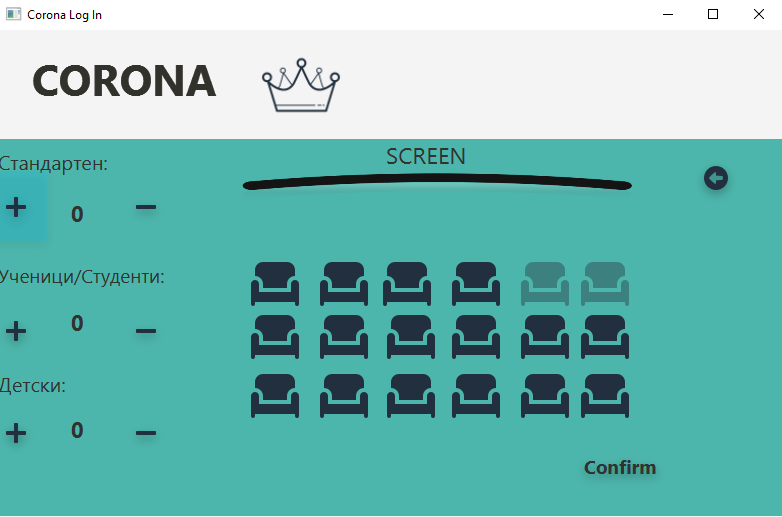
Екранът за регистрация съдържа три полета. Първото е за въвеждане на електронна поща на потребителя със съответните валидации, а другите две са за въвеждане на парола и повторно въвеждане на паролата. След успешно създадена регистрация потребителя се пренасочва към екрана за вход, чрез който може да влезе и да закупи билет.

### Основен екран за избор на прожекция на филм



При успешно влизане в системата пред потребителя се появява екран съдържащ трите прожектирани в киното филма. Всеки филм има по две прожекции, чиито часове са изобразени. За закупуване на билет за дадена прожекция се избира бутона с часа, който отвежда потребителя към екран за закупуване на билети.

### Екран за закупуване на билети



На този екран са показани заетите и свободните места за всяка една прожекция. Свободните места са показани с по-тъмен цвят от заетите. От ляво потребителят може да избере колко и от какъв вид билети иска да закупи, а от дясната страна може да избере местата в кино салона, които иска да запази.

## Обхват на разработката

Софтуерното приложение трябва да покрива всички изисквания, зададени от възложителя, и да предоставя лесен и интуитивен интерфейс за работа. Разработеният продукт предоставя на потребителите възможност да влязат в техния профил или да създадат профил, ако още нямат такъв. След влизане в приложението, потребителят може да разглежда наличните автомобили, да филтрира резултатите по различни критерии и да избере автомобил за покупка. При избора на автомобил, потребителят може да види подробности за него, включително технически спецификации, история на обслужване и снимки. Приложението предоставя също така възможност за контакт с продавача чрез форма за заявка. Включен е и функционал за показване на най-популярните автомобили, определени въз основа на броя на прегледите и заявките за покупка.

## Изисквания към програмния продукт

### Общи изисквания

Софтуерният продукт трябва да обхваща всички случаи на използване и поведението на системата да е предсказуемо. При този конкретен продукт потребителят няма възможност да въвежда данни, които да попречат или да повредят системата. Всички входни точки трябва да бъдат валидирани.

### Общи изисквания

1. **HTML, CSS, JavaScript:** За създаване на фронтенд частта на приложението.
2. **Bootstrap:** За създаване на адаптивен и естетически приятен потребителски интерфейс.
3. **Backend платформа:** За реализиране на логиката на сървъра и обработка на заявки.
4. **MySQL:** За запазване на информацията за потребителите и автомобилите.

### Общи изисквания

1. **Регистрация и вход в профил**: Потребителите могат да създадат нов профил или да влязат в съществуващ профил.
2. **Преглед на налични автомобили**: Списък с автомобили с информация за марка, модел, година на производство, пробег, цена и снимки.
3. **Търсене и филтриране**: Възможност за търсене и филтриране на автомобилите по различни критерии като марка, модел, година на производство, цена и пробег.
4. **Детайли за автомобил**: Подробна информация за избран автомобил, включително технически спецификации и история на обслужване.
5. **Заявка за покупка**: Форма за заявка за покупка, включваща избор на метод за контакт и попълване на информация за контакт.
6. **Популярни автомобили**: Показване на най-популярните автомобили на сайта въз основа на броя на прегледите и заявките за покупка.

Изисквания към хардуера и софтуера

* **Минимални хардуерни изисквания**: Компютър с интернет връзка и съвременен уеб браузър.
* **Софтуерни изисквания**: Уеб браузър, поддържащ HTML, CSS и JavaScript.

Продуктът трябва да бъде оптимизиран за работа на различни устройства, включително настолни компютри, лаптопи, таблети и смартфони. Приложението трябва да бъде тествано за съвместимост с най-популярните уеб браузъри като Chrome, Firefox, Safari и Edge.

# Обосновка на избора на система за контрол на версиите

## Причина

Една от основните причини за избора на Git беше фактът, че Git позволява на всеки участник в екипа да притежава собствено repository, в което да работи, като по този начин можем едновременно да работим върху един и същи файл. Git предлага възможност за лесно презаписване и актуализиране на файлове, както и връщане към предходно състояние на проекта, в случай че се откажем от дадена идея. Git позволява също и създаването на клонове, което ни дава възможност да разработваме различни допълнителни идеи, без да възпрепятстваме главния проект.

## Трудности при използването

Някои от трудностите при използването на Git включват:

* **Проследяване на комити**: Намирането на конкретния комит, който причинява бъг след голямо сливане, може да отнеме значително време.
* **Branch функционалност**: Реализирането на функционалността на клоновете и разрешаването на конфликти между тях може да бъде сложно и времеемко.

## Настройки

За да се използва Git, се изисква регистрация в GitHub. Трябва да се изтегли конзолно приложение за реализацията на локално repository, което после се свързва с remote repository в GitHub чрез създадените акаунти. Препоръчително е изтеглянето на помощни средства за визуализация на промените по локалното и remote repository-та. В нашия проект използвахме TortoiseGit за тази цел. Освен това, проектът беше реализиран чрез Visual Studio Code, което предоставя интеграция с Git и GitHub, улеснявайки управлението на версиите и колаборацията.

## Съпоставка

# При използването на Git, промените по основната версия на проекта могат да се извършват на стъпки, като се commit-ват само няколко файла наведнъж, придружени от детайлен коментар за извършените промени. Git използва система от repository-та (едно общо на централизирания сървър, където се съхранява главната версия на проекта, и множество локални), и когато се синхронизират данни между repository-та, всяко repository има цялата история на всички промени по проекта. Това позволява лесно преминаване към предходна версия на проекта, за разлика от други системи за контрол на версиите, при които всичко се намира на един централен сървър. В тези системи мениджърите на проекта раздават роли на участниците, които роли дават ограничен достъп до определени файлове и функции.

# Използването на GitHub чрез VS Code предоставя множество предимства, включително интегрирани инструменти за контрол на версиите, лесна визуализация на промените и конфликти, и възможност за ефективно управление на проектите чрез GitHub Actions и други CI/CD инструменти. Това значително улесни работата на екипа и подобри ефективността на разработката.

# Обосновка на избора за система за управление на проекти:

## Причина за избора;

Trello позволява лесно управление на разпределението на задачи чрез прости за използване инструменти за организиране и също така дава възможност за проследяване на прогреса. Приложението изпраща известия при възникнали промени - нови или завършени задачи, което улеснява комуникацията между членовете на екипа. Trello дава възможност за общи дискусии, прикрепяне на коментари към отделните задачи и лесно споделяне на файлове. Всички тези функционалности правят Trello идеален инструмент за управление на екипни проекти, като същевременно остава лесен за употреба и гъвкав.

## Трудности при използването;

* **Липса на връзки между задачи**: Trello няма функция, която да показва връзките между отделни задачи, т.е., коя задача трябва да се реализира, за да може да се реализира друга.
* **Проследяване на време и ресурси**: Trello не предоставя вградени инструменти за проследяване на време и ресурси, което затруднява следенето на крайни срокове и вместването в бюджета.

## Настройки, нужни за работата с него;

За използването на web версията на Trello се изисква регистрация в Trello. След това се определя project manager, който създава нов проект и добавя членовете на екипа в него. Видимостта на проекта може да бъде определена като публична или скрита (само за участниците в него).

## мобилна/desktop/web версия;

Trello също така има мобилна и desktop версии, които могат да бъдат изтеглени или закупени чрез съответно Google Play, Microsoft Store и App Store за потребителите на Apple (iPhone, iPad, Mac). Тези версии позволяват на потребителите да имат достъп до проекта от всяко устройство, което улеснява комуникацията и координацията в екипа.

## Използвани функционалности.

# **Добавяне на допълнителни изисквания**: За всяка задача могат да бъдат добавени специфични изисквания за тип на файл и коментари за срок на изпълнение.

# **Проследяване на история на промените**: В Trello се следи историята на промените към задачите, включително кой коя задача е поел, кога я е изпълнил, коя изпълнява в момента, колко задачи са изпълнени и колко остават.

# **Известия и напомняния**: Trello изпраща известия за нови или завършени задачи, което помага на екипа да е в течение с текущия прогрес.

# **Обсъждания и коментари**: Възможност за общи дискусии и прикрепяне на коментари към отделни задачи, което улеснява комуникацията в екипа.

# **Споделяне на файлове**: Лесно споделяне на файлове, свързани със задачите, което подобрява ефективността на съвместната работа.

Trello е избран заради своята лекота на употреба, гъвкавост и ефективност в управлението на задачи и проекти, като същевременно предоставя достатъчно функционалности за успешно изпълнение на проекта.

# Обосновка и разяснения към избраната методология за разработка на софтуера

Този проект е реализиран използвайки водопадния модел на работа. Водопадният модел най-често се употребява при разработка на софтуер, при който промяната на изискванията и обхвата е малко вероятна и не е критична. Зададеният проект бе достатъчно кратък и с достатъчно прости изисквания, за да може от началото да се дефинират ясно, пълно и конкретно стъпките, които да следват една след друга, за да бъде достигнат желаният резултат.

Основното предимство при този модел е, че процесите протичат линейно и последователно. Това означава, че всеки от етапите в процеса на разработка започва само когато предишната фаза е напълно завършена. При стриктно спазване на методологията връщане към предишна фаза за преправяне на продукта, поради промяна на изискванията, не се допуска. Ако евентуално се наложи, е възможно само с една стъпка назад, но това е трудно.

## Предимства:

* **Добре структуриран подход**: Всички етапи и дейности в тях са точно определени, което позволява ясно разбиране на процеса и стъпките за изпълнение.
* **Последователност**: Всеки етап завършва преди да започне следващият, което гарантира пълно изпълнение на задачите във всяка фаза.
* **По-бързо изпълнение на проекта**: Ясната структура и последователност на етапите ускоряват изпълнението.
* **Избягване на отклонения на обхвата на проекта**: Поради ясно дефинираните изисквания и стъпки се минимизира рискът от неразбиране или промяна на изискванията.
* **Конкретни резултати във всяка фаза**: Всеки етап има ясни резултати, които могат да бъдат оценени и проверени.
* **Точни срокове и бюджети**: Структурираната природа на водопадния модел позволява точно предвиждане на сроковете и разходите.

## 

## Недостатъци:

* **Малка гъвкавост**: Проектът е трудно да се адаптира към промени в изискванията или обхвата след като е започнал.
* **Трудно връщане към предишен етап**: След като един етап е завършен, връщането назад за промени е трудно и може да бъде скъпо.

**Съпоставка със спираловидния модел**

Водопадният модел е класически линеен модел, който следва последователни етапи, докато спираловидният модел следва еволюционен път. Спираловидният модел е еволюционен тип модел на жизнения цикъл на софтуера, който обединява характеристиките на прототипния модел и модела на водопада. Резултатът от спираловидния модел е серията от постепенните версии на софтуера.

|  |  |
| --- | --- |
| **Водопаден модел** | **Спираловиден модел** |
| Моделът е предпочитан от клиентите | Моделът е предпочитан от разработчиците |
| Грешки и рискове се идентифицират и отстраняват след приключване на съответния етап | Грешките и рисковете са идентифицирани и отстранени по-рано |
| Добре приложим при малки проекти | Добре приложим за големи проекти |
| Необходимо планиране в ранен етап | Не е необходимо планиране в ранен етап |
| Много малка гъвкавост | Гъвкавостта за промяна не е трудна |
| Сравнително евтин модел | Много скъп модел |
| Необходимо е задълбочено разбиране на изискванията от самото начало | Изискванията могат да се добавят в новите итерации, ако е необходимо |
| Произвежда единичен краен продукт в края на процеса | Генерира груб работещ продукт при всяка итерация |

# **Заключение**

# Изборът на водопадния модел за този проект се основава на ясно дефинираните изисквания и краткия времеви обхват. Този модел предоставя структурираност и последователност, които са идеални за проекти с добре определени и стабилни изисквания. Въпреки някои ограничения като малката гъвкавост и трудността при връщане към предишен етап, водопадният модел е подходящ за този проект, като гарантира яснота, точност и ефективност в изпълнението.

# Описание на стъпките по реализацията:

## Екип

Кристиан Георгиев

Даниел Гутцов

Димитър Сираков

## Определяне на роли

Мениджър на проекта: Кристиан Георгиев

Софтуерен архитект: Кристиан Георгиев, Даниел Гутцов

Дизайнер на потребителски интерфейс: Кристиан Георгиев

Разработчик на бизнес логиката: Даниел Гутцов, Димитър Сираков

Технически писател: Димитър Сираков, Кристиан Георгиев

Тестер: Даниел Гутцов

## Разделяне на задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Номер на задачата** | **Описание** | **Предшестваща задача** | **Време** | **Изпълнител** |
| 1 | Анализ на изискванията |  |  |  |
| 2 | Функционални изисквания | 1 | 1 ден | Кристиан Георгиев |
| 3 | Нефункционални изисквания | 1 | 1 ден | Даниел Гутцов |
| 4 | Планиране |  |  |  |
| 5 | Архитектура | 1 | 1 ден | Кристиан Георгиев |
| 6 | База данни | 1 | 1 ден | Димитър Сираков |
| 7 | Бизнес логика | 1 | 2 дни | Димитър Сираков, Даниел Гутцов |
| 8 | Потребителски интерфейс | 1 | 1 ден | Кристиан Георгиев |
| 9 | Разработка | 4 |  |  |
| 10 | Създаване на базата данни | 8 | 1 ден | Димитър Сираков |
| 11 | Записване на информация в базата | 10 | 1 ден | Димитър Сираков |
| 12 | Потребителски интерфейс | 8 | 2 дни | Кристиан Георгиев |
| 13 | Бизнес логика | 13 | 3 дни | Даниел Гутцов |
| 14 | Цялостно тестване | 13 | 1 ден | Кристиан Георгиев |
| 15 | Документиране | 9 |  |  |
| 16 | Обща документация | 14 | 2 дни | Димитър Сираков |
| 17 | Потребителска документация | 15 | 1 ден | Даниел Гутцов |

## 

## Определяне на срокове

Начална дата: 26/04/2024

Анализ на изискванията: 26/04/2024

Планиране: 27/04/2024

Разработка: 02/05/2024

Тестване: 09/05/2024

Документиране: 14/05/2024

Програмни средства за реализацията

* **Език за програмиране:** HTML, CSS, JavaScript
* **Платформа за визуализация:** Bootstrap
* **База данни:** MySQL
* **IDE:** Visual Studio Code
* **Допълнителни помощни средства:** GitHub, TortoiseGit

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

Избраният модел на разработване прилага последователни стъпки на изпълнение на зададените задачи. В тази методология задачите се изпълняват последователно и всяка една приключва с документация на етапа.

## Жалонни точки

|  |  |
| --- | --- |
| **Събитие** | **Описание** |
| 0 | Начало |
| 1 | Изискванията са анализирани |
| 2 | Планирани са архитектурата и базата данни |
| 3 | Планирана е бизнес логиката |
| 4 | Проектиран е дизайнът |
| 5 | Базата данни е създадена |
| 6 | Записана е информация в базата и потребителският интерфейс е направен |
| 7 | Бизнес логиката е реализирана |
| 8 | Тестването е приключило |
| 9 | Общата документация е завършена |
| 10 | Потребителската документация е завършена |

## Timeline

A screenshot of a computer

Description automatically generated

# Microsoft Project

## Календар

A screenshot of a calendar

Description automatically generated

A screenshot of a computer

Description automatically generated

## Диаграма на Гант

A screenshot of a computer

Description automatically generated

## Ресурсен лист

A screenshot of a computer screen

Description automatically generated

## Проследяване на завършеност на задачите

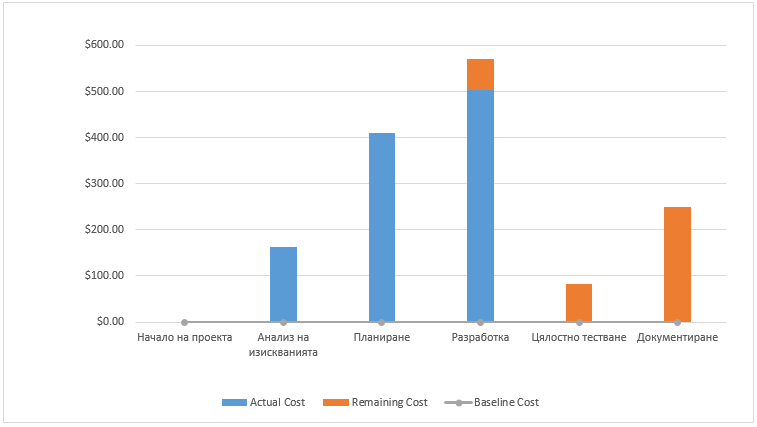
A screenshot of a computer

Description automatically generatedA screenshot of a computer

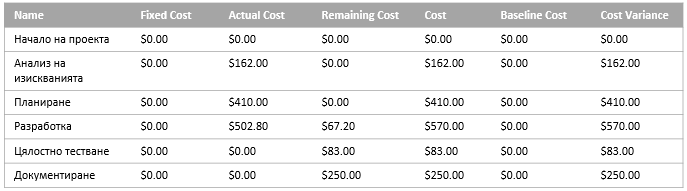
Description automatically generated

## Диаграма на свършената работа

## Диаграма на бюджет



## Таблица на бюджет



## Диаграма на ресурси



# Бизнес правила:

1. Потребителят няма право да публикува обява за продажба на автомобил без регистрация.
2. Един автомобил може да има само една обява.
3. Всеки автомобил може да бъде обявен за продажба само веднъж.
4. Потребител не може да закупи автомобил, който вече е продаден.
5. Потребител не може да публикува обява за един и същ автомобил два пъти.
6. Един потребител може да публикува повече от една обява за различни автомобили.

# Обекти и техните атрибути

**Automobile:**

* **automobileId** – уникален идентификатор на автомобила
* **automobileName** – име на автомобила (марка и модел)
* **automobileYear** – година на производство
* **price** – цена на автомобила
* **description** – описание на автомобила
* **sellerId** – идентификатор на продавача (потребител)

**User:**

* **userId** – уникален идентификатор на потребителя
* **name** – име на потребителя
* **password** – парола на потребителя
* **contactInfo** – контактна информация на потребителя

# Изводи

Използването на системи за управление на проекта значително повишава производителността на участниците в него. Системата за контрол на версиите Git дава възможност за независима разработка и лесно синхронизиране на извършената работа, лесно придвижване между версии на приложението и създаване на разклонения за изработване на допълнителни функционалности. За достигане на успешно завършен проект, удовлетворяващ потребителските изисквания, който влиза в бюджета и заложеното време за изпълнение, е нужно да се отдели достатъчно време на всеки един от етапите и допълнително внимание на етапите по планиране и анализ на отделните детайли от проекта.