**Курсов проект**

**по**

**управление на софтуерни проекти**

**Изготвили:** Добромир Начев, Александър Стоименов

**Факултетен номер:**186217900,18621697

**Специалност:** СИТ

**Курс:** III

**Група:** 2б

**Техническо задание на проекта**

***1.Бюджет***

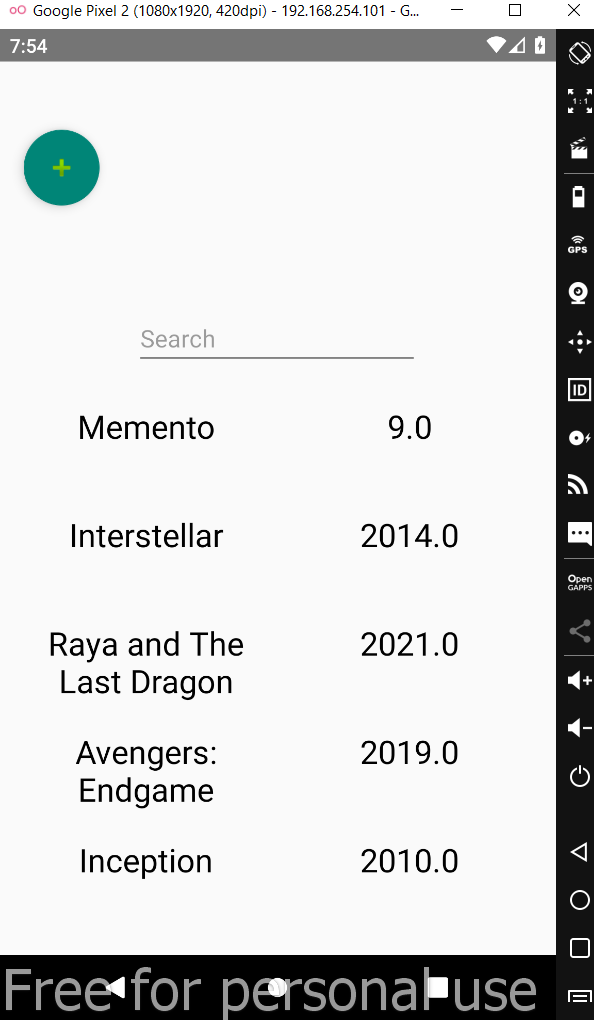
Бюджета е изчислен въз основа на разходите, които трябва да се покрият по време на разработването на проекта. Избор на подходяща архитектура за изготвянето на програмния код. Изчисляване на необходимото време за завършването на проекта и изчисляване на почасовото заплащане на разработчиците както и използваните от тях ресурси.

***2.Предмет на техническото задание***

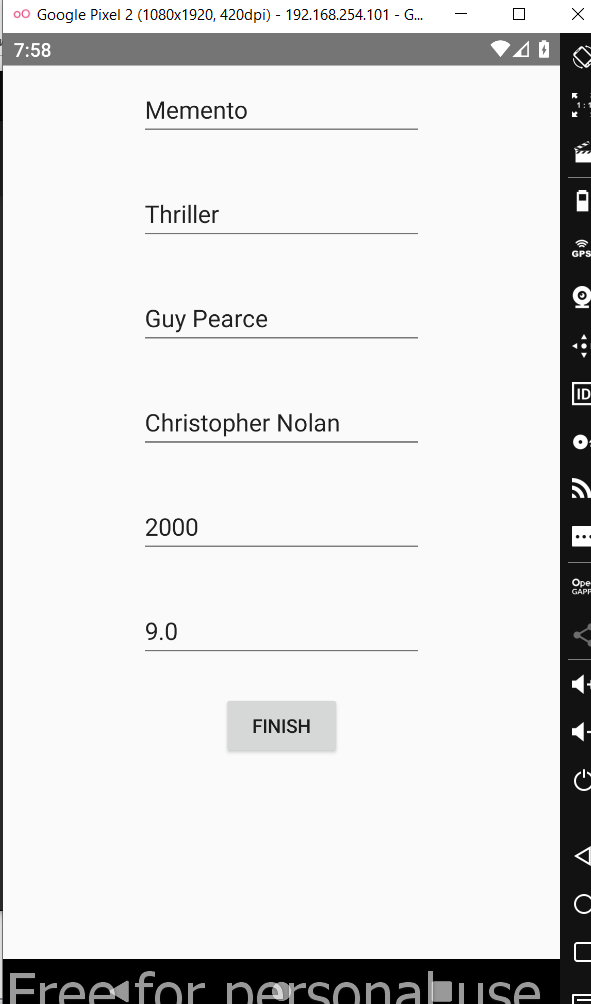
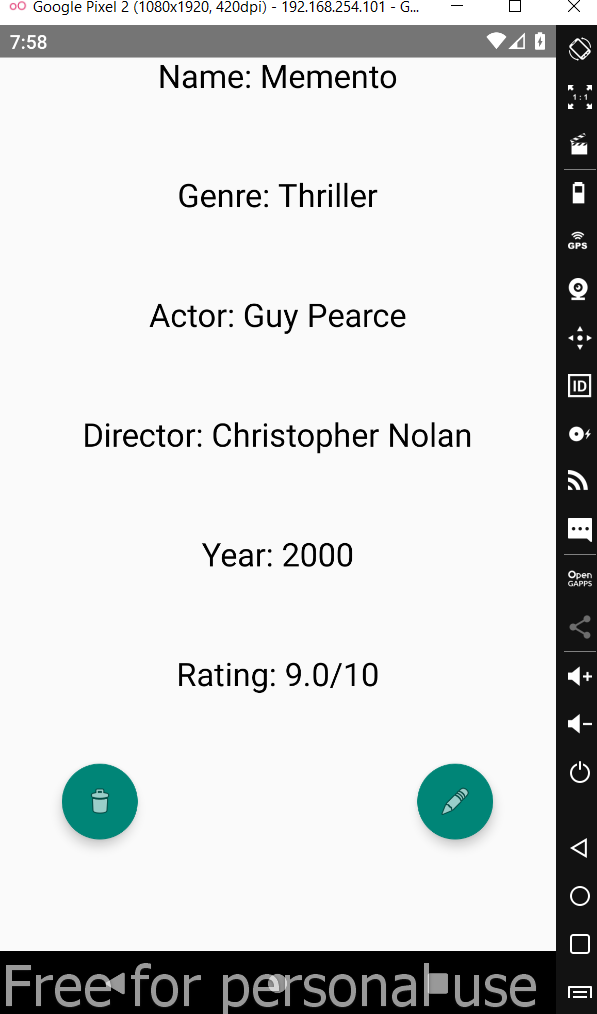
Система предлага съхраняване на данни за филми, филтриране по определени категории и тяхното извличане . В техническото задание се описват пълните и точни първоначални изисквания към проекта, неговите характеристики, както и изпълняваните от него функции и крайната цел и идея на програмния продукт.

Изисквания

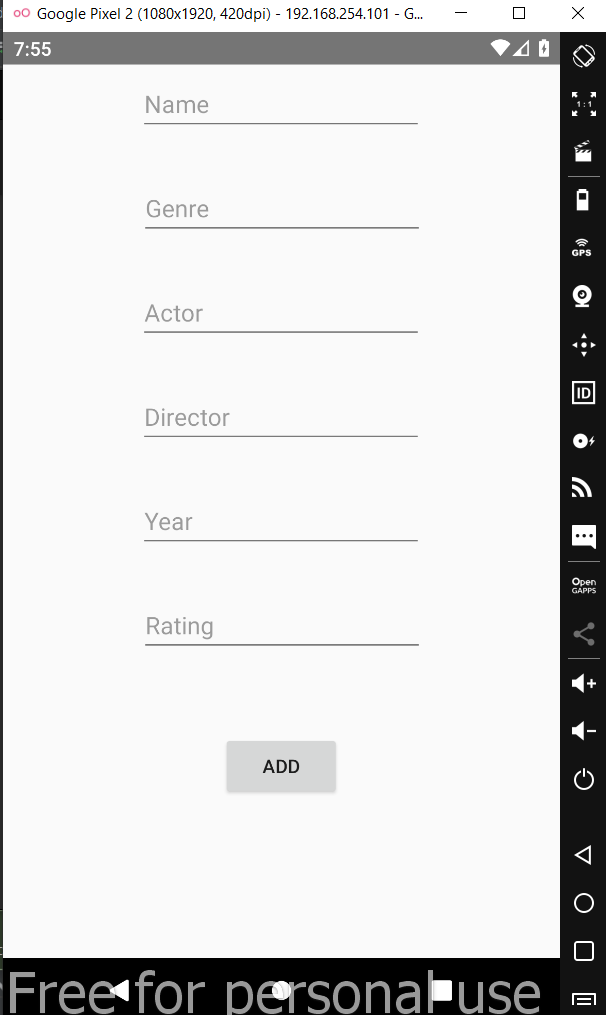
*Начален екран*

**

Да съдържа списък със всички въвеждани филми. За всеки филм се извежда име и оценка, дадена му от потребителя. В горния десен ъгъл да има бутон за добавяне на филм. При избор на филм да се извежда цялата информация за филма. В долния ляв ъгъл да има бутон за изтриване на филма, а в долния десен бутон за редакция, който да води към съответния екран. Екрана за корекция на данните да позволява редактиране на всички атрибути на филма. Отново съхранението на редакциите да става едва когато данните са напълно коректни.



*Екран за добавяне на нов филм*

**

Да съдържа полета за въвеждане на име на филма, година на издаване, режисьор, актьори, жанр и оценка, дадена му от потребителя. За избор на оценката да има скала от 0 до 10. В долния ляв ъгъл да има бутон за изтриване на филма, а в долния десен бутон за редакция, който да води към съответния екран.

3.**Обхват на разработката**

Разработеният софтуерен продукт трябва да удовлетворява очакванията на потребителите и да им предоставя възможност за улеснено използване на заложените функционалности. Потребителите ще имат на разположение продукт, който им позволява да съхраняват информация за вече гледани филми, давайки им рецензия, по която в бъдеще да сортират предпочитани за повторно гледане филми, правейки листинг на любими филми. Въведената допълнителна информация като актьори, жанр, година и режисьор дават допълнителни възможности за филтриране и извеждане на справки.

# **4.Изисквания към програмния продукт**

***4.2.Функционални изисквания – системата трябва да предоставя следните възможности:***

***4.2.1. Модул за въвеждане на информация за филм***

- възможност за въвеждане на име;

- възможност за въвеждане на актьори;

- възможност за въвеждане на режисьор;

- възможност за въвеждане на година;

- възможност за въвеждане на жанр;

- възможност за въвеждане на рейтинг

### ***4.2.2. Модул за търсене на филми, според определени критерии***

- възможност за търсене по име;

- възможност за търсене по жанр;

- възможност за търсене по актьори;

- възможност за търсене по година;

- възможност за търсене по режисьор

***5.Система за контрол на версиите***

[Система за контрол на версиите](http://bg.wikipedia.org/wiki/Система_за_контрол_на_версиите) се ползва за записване на промените по даден файл или колекция от файлове. Тя позволява да се запази история на промените, да се върне предишна версия, да се добави описание на промените.

**Git**

За разработването на софтуерния продукт е използвана децентрализираната система за контрол на версиите – Git. Git е бърза, ефективна и разполага със система за нелинейна разработка. С Git много потребители могат да работят едновременно по един и същ файл, без това да доведе до хаос в съдържанието на файла. Предимства:

- скорост;

- опростен дизайн;

- мощна поддръжка за нелинейна разработка;

- напълно разпределена работа.

Недостатъци:

- състои се от много команди и опции и е нужно високо ниво на разбиране, тъй като някои команди и аргументи са непоследователни до някаква степен;

- не достатъчно добра поддръжка под Windows;

- слаби инструменти за бинарни файлове;

- не поддържа празни директории;

- липса на ограничение за контрол на достъпа;

***6.Система за управление на проекта***

Trello е безплатно уеб приложение за управление на проекти, подпомагащ увеличаването на производителността при изпълнение на колективна работа. Trello използва методологията Канбан. To предоставя дъски (boards), списъци (lists) и карти (cards), за да се организират и приоритизират проектите. Най-общо една дъска представлява един проект. Към нея може да се изградят няколко на брой списъка – например TODO, IN PROCESS, DONE. В тези списъци се поставят задачи – карти.

Предимства:

- изключително интуитивен софтуер за управление на проекти;

- бърз поглед на подреждането на картите позволява на потребителите да знаят докъде е стигнал проекта и какви са задачите им за следващия;

- на предната част на картата може да се напише основната идея. На “гърба” обаче може да се попълнят различни данни, отнасящи се до конкретната задача, като всичко това може да бъде организирано лесно и визуално.

Недостатъци:

- простотата и минимализмът не позволяват проследяване на проектите в по-дълбока детайлност;

- не е подходяща система за управление на сложни проекти;

- няма възможност да се видят задачите, подредени по потребител или краен срок.

**7.Методология на разработване**

**Екстремно програмиране, Extreme Programming, XP**

Методологията на разработване, която сме избрали е Екстремно програмиране, тази методология спада към Гъвкавите технологии.

Причината за избора на тази методология е, че тя е предназначена за малки и средни колективи, разработващи софтуер при неясни или бързо променящи се изисквания, както и заради скоростното получаване на конкретен резултат. Тъй като проектът може да претърпи промени в бъдеще, тази методология е предпочитана пред стандартните методологии, при които цената за реализация на промените е скъпоструваща.

Екстремното програмиране притежава следните характеристики:

- кратки цикли;

- метод на инкрементално планиране(стъпка по стъпка);

- непрекъснато тестване като тестовете се задават от потребителя;

- непрекъсната интензивна комуникация;

- тясно сътрудничество м/у програмистите;

- ще могат да променят насоката на проекта по време на реализацията му без това да бъде свързано с големи разходи.

***8.Екип, роли и задачи***

**Екип**

1. Добромир Начев

2. Александър Стоименов

**Роли**

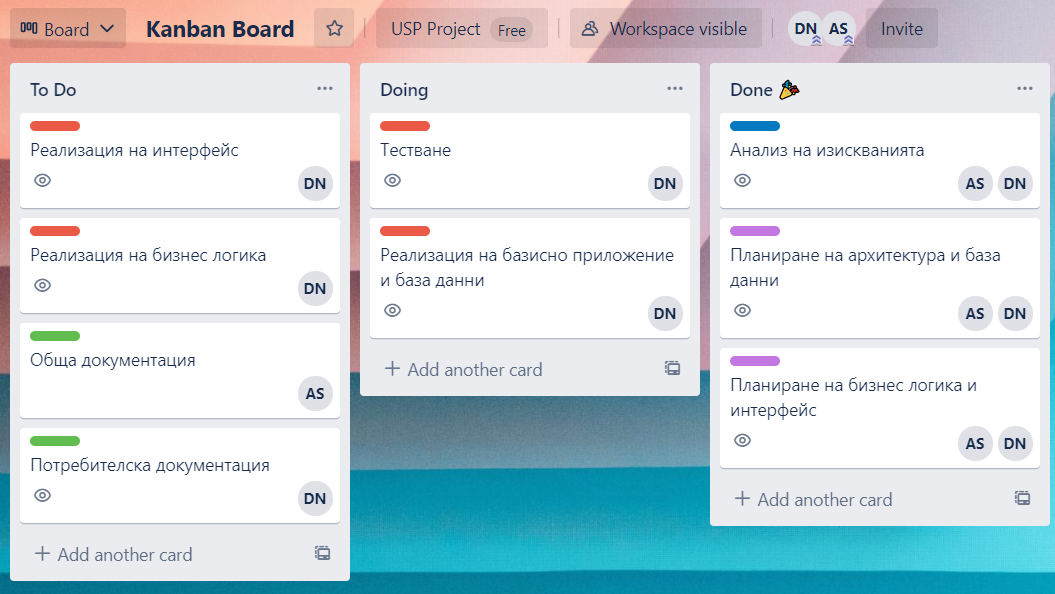
Добромир Начев – мениджър, разработчик, дизайнер, тестер

Александър Стоименов – автор техническа документация, дизайнер, тестер

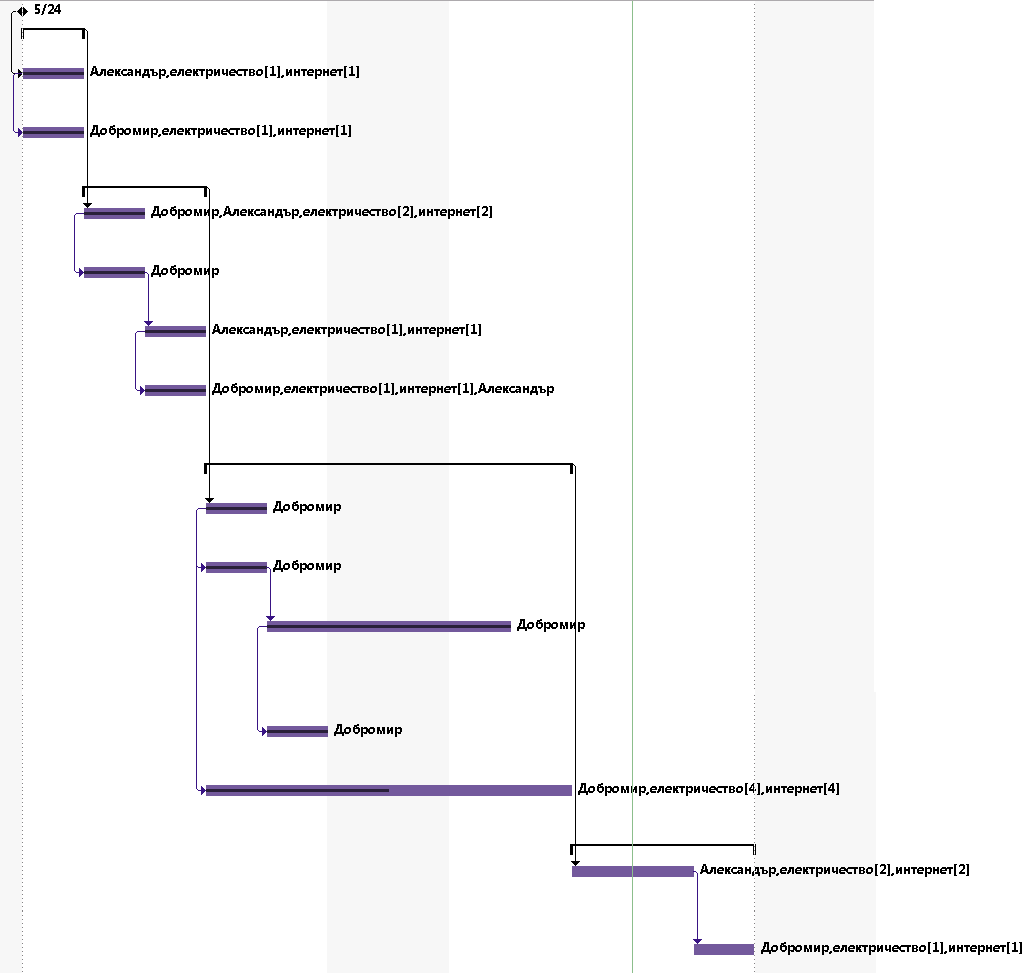
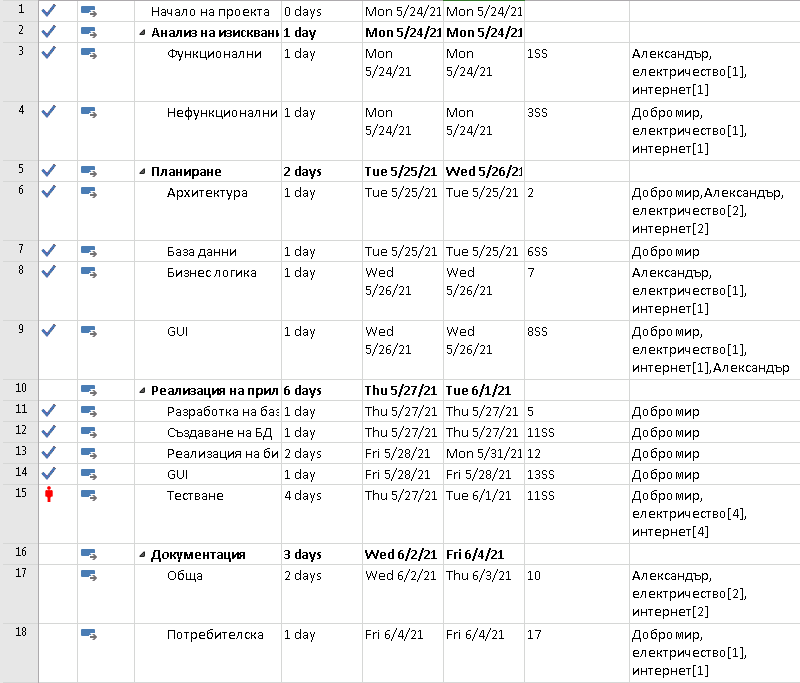
**Задачи**

***9.Time-line***

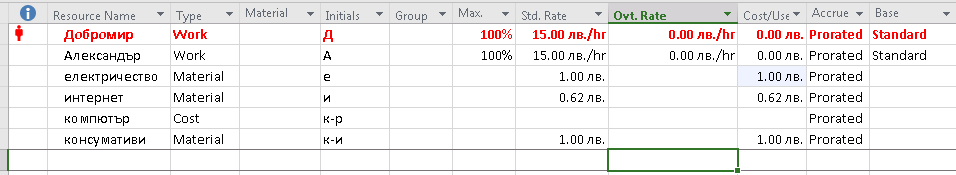
**Trello taskboard**



***10. Диаграма на Гант***



***11. Ресурсен лист***

***12. Диаграма на свършената работа***

***Правила***

1. 1 филм може да има само едно заглавие
2. 1 филм може да има много жанрове
3. 1 филм може да има повече от един режисьор
4. 1 филм моце да има повече от един актьор
5. 1 филм може да има само един рейтинг

***Процедурни правила***

1. Всичко полета трябва да са попълнени при записа на нов филм
2. Рейтингът не може да е по-малък от 1 и по-голям от 9
3. Годината не може да е по-малка от 0