**Курсов проект**

**по**

**Управление на софтуерни проекти**

**Изработили:Даниела Иванова,**

**Христо Фотакиев,**

**Жени Минчева**

**Факултетен №: 17621834,**

**17621723,**

**17621768**

**Специалност: СИТ**

**Курс: III**

**Група: 3б**

Съдържание

​​

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​

​​

Техническо задание на проекта

Бюджет

Бюджета за проекта е съставен въз основа на всички разходи по анализирането, проектирането, разработването и документирането на проекта. Анализа на изискванията е много важен етап. Той включва анализа на функционални и нефункционалните изисквания към проекта. На него са отделени два работни дни, които се изпълняват паралелно от двама участници на екипа. За планирането на проекта се поставени четири работни дни. Този етап отнема повече време, за да избегнат евентуални бъдещи грешки, чието поправяне би струвало все повече спрямо в кой период на работата по проекта са открити. В този етап се проектира архитектурата на приложението, която трябва да е подходяща за евентуално бъдещо развитие и позволяваща бързо и лесно адаптиране нови членове към екипа. От нея също така зависи и добре написаният код, при който липсва повторение и всичко е добре структурирано. За разработката са отделени пет работни дни, защото след като проектирането е минало реализацията е ясна и не се очаква да времеотнемаща. Спрямо обхвата на приложението за тестването е отделен един работен ден, а за изготвянето на двете документации са отделени три работни дни.

Спрямо представената информация общата стойност на проекта възлиза на **2,041.25**лв. Тя включва използването на ресурси на екипа в случая компютри (по 100 лв на човек) и добавени 15% за евентуални бъдещи процеси отнасящи се към поддръжката и на приложението.

Предмет на техническото задание

Изискванията към приложенито включват показване на прожектирани от едно кино филми и избор за покупка на билет за дадена прожекция, включвайки избор на място. При покупна на билет се показват на потребителя свободните позии в кино салон спрямо желаната от него прожекция. Всички екрани и техните функционалности са описани подробно в документацията. Приложението има заявено изискване за показване на най-популярен филм. Популярността на филм се изчислява чрез броя на заетите места за неговите прожекции.

Начален екран

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

Екран за вход

A screen shot of a movie ticket

Description automatically generated

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

A screen shot of a movie ticket

Description automatically generated

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

A screenshot of a movie

Description automatically generated

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

A screen shot of a screen

Description automatically generated

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, които възложителя е дал, както 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 избере прожекция на даден филм, която иска да посети. При избиране на прожекцията потребителят може да избере тип на билетите които иска да закупи и брой билети за всеки тип. Предоставя се възможност на потребителя да избере местата, които иска да заеме, като вече заетите места са забранени за избиране. В приложението е добавен и функционал за показване на най-популярен филм.

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 за въвеждане на данни, които да попречат или да поредят на системата. Всички входни точки са валидирани. Продуктът е десктоп приложение на програмния език Java. Той е използван защото е масов, поддържан и има възможност за пускане на всякакви операционни системи.

За запазване на информация е използвана MySQL база, която е избрана заради високата си производителност, лесното си използване .

Приложението има минимални изисквани към хардуера, на който трябва да бъде инсталирано. Относно софтуера единственото изискване е да има  инсталирана Java, която е нужна, за да се създаде виртуалната машина, на която се пускат всички Java приложения.

Обосновка на избора на система за контрол на версиите

Причина

Една от основните причини за изборът на Git беше фактът, че Git позволява на всеки от нас да притежава собствено repository, в което да работи, по който начин можем едновременно да работим върху един и същи файл. Git позволява 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, позволявайки ни да разработваме различни допълнителни идеи, без да възпрепятстваме главния проект.

Трудности при използването

Проследяването на комит, който причинява бъг след голямо сливане отнема голям период от време. Реализирането на функционалността на branch-овете и разрешаването на конфликти между тях е сложно.

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, като в при реализирането на нашия проект използвахме TurtoiseGit

Съпоставка

При използването на Git има възможност при промяна или добавяне на повече файлове промените по основната версия на проекта да се извършат на стъпки, като се commit-ват само по няколко файла наведнъж, като се включи и детайлен коментар за извършените промени. При гит се използват repository-та(едно общо на централизиран 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, като може лесно да се премине към предходна версия на проекта, за разлика от останалите системи за контрол, където всичко се намира на един централен сървър и мениджърите на проекта раздават роли на хората асоциирани с него, които роли дават ограничен достъп до определени файлове и функции.

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 дава възможност за проследяване на прогреса. Приложението също така праща известия при възникнали промени - нови или завършени задачи, като улеснява комуникацията между членовете на екипа. Trello дава възможността за общи дискусии, прикрепяне на коментари към отделните задачи и лесно споделяне на файлове.

Трудности при използването;

Тrello няма функция, която да показва връзките между отделни задачи - коя задача трябва да се реализира, за да може да се реализира друга. Trello не дава възможност за проследяване на време и ресурси, за да се следят по-лесно крайни срокове и вместване 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 към Trello и добавя хората от екипа в него. Избира се възможността на видимост на проекта - публичен или скрит(само за участниците в него).

мобилна/desktop/web версия;

Trello също така има мобилна и desktop версии, които могат да бъдат изтеглени/закупени чрез съответно Google Play и Microsoft store (App Store за потребителите на Apple(IPhone, IPad, Mac)).

Използвани функционалности.

Добавяне на допълнителни изисквания за тип на файл към всяка задача и коментари за срок на изпълнение. Проследяване на история на промените към задачи, кой коя задача е поел, кога я е изпълнил, коя изпълнява в момента, колко задачи са изпълнени и колко остават.

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, но е трудно.

Предимства:

* Добре структуриран подход
* Последователните етапи и дейности в тях са точно определени
* По-бързо изпълнение на проекта
* Избягване на отклонения на обхвата на проекта поради неразбиране на определена част от изискванията
* Във всяка фаза се виждат конкретни резултати
* Този подход води до точни срокове и бюджети

Недостатъци:

* Малката гъвкавост на проекта
* Трудното връщане към предишен етап, който вече е завършил

Съпоставка със спираловидния модел:

Водопадният модел е известен класически, линеен модел, който следва последователни етапи, докато спираловидният модел следва еволюционния 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** |
| 1.Моделът е предпочитан от клиентите | Моделът е предпочитан от разработчиците |
| 2.Грешки и рискове се идентифицират и отстраняват след приключване на съответният етап | Грешките и рисковете са идентифицирани и отстранени по-рано |
| 3.Добре приложим при малки проекти | Добре приложим за големи проекти |
| 4.Необходимо планиране в ранен етап | Не е необходимо планиране в ранен етап |
| 5.Много малка гъвкавост | Гъвкавостта за промяна не е трудна |
| 6.Сравнително евтин модел | Много скъп модел |
| 7.Необходимо е задълбочено разбиране на изискванията от самото начало | Изискванията могат да се добавят в новите итерации, ако е необходимо |
| 8.Произвежда единичен краен продукт в края на процеса | Генерира груб работещ продукт при всяка итерация |

Описание на стъпките по реализацията:

Екип

Даниела Иванова

Христо Фотакиев

Жени Минчева

Определяне на роли

Мениджър на проекта: Жени Минчева

Софтуерен архитект: Жени Минчева, Христо Фотакиев

Дизайнер на потребителски интерфейс: Даниела Иванова

Разработчик на бизнес логиката: Даниела, Христо

Технически писател: Жени, Христо

Тестер: Даниела

Разделяне на задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер на задачата | Опсание | Предшестваща задача | Време | Изпълнител |
| 1 | **Анализ на изискванията** |  |  |  |
| 2 | Функционални |  | 1 day | ДИ |
| 3 | Нефункционални |  | 1 day | ЖМ |
| 4 | **Планиране** |  |  |  |
| 5 | Архитектура | 1 | 1 day | ХФ |
| 6 | Базата данни | 1 | 1 day | ДИ |
| 7 | Бизнес логика | 1 | 2 days | ХФ, ДИ |
| 8 | Потребителски интерфейс | 1 | 1 day | ДИ |
| 9 | **Разработка** | **4** |  |  |
| 10 | Създаване на базата | 8 | 1 day | ДИ |
| 11 | Записване на информация в базата | 10 | 1 day | ЖМ |
| 12 | Потребителски интерфейс | 8 | 2 days | ДИ |
| 13 | Бизнес логика | 13 | 3 days | ХФ |
| 14 | Цялостно тестване | 13 | 1 day | ДИ |
| 15 | **Документиране** | **9** |  |  |
| 16 | Обща документация | 14 | 2 days | ЖМ |
| 17 | Потребителска документация | 15 | 1 day | ХФ |

Определяне на срокове

Начална дата: 26/03/2020

Анализ на изискванията: 26/03/2020

Планиране: 27/03/2020

Разработка: 02/04/2020

Тестване: 09/04/2020

Документиране: 14/04/2020

Програмни средства за реализацията

Език за програмиране: Java

Платформа за визуализация: JavaFX

База данни: MySQL

IDE: IntelliJ IDEA

Допълнителни помощни средства: Scene Builder, XAMPP

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 | Опсание |
| 0 | Начало |
| 1 | Изискванията са анализирани |
| 2 | Планирани са архитектурата и базата |
| 3 | Планирана е бизнес логиката |
| 4 | Проектиран е дизайна |
| 5 | Базата е създадена |
| 6 | В базата е записан информация и потребителския интерфейс е направен |
| 7 | Бизнес логиката е реализирана |
| 8 | Тестването е приключило |
| 9 | Общата документация е завършена |
| 10 | Потребителската документация е завършена |

Timeline

Microsoft Project

Календар

A blurry image of a computer screen

Description automatically generated

Диаграма на Гант

Ресурсен лист

Проследяване на завършеност на задачите

Диаграма на свършената работа

Диаграма на бюджет

Таблица на бюджет

Диаграма на ресурси

Бизнес правила:

Потребителят няма право да закупи билет за филм без регистрация.

Един филм може да има само едно име.

Всеки филм има по точно 2 прожекции дневно.

Потребител не може да закупи билет за място, за което друг потребител вече е закупил билет.

Потребител не може да закупи билет за една и съща прожекция за същото място два пъти.

Един потребител може да купи повече от един билет за една и съща прожекция на един филм.

Един филм може да има повече от един режисьори.  

Обекти и техните атрибути

Movie:

* movieId – уникален идентификатор на филма
* movieName – име на филма

Projection:

* id\_projection – уникален идентификатор на прожекцията
* id\_movie – идентификатор за филма

User:

* name – име на потребителя
* password – парола 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.