Технически университет – Варна

Картина, която съдържа текст

Описанието е генерирано автоматичноКатедра „Софтуерни и интернет технологии”

Документация

по Управление на софтуерни проекти

**Изготвили***: Стела Николаева ф.н. 18621771,* **Проверил**: ………….

*Теодора Николова ф.н. 18621781, / ас. Д. Димитров /*

*Никола Димов ф.н. 18621777,*

*СИТ 5а група, 3 курс*

# Техническо задание на проекта

## Бюджет:

## Изготвянето на бюджета следва логическата последователност на предвидените в проекта дейности. Направено е остойностяване на всяка дейност въз основа на цените, валидни в момента на изготвяне.

Определянето на бюджета включва контролиране на разходите, което е наложително за воденето на успешен бизнес. С въпросния бюджет, компанията има ясна представа колко ще струва всяка дейност. Имайки предвид тази базова линия, можем да сравним действителните разходи за всяка дейност в рамките на проекта с прогнозирания бюджет, за да определим дали проектът е над бюджета, под него или по цел. Възможността да се види това по време на прогреса на проекта помага на ръководителя да определи дали трябва да направи някакви промени в него, за да гарантира, че остава на целта.

Има две основни цели за създаване на бюджета на проекта: получаване на финансиране и контрол на бизнес разходите. Важно е да се информират заинтересованите страни в бизнеса, колко финансиране би било необходимо, за да завършване на проекта. Заинтересованите страни в бизнеса могат да включват инвеститори, управленски екип на компанията или други собственици на бизнес. Създаването на бюджет варира от компания до компания и от проект до проект.

Спрямо представената информация общата стойност на проекта възлиза на 6 000лв. Тя включва използването на ресурси на екипа, в случая компютри и добавени 20% за евентуални бъдещи процеси отнасящи се към поддръжката и на приложението.

## Предмет на техническото задание:

Предмет на този документ са изискванията на възложителя относно желаната функционалност на системата. Разработваната система за работа с рецепти от „готварска книга“ предлага съхранение на данни за рецепта, което включва стъпките и съставките. Съдържа се следните екрани:

* Начална страница – визуализира наименованието на сайта
* Страница с рецепти – в нея ще видите всички рецепти, които книгата предоставя. Ако желаете имате право да добавите своята рецепта, да разгледате в детайли вече добавена, да редактирате някоя или да я изтриете.
* Страница със съставки – визуализират се всички съставки, използвани за създаването на рецепти. Възможни „CRUD“- операции
* Страница със стъпки – визуализират се всички съставки, използвани за създаването на рецепти. Възможни „CRUD“- операции
* Страница, която дава възможност за свързване на вече съществуващи рецепти и съставки. Възможни „CRUD“- операции

## Обхват на разработката:

Разработеният софтуер трябва да удовлетворява очакванията на потребителите и да им предостави възможност за улеснено използване на функционалностите. Също така трябва да покрива всички изисквания, които възложителят е дал, както и да предоставя удобен интерфейс за работа.

## Изисквания към програмния продукт:

### Общи изисквания

Софтуерният продукт трябва да обхваща всички възложени случаи, както и поведението на системата да е предсказуемо. Продуктът представлява уеб-приложение, което е изработено на ASP.Net Core. За запазване на информация е използвана SQL Server, избран заради високата си производителност и лесното си използване .

**Изисквания:**

Операционна система: Windows 7 или по-нов.

Хардуер:

* Процесор: Базова честота от 1.8 GHz или повече.
* RAM: 4 GB или повече.
* Дисково пространство: 75 MB или повече.

Софтуер:

* Visual Studio (или друго подобно).
* Уеб браузър.

### Функционални изисквания – системата трябва да предоставя следните възможности:

Страница за въвеждане на информация за рецепта:

* Търсене на дадена рецепта;
* Добавяне на нова рецепта;
* Редактиране не вече съществуваща рецепта;
* Изтриване на рецепта;
* Преглед на детайлите на дадена рецепта.

Страница за въвеждане на информация за съставка:

* Добавяне на съставка към рецепта;
* Изтриване на съставка;
* Редактиране на съставка.

Страница за въвеждане на информация за стъпка:

* Добавяне на стъпка към дадена рецепта;
* Изтриване на стъпка;
* Редактиране на стъпка.

## Изисквания към дизайна:

* Приложението трябва да бъде с максимално опростен дизайн;
* Информацията трябва да бъде лесно достъпна за крайния потребител.

### Изисквания към документирането:

* Приложението трябва да има ръководство за потребителя

## Системна документация:

Изпълнителят трябва да подготви следната документация:

* Описание на структурата и принципа на работа със системата;
* Ръководство за конфигуриране контрол и поддръжка.
* Документите трябва да бъдат на български език и да бъдат представени както на хартиен, така и на електронен носител.

## Софтуерна поддръжка:

Поддръжка на уебсайт е услуга, при която ние поемаме цялата техническа работа по сайта Ви - регулярни ъпдейти на системата, защита от хакерски атаки, регулярни бекъпи и още много други. Освен това, всички наши клиенти може да заявяват неограничен брой промени по текущата визия, които са включени в месечната такса. Обновяването на съдържание като рецепти , продукти, или стъпки - само трябва да ни изпратите заявка по [email](mailto:recipe.book@gmail.com). Цената за уеб сайт поддръжка започва от 23 на месец и варира според големината на проекта и задачите, които трябва да се изпълняват.

## *Инсталиране и тестванe:*

Няма нужда от инсталация. Тестове се извършват по отношение на всички функционалности, както в потребителската част, така и в администраторската. При възникване на аномалии в поведението на програмния продукт, екипът от разработчици ще следи и отстранява грешки.

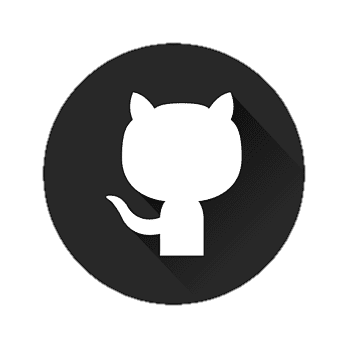
## Картина, която съдържа текст, екранна снимка, монитор, екран Описанието е генерирано автоматичноСтруктура на системата:

# Система за контрол на версиите:

Системата за контрол на версиите е механизъм, чрез който по-лесно и ефикасно се управлява работата по даден софтуерен проект. Тя използва едно централно хранилище, в което се съхраняват файловете на проекта. Когато някой започне работа по проекта, той създава копие на файловете от това хранилище на своята система и работи с тези копия. В процеса на работа потребителя може да изпраща към хранилището направените от него промени и да получава промените, направени по проекта от другите хора в екипа. След обновяване на хранилището, в него се запазват старите версии на променените файлове. Така тези файлове могат да бъдат възстановени при необходимост. Системата за контрол на версиите също така следи за конфликти – различни промени на различните потребители, които ползват хранилището.

**Git**

За разработването на този софтуерен продукт е използвана децентрализираната система за контрол на версиите **– Git.** Създадена е през 2005 с отворен код. От тогава насетне изключително голям набор от програмисти разчитат на нея за създаването на комерсиални и некомерсиални проекти.

**GitHub**

И в по-голяма точност, използвали сме GitHub. GitHub е най-голямата онлайн услуга за съхранение на Git хранилища и централна точка за съвместна работа на милиони разработчици и проекти. Голям процент от всички Git хранилища се хостват в GitHub и много проекти с отворен код ползват платформата като средство за проследяване на проблеми, преглед на код и много други неща. Услугата може да бъде както платена за частни проекти, така и безплатна за т.нар. проекти с общодостъпен код, като и в двата случая потребителите могат да ползват всички възможности на услугата.

## Причина за избора

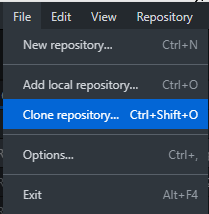
* Възможността всеки да има индивидуален достъп до файловете
* **Git** е най-популярната и широко използвана система за контрол на версиите
* **GitHub** е най-голямата онлайн услуга за съхранение на Git хранилища

## Трудности при използването

* Изисква се време за разбиране на командите и операциите, свързани с Git.
* Проблем при свързването със средата за разработка на програмен код.

## Настройки, свързани за работата с него

Препоръчително е да се изтегли GitHub Desktop, за по-голямо улеснение. От там се свързва с вече съществуващо хранилище и се прави копие на личния ни компютър.

Картина, която съдържа текст, екранна снимка, черен

Описанието е генерирано автоматично

След това в Visual Studio излиза менюто Git. Командите „pull“, „push“, „fetch“, „commit“ се изпълняват от въпросното контекстно меню в IDE-то.

## Съпоставка на избрания софтуер с други

**Предимството на Git пред SVN/CVS и TFS**

Използването на Git решава най-големия недостатък на другите две – в случай на повреда в централния или локалния компютър, всички данни на проекта могат да бъдат загубени, тъй като се намират на едно физическо място. Предимството на Git е, че тази система за контрол на версиите е децентрализирана поради, което проектът се сваля и копира локално на компютъра на всеки участващ потребител, като така се създава бекъп. В случай на необходимост, проектът може да се възстанови от локалното копие на някой от потребителите.

**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

# Система за управление на проекти:

## Причина за избора

Asana е софтуер, който предоставя възможност за цялостно наблюдаване на работата по даден проект. Това е и един от най-лесните начини да се проследява работа.

Asana е интуитивна система за управление на задачите в ИТ проекти. Тя работи изключително добре за екипи, които имат нужда от взаимодействие в реално време. По този начин платформата става изключително полезна за екипи за разработка на ИТ проекти, при които един, няколко или всички членове, работят на различни локации .

Asana може да персонализира въз основа на размера на вашия екип и гъвкавостта на проекта.

Други предимства на въпросния софтуер са:

* Лесна автоматизация на повтарящи се задачи
* Множество интеграции
* Архивиране на данни
* Наличен безплатен план
* Персонализиращ се работен поток
* Лесни за използване шаблони
* Изчистен дизайн

## Трудности при използването:

Поради използване на безплатна версия, има ограничени функционалности. Има лимит до 15 човека и 1000 задачи, въпреки че за малък отбор това е напълно достатъчно. Основният недостатък на Asana е невъзможността да се възлагат задачи на множество хора.

## Настройки, нужни за работата с него

Първоначално се изисква да се създаде акаунт и след това да създадем проект. Няма необходими настройки за работа. Свързване с отделните участници на екипа.

## Мобилна/desktop/Web версия

Използвана е уеб версията на софтуера. Има и мобилна версия, но тя е тежка и трудна за употреба

## Използвани функционалности.

Asana, освен опростения си дизайн, има и много функции, интегрирани в нея. Към момента са използвани:

* Задачи – лесно създаване и назначаване на задачи;
* Селекции и колони – Asana се настройва, за да отговаря на бизнес процесите в организацията;
* Шаблони за изграждане на проекти– Възможност да се създадат шаблони, за по-бързо създаване на следващи проекти;
* Подзадачи – разбиване на работата по изпълнение на задача на по-малки части;
* Крайни срокове – назначаване на крайни срокове за изпълнението на задачи

Други неизползвани функционалности са : възможност за прикачване на файлове от устройството, преобразуване на задача в проект

# Методология за разработка на софтуера:

## Причина за избора;

Водопадният модел се отличава с това, че дейностите протичат в процес, при който фазите на разработка на софтуера следват точно определен ред, който най-общо може да се опише по следния начин:

1. Анализ на изискванията
2. Анализ
3. Дизайн
4. Имплементация
5. Тестване
6. Интеграция
7. Поддръжка

Водопадният модел е добре структуриран подход. Процесите му протичат линейно и последователно и те са точно определени.. Това позволява по-бързо изпълнение на проекта. Всяка фаза на разработка, започва само когато предната е завършена, което води до конкретни резултати, точни срокове и бюджети и има избягване на отклонения на обхвата.

Наложените ни изисквания са осъзнати и ясно формулирани, ролите са организирани и ясно дефинирани и промяната на изискванията е малко вероятна и не е критична. Затова водопадният водел се прилага идеално.

## Предимства на подхода;

* принципите на действие на модела лесно могат да бъдат обяснени на потребителя;
* има добре структуриран подход към разработката;
* различните етапи и дейностите в тях са точно определени;
* планирането на проекта и изпълнението му по график е по-лесно;
* тестването на всеки етап осигурява ранното откриване на грешки, както и избягване на отклонения от обхвата на проекта поради неразбиране на определена част от изискванията на клиента;
* във всяка фаза се виждат конкретни резултати.

## Недостатъци на подхода, открити при изпълнението на задачата

* изключително трудно може да се върне екипа на предишен етап, който вече е завършил;
* има твърде малко гъвкавост по регулацията на обхвата на проекта и всяка промяна по него е трудоемка и скъпа.
* трудно се разграничават отделните фази на модела
* качественият контрол се извършва в края на процеса
* крайният потребител вижда системата едва след като е приключил процеса на разработка

## Съпоставка на подхода с други.

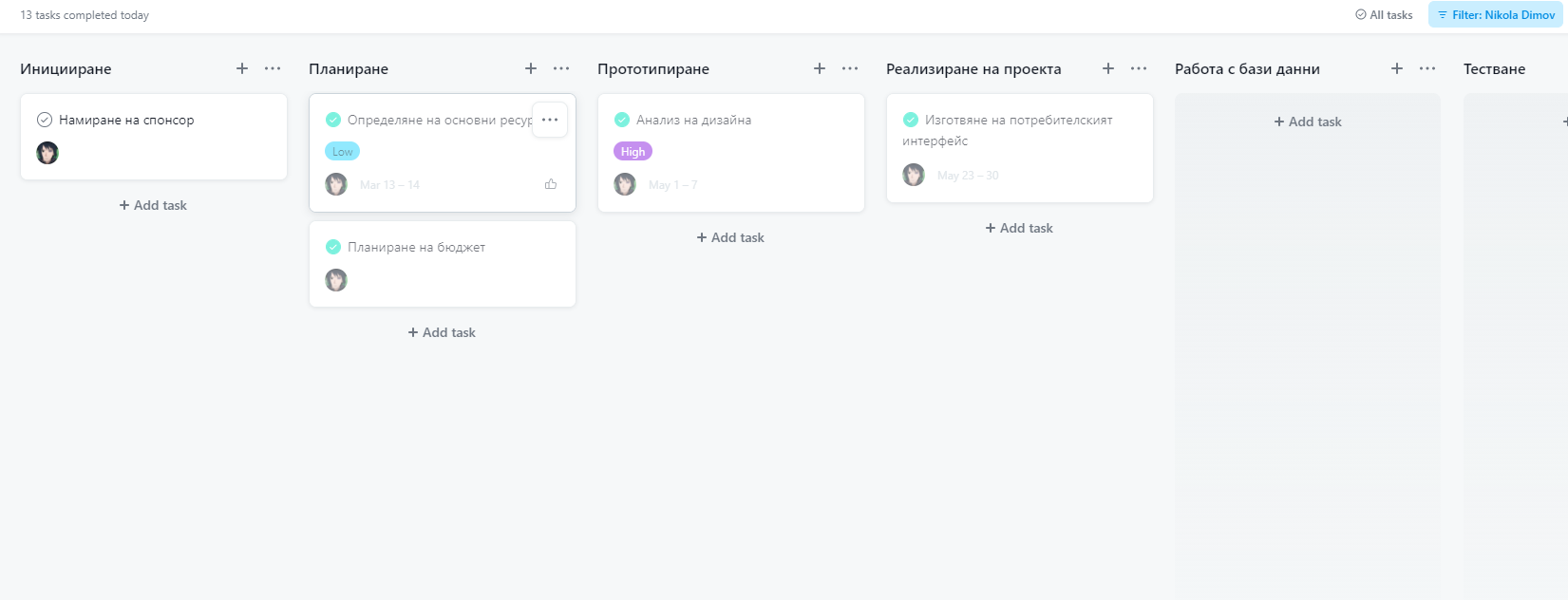
* **Спираловидният** и **Водопадният** модел имат почти идентични фази, но чрез анализ на риска в началото и проверката накрая, моделът е по-динамичен в сравнение с водопадния. Спираловидният модел може да се адаптира към малки промени в изискванията. Главната разликата е, че моделът на водопада се използва за по-малки проекти и проекти с ясни изисквания, докато спиралният модел се използва за големи, сложни проекти
* Основната разлика между **V-образния** и **Водопадния** модел е ранното планиране и тестване на първия, но е също толкова скъп, времеемък и не гъвкав. Има по-големи шансове за успех в сравнение с Водопадния, което се дължи на изготвянето на тест планове в началото на всеки ден по време на жизнения цикъл

# Описание на стъпките по реализацията:

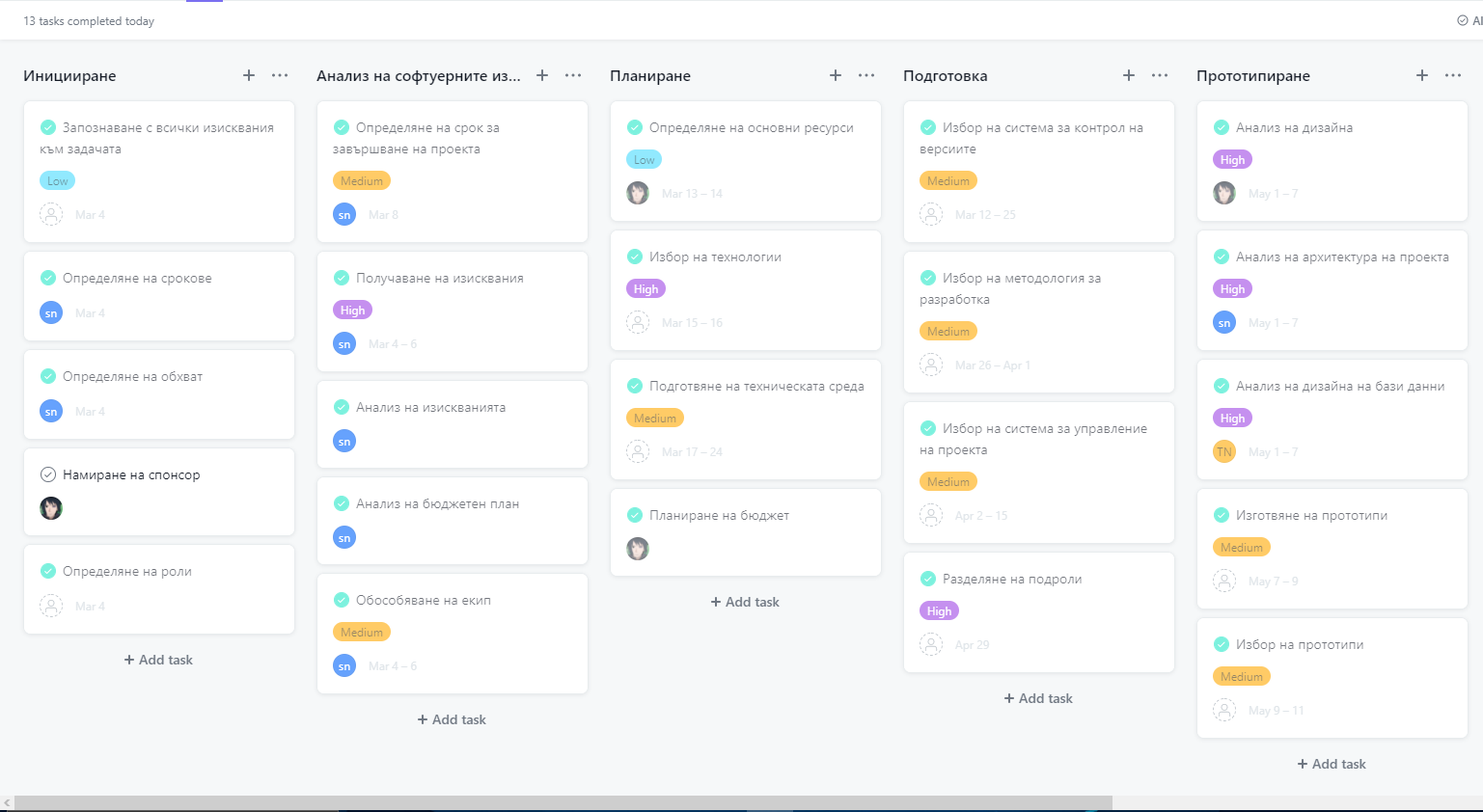
# Определяне на роли

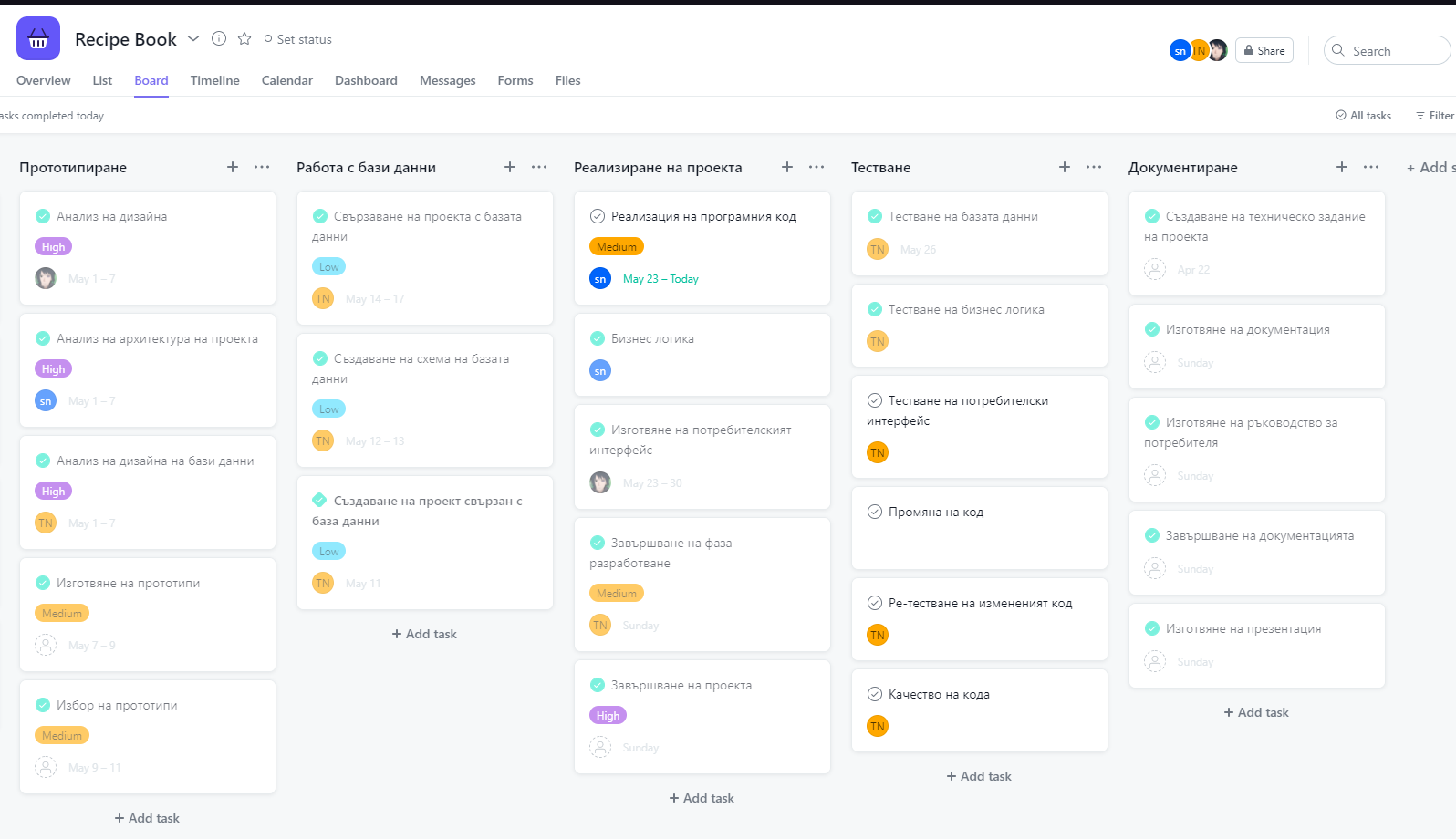
* Стела – Project Manager, Backend Developer
* Теодора – Database Developer, Quality Assurance
* Никола – Frontend Developer, Bussines Analyst

## Разделяне на задачи

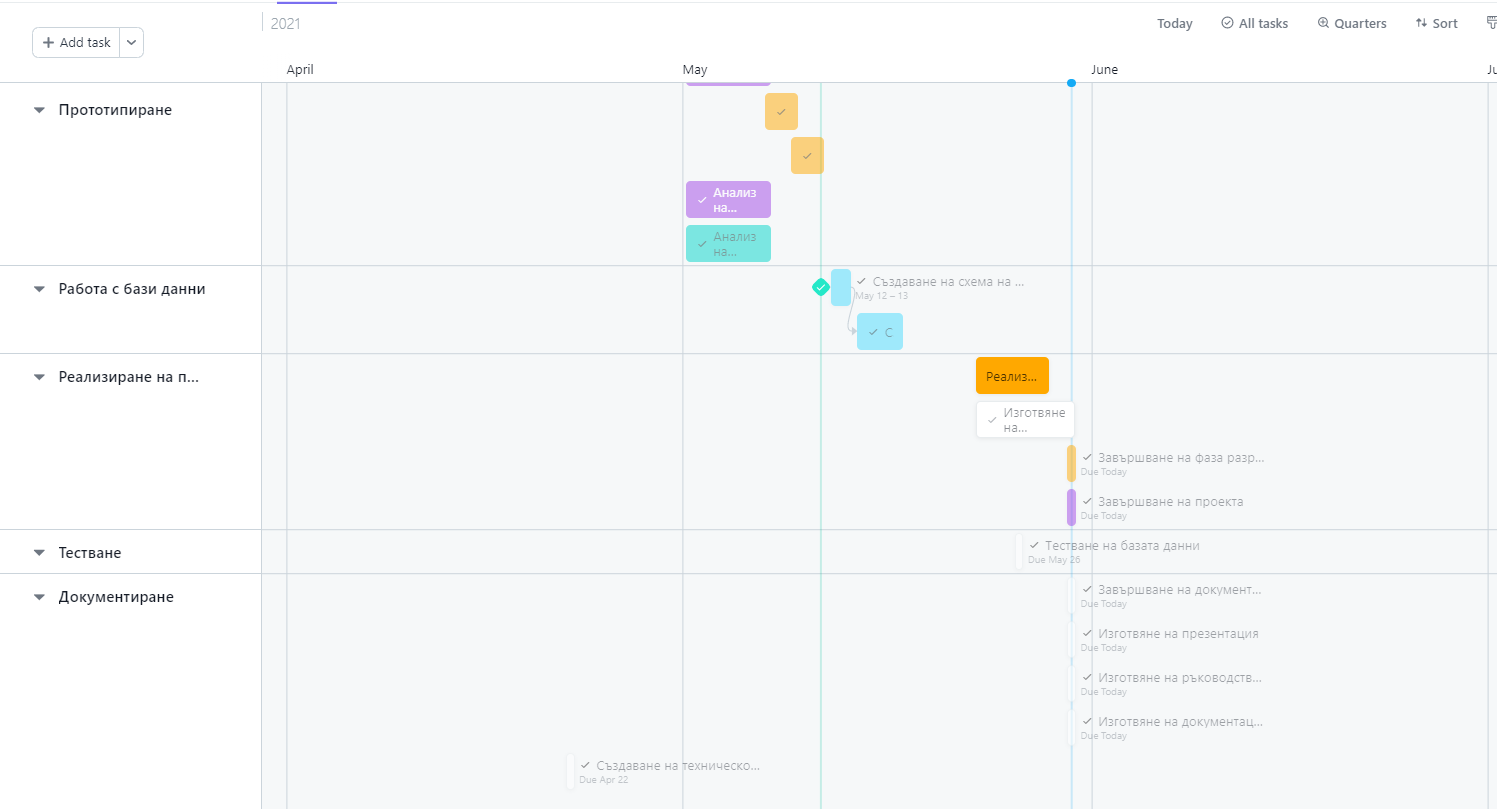


## Определяне на срокове за: проектиране, кодиране, документиране, интегриране, тестване





## Timeline (Асана)



Освен Asana е използвано и Trello, за отделните части на проекта.

Картина, която съдържа стрелка

Описанието е генерирано автоматичноКартина, която съдържа текст, шкаф, табло, мебели

Описанието е генерирано автоматично

## Отчети, генерирани от софтуера за управление на проекти, и анализ на отчетите

## 

# Изводи

За достигане на успешно завършен проект, удовлетворяващ потребителските изисквания и влизащ в бюджета и заложеното време за изпълнение, е нужно да се отдели достатъчно време за етапите на анализа на клиентските изисквания и проектирането. Използването на системи за управление на проекта значително повишава производителността на участниците в него. Системата за контрол на версиите дава възможност за независима разработка и лесно синхронизиране на извършената работа, както и създаване на back-up на проекта в хранилището.