

Архитектурен проект

TheDailyRoutine - Приложение за поставяне и
следене изпълнението на навици с цел
подобряване на ежедневието на хората

Автори: Атанас Юруков, Владимир Ганчев, Иван
Дражев, Огнян Барух, Петко Петков

2/11/2024

История на документа.....	3
Въведение.....	4
Обхват.....	4
Актьори.....	4
Използвани термини.....	4
Архитектурен обзор.....	5
Use-case диаграма.....	5
Логически изглед.....	6
Изглед на данните.....	7
Изглед на внедряването.....	8
Нефункционални изисквания.....	9

История на документа

Дата	Версия	Автори
2.11.2024	1	Атанас Юруков Владимир Ганчев Иван Дражев Огнян Барух Петко Петков

Въведение

Този документ цели да разгледа реализацията на приложение за поставянето и следенето на полезни навици в ежедневието на хората. В него са разгледани различните изгледи на проекта, неговите участници, както и функционалностите, които то ще предоставя.

Основна цел е приложението да може да бъде лесно за бъдещо развитие, тъй като темата за човешки навици и ежедневие е много комплексна и предразполага множество посоки за доразвиване на проекта.

Обхват

Този документ обхваща началната фаза на жизнения цикъл на проекта. Ще се разгледат основните функционалности, системи и модел на приложението, както и неговите различни изгледи.

Актьори

В приложението ще има потребителска роля, тъй като цели да бъде помощник на хората, който те да могат да персонализират максимално за техните нужди.

Освен потребители, ще има и администратори, които ще могат да обновяват списъка с предефинирани навици, от които потребителите могат да избират

Използвани термини

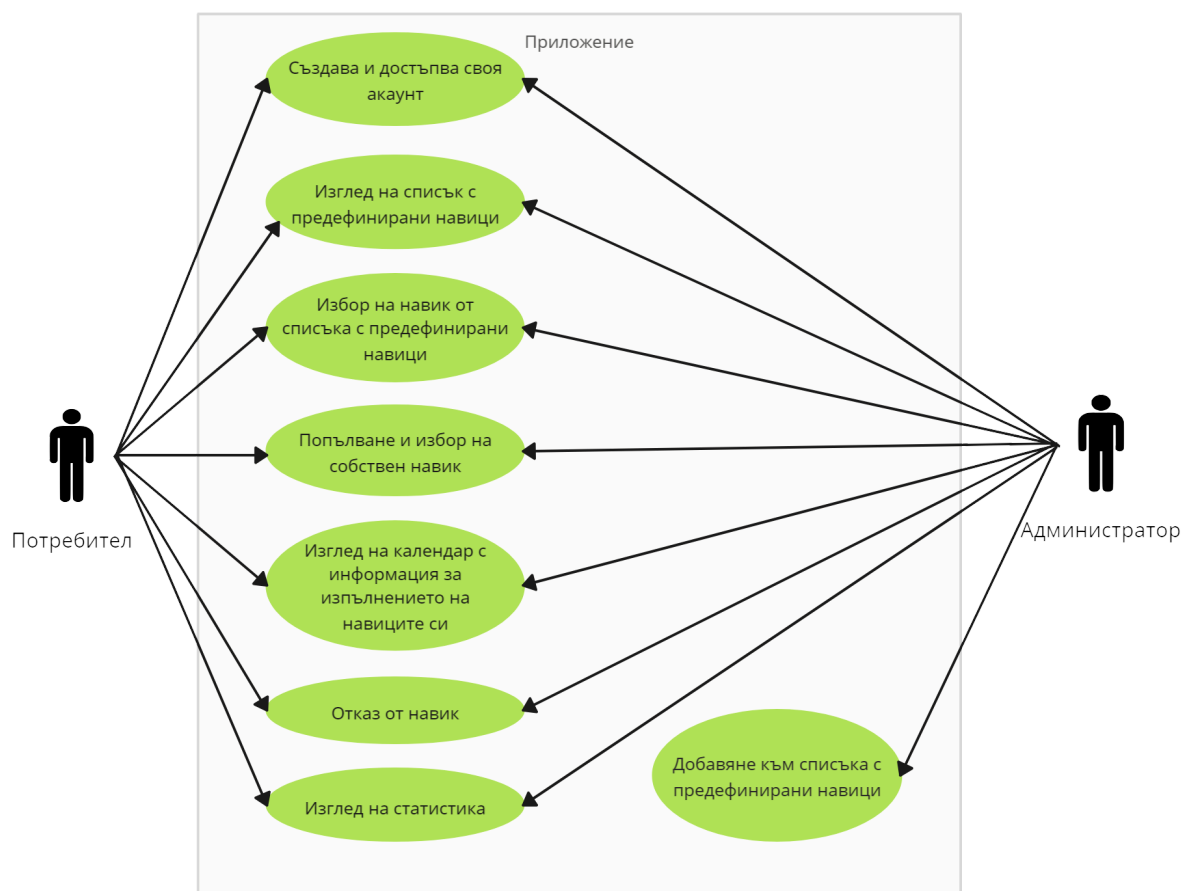
1. Frontend - визуализационната част на проекта
2. Backend - частта на проекта, която обработва данните
3. MVC - Model View Controller

Архитектурен обзор

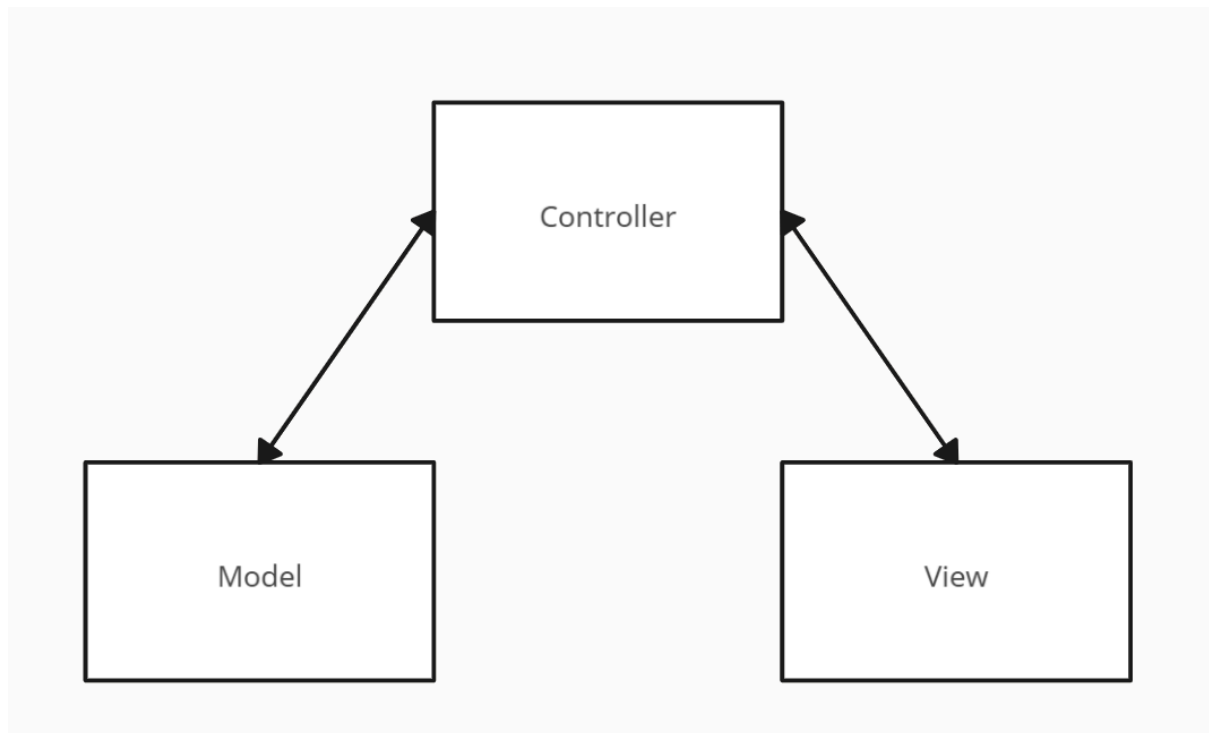
В следващите страници ще бъдат разгледани следните точки (в тази последователност):

- Use-case диаграма - операциите, които могат да извършват потребителите
- Логически изглед - диаграма, съдържаща архитектурно свързаните компоненти на приложението
- Изглед на данните - диаграма, съдържаща изгледа на данните, както и логическите връзки между тях. В случая на разглеждания проект, се говори за схема на база от данни.
- Изглед на внедряването - диаграма, съдържаща хардуерните ресурси за изпълнението на описаните в този документ процеси

Use-case диаграма



Логически изглед



Тази система ще бъде изградена на принципа MVC. Така можем да разделим приложението на няколко самостоятелни части, които ще си комуникират помежду си:

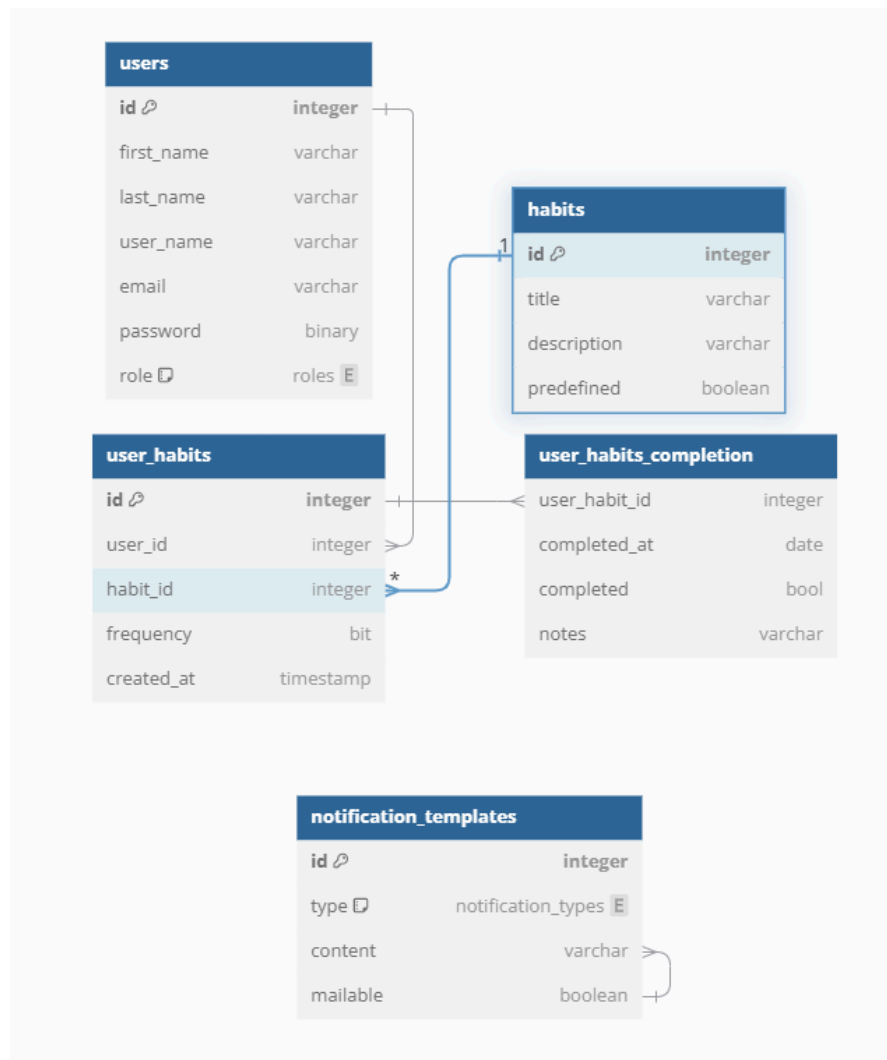
- **Model** - Абстрактно представяне на различните обекти в системата. Записва и обновява техните данните в базата данни.

- **Controller** - Използва Model на обектите, за да може да обработва различни данни, заявки и процеси.

- **View** - Показва нагледно данните на потребителя, като съдържа в себе си собствена логика за управлението на графичния интерфейс и неговите параметри.

Изглед на данните

Приложението разделя данните си по следния начин, което позволява лесно и подредено разпределение на данните:

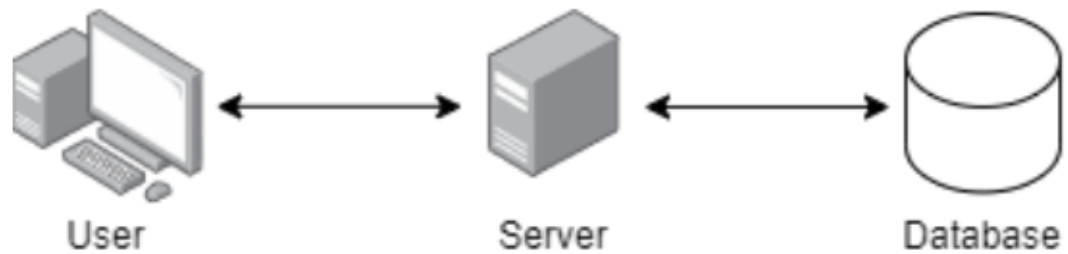


Приложението също и предоставя два типа данни - enumerations:

```
Enum roles {  
  User  
  Administrator  
}
```

```
Enum notification_types {  
  Greeting  
  Reminder  
  Achievement  
}
```

Изглед на внедряването



Тази схема показва комуникацията между различните устройства, които участват в системата:

- **User** - Потребителя, който ще се свързва към сървъра чрез заявки, с цел обработване на данни
- **Server** - Обработва данните и заявките, които постъпват в системата. Комуникира си с потребителските устройства и базата от данни с цел обновяване на данните.
- **Database** - Съхранява, записва, обновява и изтрива данните от системата.

Нефункционални изисквания

1. Разширяемост

Тъй като системата е имплементирана, използвайки MVC модела, нейната функционалност може лесно да се разшири до такава степен, че да включва и останалите етапи на дипломната разработка.

2. Използваемост

Системата предоставя графичен потребителски интерфейс, който да улесни потребителя в използването на функциите ѝ.