



**Пловдивски Университет
„Паисий Хилендарски“**

**Факултет по Математика и Информатика
Катедра „Софтуерни технологии“**

***Дипломна работа на тема:
„Система за управление на семейни активности“***

Дипломант: Миглена Додлева
Специалност: „Софтуерно инженерство“
Фак. №: 2001321045

Научен ръководител:
гл. ас. д-р Никола Вълчанов

Съдържание:

Въведение	4
ГЛАВА I	8
1. FamilyWall	8
2. Cozi Family Organizer	10
3. Family Tools App	11
ГЛАВА II	15
Регистрация и автентикация:	15
Управление на потребителите:	15
Календари:	15
Създаване на събития:	16
Задаване на различни права за достъп и йерархия в семейството:	16
Гледане и споделяне на календари:	17
Филтриране на събития:	17
Цветова диференциация на събитията:	18
Получаване на известия:	18
Покани:	18
Създаване на Семейство:	19
Управление на Списъци със задачи за изпълнение:	19
Споделяне на задачи и колаборация:	19
Сигурност и достъпност:	19
Приятелски семейства и споделяне между тях:	20
Сравнение на множество календари:	20
Семейна йерархична структура:	20
Работен процес по регистрация и автентикация:	22
Работен процес по създаване и управление на календари:	23
Работен процес по създаване и управление на събития в календара:	24
Работен процес по сравнение на множество календари:	25
ГЛАВА III	26
1. MS SQL Server	26
2. ASP.NET Web API и Entity Framework.Core	28

3. <i>React</i> и <i>Material UI</i> components	32
4. <i>JWT Token</i> и <i>Axios</i>	35
ГЛАВА IV	39
1. Структура на базата данни	39
2. Сървиси и тяхната значимост при организирането на архитектурата в рамките на проекта.....	46
3. Модули в системата.....	46
4. REST API	49
ГЛАВА V	53
<i>Наръчник на потребителя</i>	53
1. Вход и Регистрация в системата	53
2. Начална страница.....	56
3. Управление на профила	57
4. Управление на Семейства	58
5. Управление на Списъци със задачи	59
6. Управление на Календари и Събития	62
Заключение	66
Използвана литература:	67

Въведение

Развитието на информационното общество и глобалното навлизане на технологиите във всички сфери на обществения живот през последните години и новите технологични иновации, които не само влияят върху работния ни процес, но и на начина, по който живеем и управляваме ежедневието си. Тези нововъведения могат да бъдат използвани както в нашите работни практики, така и за организиране на домакинските ни задачи, и планиране на семейни дейности. Използването на цифрови календари за запазване на важни семейни дати или създаването на списъци за пазар в мобилни приложения или в чатове, са само пример за това как информационните системи са вече част от управлението на семейните ангажименти. Софтуерните платформи, които безспорно участват в живота ни, са важна част от планиране на действия в семейството. Приложението за споделен семеен календар е предназначено за отбелязване на важни събития, задачи, бележки, контакти и бюджет между членовете на едно домакинство. Координацията на общите дейности и ваканциите става трудна задача поради разнообразието от графици на членовете на семейството. Родителите и техните деца често са заети с различни ангажименти, което затруднява планирането на общи семейни почивки или събития. Много пъти, дори в семейна обстановка, е необходимо да интегрираме няколко платформи и приложения, създадени от различни производители. Платформите стават все по-незаменими за обикновения живот и са важна част от планирането на действия. От друга страна, използването на множество различни платформи и приложения понякога може да доведе до объркване и намалена ефективност. Няколко примера, какво се случва, когато използваме различни платформи:

1. Разпокъсаност в управлението на ежедневните задачи;
2. Противоречивост на информацията, която затруднява координацията между семейните устройства;
3. Загуба на време и ресурси в опити за координиране и актуализиране на общи планове;
4. Възможност за грешки при въвеждане и споделяне на информация, които могат да причинят повторение или пропускане на важни семейни събития.

Тези предизвикателства са причина за разделение не само в бизнес средата, но и в личния живот, което може да доведе до напрежение и комуникационни проблеми в семейството. В резултат на това, недоразуменията и липсата на координация водят до непълноценно използване на свободното време и загуба на възможности за качествено прекарано време.

Целта на тази дипломна работа е създаването на цялостен продукт, който да обедини всички необходими нужди на едно семейство. Той трябва да включва следните процеси:

1. **Управление на потребители;**
2. **Управление на семейства;**
3. **Управление на личен календар и събития;**
4. **Управление на семейни календари:** Централизация на всички събития и ангажименти на семейството;
5. **Графици на потребителите;**
6. **Управление на събития:** Автоматични напомняния за предстоящи събития или специални случаи - рождени дни, годишнини и други значими семейни събития;
7. **Управление на списъци със задачи:** Създаване и споделяне на списъци за пазар или домакински нужди, които всички членове на семейството могат да актуализират;
8. **Управление на задачи:** Създаване, назначаване и проследяване на индивидуални и общи задачи;
9. **Синхронизация на графици и свободно време;**
10. **Синхронизация на покани за членство в семейства;**
11. **Автоматично напомняне за бъдещи събития;**
12. **Съхранение на хронология на събития;**
13. **Контролиране на достъпа на членовете на семейството чрез управление на правата - определяне на роли.**

Тази система за управление на семейни активности обединява и оптимизира всички процеси в семейството, като улеснява кординирането на графици, споделянето на задачи, пазарски списъци, както и организирането на семейни събития в една централизирана

платформа. По този начин, всеки член на семейството може да добавя, разглежда и обновява общите планове, което значително намалява риска от грешки и помага за оптимизиране на свободното време. Система, която е централизирана, не само намалява необходимостта от различни приложения за всеки аспект на семейния живот, но и подобрява комуникацията и взаимодействието между членовете на семейството.

За да осъществим зададените по-горе цели за разработване на система за управление на семейни активности, трябва да се изпълнят следните основни задачи:

1. **Проучване и анализ на съществуващите решения:** Изучаване на пазара за налични системи и приложения, които вече предлагат решения за управление на семейни активности, за да се идентифицират техните силни и слаби страни. Този анализ ще помогне за определяне на нуждите, които не са удовлетворени адекватно и ще насочи разработката на новата система.
2. **Проектиране на функционалности на системата:** Разработка на детайлен план за функционалностите, които системата трябва да предоставя. Този етап включва дефиниране на интерфейса на потребителя, функции за управление на календари, задачи, събития и други семейни ресурси, които подкрепят ежедневната организация и координация на семейния живот.
3. **Анализ на използваните технологии и програмни езици:** Детайлно описание и оценка на технологиите, които са използвани за разработка на системата.
4. **Създаване на ръководник за потребителя:** Разработване на подробен ръководник, който да помага на крайните потребители лесно да навигират и използват системата.

Тези задачи ще осигурят създаването на всеобхватен и функционален продукт, който съчетава най-добрите практики и иновации в областта на технологиите. Така системата предлага на семействата инструменти, които значително улесняват организацията на техния семеен живот и повишава качеството и комфорта на ежедневието им.

Дипломната работа се състои от въведение, пет глави, заключение и използвана литература. Всяка глава е насочена към различен аспект от разработката и внедряването на системата.

Глава I представя обстоен обзор на съществуващите софтуерни приложения за управление на семейни задачи и дейности. Анализират се техните функционалности и ограничения, което помага за изграждане на основата на новата система, така че да отговоря по-ефективно на нуждите на семействата.

В **Глава II** са изложени изискванията към системата, обект на дипломната работа. Те са формулирани на базата на предварително идентифицираните потребности по време на проучването, изложено в първа глава. Определени са ключовите функции, които системата трябва да предлага.

В **Глава III** се разглеждат използваните технологии и програмните езици, избрани за разработката на приложението. Описани са предимствата и недостатъците на избраните платформи и инструменти, необходими за изграждането на надеждна и гъвкава система.

Глава IV представя технологичният дизайн на системата. В нея е включено детайлно описание на структурата на базата от данни, използвания архитектурен модел, модулите на разработка и интегрираните сървиси. Обсъдени са стратегиите за разгръщане (deployment) и основните технически реализации, които подпомагат функционалността и ефективността на системата.

Глава V предоставя наръчник за потребителя, който включва подробно описание на потребителският интерфейс и функционалностите на софтуерното приложение. Наръчникът е предназначен за подпомагане на крайния потребител в използването на системата и за достъп до всички ресурси, които му са предоставени.

ГЛАВА I

Обзор

Много софтуерни продукти, които улесняват управлението на домакинствата, се предлагат под формата на „софтуер като услуга“. Този подход позволява удобен достъп до тях през уеб браузър от всякакви устройства и по всяко време, без нужда от допълнително инсталиране. Въпреки това, някои приложения все още изискват директно изтегляне на мобилните устройства, за да предоставят всичките си функционалности.

Примери за такива системи са “FamilyWall”, “Cozy Family Organizer”, “Family Tools App”. Те предоставят възможности за автентикация с електронна поща, като всяко семейство може да изпраща покани за присъединяване, да се следят събития, графици и да се проследяват поставените задачи за деня.

1. FamilyWall



Системата FamilyWall е проектирана като всеобхватно приложение за организиране на семейния живот, което позволява ефективно координиране на членовете на семейството и техните ангажименти. Предлага функции като общ календар за цялото семейство, списъци, бюджетиране, проследяване на местоположението и сигурно пространство за споделяне на снимки и съобщения. Приложението предлага безплатна версия с основните функции и премиум абонамент, който включва допълнително пространство за съхранение, по-напреднало проследяване и повече инструменти за планиране.

Основните функции са:

- Проследяване на местоположението в реално време за сигурност на членовете на семейството;
- Персонализирани календари и списъци за по-добро управление на графици;
- Споделяне на снимки и видеоклипове в сигурна среда;
- Абонаментни планове.

Предимства:

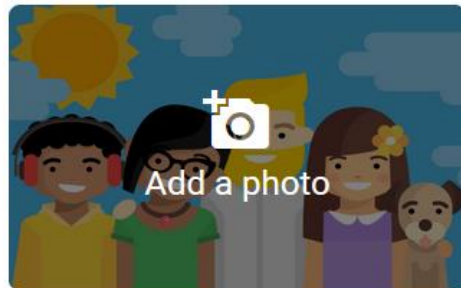
- Обширен набор от функции, покриващ различни аспекти на семейното управление;
- Допълнителни функции като рецепти и хранителни планове;
- Актуализации и известия в реално време, които подобряват безопасността и координацията.

Недостатъци:

- Някои напреднали функции изискват премиум абонамент;
- Интерфейсът на приложението може да бъде сложен за потребители, които използват само базови функции;
- При регистрация се задължава потребителя да създаде семеен кръг, което изключва възможността системата да бъде ползвана от потребители без семейство. Това бизнес решение ограничава достъпа на индивидуални потребители и води до загуба на бъдещи клиенти.



About your 1st Family Circle



Circle Name

What is your family name?

Next

Фиг.1: Интерфейс на регистрационната форма в приложението FamilyWall

FamilyWall представлява отличен избор за семейства, които търсят комплексен инструмент за управление на семейните задачи и дейности, осигуряващ набор от функционалности за подобряване на семейната координация и сигурност.

2. Cozi Family Organizer

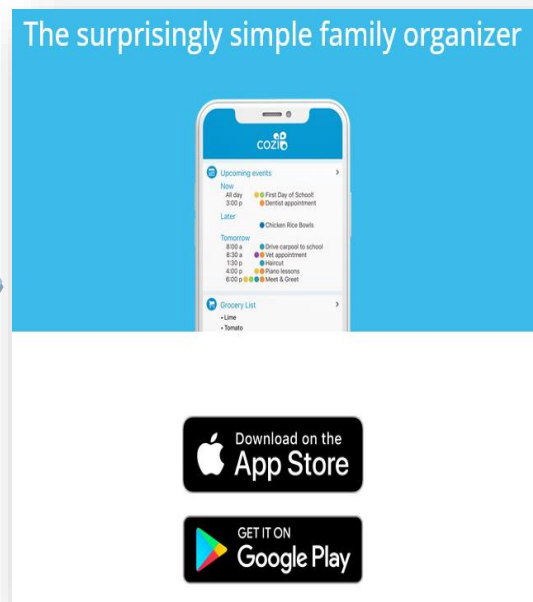


Приложението Cozi Family Organizer е създадено с цел организация на домакинства, предоставяйки функции за управление на графици, дейности и срещи. Известно е със своя приятен потребителски интерфейс и ефективност при управлението на сложни семейни операции. Тази система е идеална за модерни семейства, които търсят централизиран начин за синхронизация.

Основни функции:

- Календари с избор за цвят, което улеснява проследяването на графика на всеки член от семейството;
- Функции за управление на списъци за пазаруване и задачи;
- Възможности за настройка на напомняне и изпращане на имейли с дневния ред на различните членове.

Достъпност: Приложението е налично само за мобилни устройства (iOS и Android) и не предлага уеб версия, което може да ограничи достъпа до потребители, които искат да използват компютър. Въпреки че има уебсайт, той служи единствено за поддръжка и наръчник. Тази липса на действаща десктоп платформа може да отблъсне клиенти, които търсят по-гъвкави опции за достъп.



Фиг.2: Уебсайта на приложението Cozy Family Organizer

Предимства:

- Потребителски интерфейс: Интуитивен и лесен за използване.
- Безплатна версия: Предлага основни функции без начални инвестиции, което го прави достъпен за широка аудитория.

Недостатъци:

- Има наличен сайт - наръчник за системата, но няма налична Уеб версия, както споменах по-горе.
- Реклами: В безплатната версия на приложението се срещат реклами, които могат да бъдат натрапчиви.

Cozi Family Organizer предлага ефективни решения за ежедневните предизвикателства на семейния живот, като улеснява планирането на дейности. Това го прави ценен инструмент за семейства, които искат да подобрят своята организация и комуникация. Наличието на мобилно приложение подкрепя мобилността и активния начин на живот на съвременните семейства, но липсата на уеб версия ограничава достъпа от други устройства като компютри и таблети.

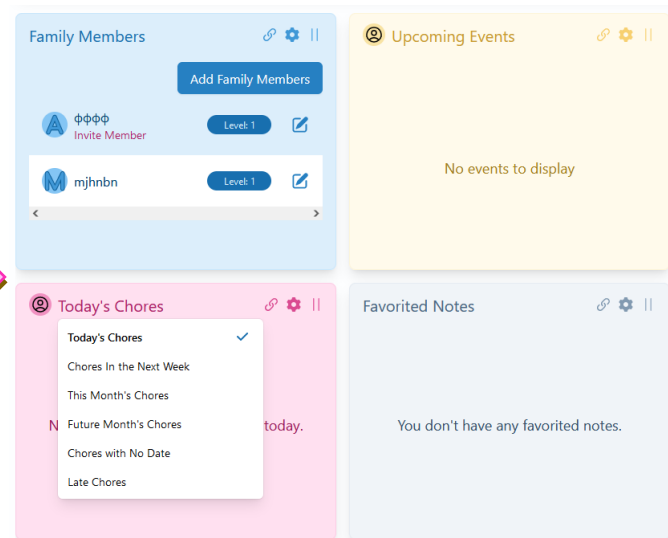
3. Family Tools App

Системата Family Tools App се фокусира върху предоставянето на основни инструменти за управление на семейството, които улесняват ежедневните задачи чрез споделени календари и списъци за домакински задължения. Приложението се стреми да опрости организацията на семейния живот, като предлага лесен за използване интерфейс, който прави планирането и проследяването на задачи лесно и достъпно за всички членове.

Основни функции на приложението включват:

- Опростен споделен календар: Позволява на членовете на семейството лесно да координират и визуализират ангажиментите си.

- Създаване на списъци: Включва възможности за планиране на събития и съставяне на различни списъци, които помагат в домакинското управление.
- Инструменти за управление на задачи, които помагат при организацията и приоритизирне на ежедневните задачи. Разполага с информационно табло, на което се показват всички задачи, събития и бележки за деня, по този начин се създава централизиран общ поглед на всички важни ангажименти.



Фиг.3: Интерфейс на информацията таблица в приложението Family Tools App

Family Tools App е идеален за семейства, които търсят просто и лесно за използване приложение, което ефективно помага в ежедневната организация без излишни функции, които могат да усложнят процеса.

Предимства:

1. Прост и интуитивен интерфейс: Приложението е проектирано така, че да бъде лесно за използване от всички членове на семейството, без значение от техническите им умения.
2. Ефективност при планиране и управление на задачи: Осигурява базови инструменти, които са достатъчни за управление на семейните ангажименти.

Недостатъци:

- Ограничен обхват на функциите: В сравнение с по-сложните приложения като FamilyWall и Cozi, Family Tools App предлага по-малко функции, което може да не отговаря на нуждите на по-заети и по-големи семейства.

- По-малка популярност: По-ниската популярност на приложението може да влияе на доверието и наличността на поддръжка за потребителите.

Family Tools App е особено подходящ за тези, които ценят простотата и директната функционалност. Предоставяйки прости инструменти без добавени сложности, това приложение е идеално за семейства, които искат да поддържат организацията си без ненужни усложнения.

За да систематизирам основните функционалности на системите, анализирани в обзорната част на дипломната работа, представям следната таблица:

Функционалности	FamilyWall	Cozi Family Organizer	Family Tools App
Автентикация	✓	✓	✓
Редакция на профила	✓	✓	✓
Създаване на семейство	✓	✓	✓
Добавяне на членове	✓	✓	✓
Покана на потребители	✓	✓	✓
Семеен календар	✓	✓	✓
Споделяне на календари	✓	✓	✓
Сравняване на множество календари	✗	✗	✗
Семейни роли	✓	✓	✓
Приятелски Семейства	✗	✗	✗
Споделяне между семейства	✗	✗	✗
Проследяване в Рално Време	✓	✗	✗
Списъци със Задачи за изпълнение	✓	✓	✓
Планиране на Хранения	✓	✓	✗
Семеен Дневник	✗	✓	✗
Споделяне на Снимки и Видео	✓	✗	✗
Лични Съобщения	✓	✗	✗
Достъп от Множество платформи	✓	✗	✓
Известия в Рално Време	✓	✓	✓
Налична безплатна версия	✓	✓	✓
Необходим Премиум Абомамент за Пълни Функции	✓	✓	✓

Фиг.4: Сравнителна таблица за наличните функционалности на откритите приложения

Представям обобщен текстов анализ на информацията, представена в таблицата от Фиг.4:

Всяко приложение отговаря на различни нужди за управление на семейни дейности:

- FamilyWall е идеален за семейства, които търсят богато на функции приложение, което може да управлява сложни семейни операции и функции за безопасност, хранителни режими и водене на дневници.
- Cozi Family Organizer е най-подходящ за домакинства, които имат нужда от мобилно приложение за организация и планиране на важни събития.
- Family Tools App е предназначен за тези, които дават приоритет на простотата и директната функционалност без необходимостта от напреднали функции или синхронизация между множество устройства.

Всяка система предлага общи и индивидуални функционалности, адаптирани към различните нужди на семействата. В зависимост от конкретни изисквания, като бюджет, проследяване в реално време, планиране на хранене, лекота на използване или специални функции, всяко семейство може да избере системата, която най-добре отговаря на начина им на живот.

ГЛАВА II

Изисквания към системата

На база изложеното изследване, може да се направи следният извод - към момента на пазара липсва система, която интегрира всички основни функционалности, необходими за управление на семейни активности в едно цялостно софтуерно приложение.

За да отговори адекватно на тези семейни нужди, към системата за управление на семейни активности, разработена в тази дипломна работа, са поставени редица специфични изисквания, които да гарантират ефективността ѝ в ежедневието на потребителите и да подобри организацията на техните дейности. Тези изисквания включват:

Регистрация и автентикация:

Потребителите трябва да имат възможност да създават свои профили чрез формуляр за регистрация, който изисква базова информация като име, имейл, потребителско име и парола. След успешна регистрация, новият потребител трябва да има достъп до системата чрез автентикационен механизъм.

Управление на потребителите:

Системата за управление на семейни активности да поддържа възможност за управление от различни типове потребители. Като всеки да има достъп до своя индивидуален профил след успешна автентикация. Потребителите трябва да имат възможност да редактират своите лични данни в профила си, а акаунтите, които вече не се използват, да могат да бъдат деактивирани или архивирани.

Календари:

Системата трябва да предлага интуитивен интерфейс за календари, улесняващ потребителите в създаването, преглеждането, редактирането и изтриването на лични и семейни данни. Всеки календар има възможност да бъде споделен с останалите членове на семейството или с външни лица, като се предоставят регулируеми настройки за достъп.

Също така те трябва да поддържат възможност за добавяне на множество събития, както и да предоставят избор на потребителя между различни видове изгледи (дневен, седмичен, месечен и годишен).

Създаване на събития:

Системата трябва да осигурява лесен и интуитивен начин за създаване на събития в календарите, улеснявайки потребителите в добавянето на необходимите детайли и настройка на напомния. Това включва достъп до прост и разбираем интерфейс, чрез който може да се въведе информация за събитията, включително име, дата, време и описание. Потребителите да имат възможност да посочат дали събитието е целодневно и да настроят неговата периодичност – да се повтаря дневно, седмично, месечно или годишно. Тази функционалност е идеална за планиране на рождени дни, годишнини и редовни срещи, което прави системата изключително полезна за управление на лични и професионални ангажименти.

Задаване на различни права за достъп и йерархия в семейството:

Системата за управление на семейни активности трябва да включва вградена йерархична структура с различни роли за членовете на семейството, като осигурява адекватен контрол и безопасност. Основните роли, които се предлагат, са "Админ" (Admin) и "Член" (Member), като всяка роля има специфични правомощия и достъп до функционалностите на системата.

- **Админ (Admin):** Потребители с тази роля трябва да имат най-високо ниво на достъп. Те могат да управляват профилите на всички членове на семейството, да канят нови членове, да редактират или изтриват съществуващи профили, и да настройват правилата за достъп и видимост на календарите и задачите в системата.
- **Член (Member):** Обикновени потребители в системата, които имат достъп до собствените си профили и календари. Те могат да добавят, редактират и изтриват собствени събития и задачи, както и да виждат публични или споделени с тях календари и списъци. Тези членовете не могат да модифицират профили на други членове или да променят данни, до които нямат достъп.

Гледане и споделяне на календари:

Системата трябва да предоставя възможности за лесно гледане и споделяне на календари цялостно или частично с други членове на семейството, което улеснява координацията и планирането на общи събития. Потребителите трябва да могат да настройват видимостта на своите календари, като избират кои да бъдат публични и кои – лични.

Филтриране на събития:

Филтрирането на събития е ключова функционалност за управление на големи и сложни календари, осигурявайки на потребителите бърз и лесен начин за откриване на необходимата им информация. Системата трябва да предлага разнообразни опции за филтриране, за да улесни търсенето и организацията на събитията.

- **Филтри по име на събитието:** Позволява на потребителите да търсят събития по тяхното конкретно име, което е идеално за бързо намиране на специфични ангажименти.
- **Филтри по тип събитие:** Потребителите могат да филтрират събития според тяхната продължителност – дали са ежедневни или повтарящи се през времето събития. Този филтър е полезен за разграничаване на дълготрайни ангажименти от по-кратки срещи или задачи.
- **Филтри по дата и време:** Този филтър улеснява потребителите да ограничават показването на събития до определени дни, седмици или месеци. Може да се използва за планиране или преглед на предстоящи ангажименти в конкретен времеви период.

Цветова диференциация на събитията:

Цветовата диференциация на събитията в календара е важен елемент, който помага на потребителите да различават ангажиментите си с един поглед. Системата трябва да предоставя възможност на потребителите да избират и прилагат специфични цветове за своите събития, което от своя страна улеснява бързото идентифициране на приоритетната организация. Тази функционалност е особено полезна за:

- **Визуална организация:** Потребителите могат лесно да организират и преглеждат своя календар, като визуално групират събитията по цвят.
- **Персонализация:** Предоставя възможност за избор на любими цветове или цветови схеми, които отговарят на личните предпочитания на потребителя.

Получаване на известия:

Автоматизираните известия са критична функционалност за поддържане на потребителите информирани за предстоящи събития и ангажименти. Системата трябва да включва механизми за изпращане на напомняния чрез различни канали, които да гарантират, че потребителите никога няма да пропуснат важно събитие. Тези известия могат:

- Напомняне за начало на събитие;
- Напомняне за приближаване на рождени дни или годишнини;
- Повторяеми аларми;
- Изпращане на известия чрез SMS, електронна поща или чрез интегрирани системи за известия на десктоп устройства.

Покани:

Системата трябва да включва функционалност за изпращане на покани до потенциални членове на семейството или приятелски семейства. Поканите могат да бъдат изпратени чрез електронна поща или SMS, като предоставят линк за регистрация или присъединяване към съществуващо семейство.

Създаване на Семейство:

Системата трябва да предоставя интуитивен интерфейс за създаване на семейство, който автоматично назначава инициатора на процеса за администратор. Тази роля има пълни правомощия да управлява настройките на семейството, да кани нови членове и да регулира достъпа до общите ресурси. Така се гарантира ефективно и централизирано управление на семейните ангажименти, което улеснява комуникацията и координацията между членовете.

Управление на Списъци със задачи за изпълнение:

Системата да е изградена от гъвкави инструменти за създаване, редактиране, споделяне и изтриване на списъци със задачи. Всяка задача има инициатор и изпълнител. Други елементи на задачата включват име, описание към нея, краен срок за завършване и текущ статус за проследяване на напредъка. Използването на списъци със задачи, улеснява организацията и разпределянето на ангажименти.

Споделяне на задачи и колаборация:

Системата трябва да улеснява споделянето на задачи между членовете на семейството, което от своя страна стимулира колаборацията и повишава ефективността при изпълнението на общи задачи. Потребителите могат да дават достъп до своите списъци със задачи на други членове, което позволява на всички участници да виждат прогреса в реално време, да добавят нови задачи, или да коментират и актуализират съществуващите.

Сигурност и достъпност:

След регистрация потребителят трябва да влезе в системата използвайки механизъм за автентикация. По този начин, системата гарантира високо ниво на сигурност и защита на данните, като използва модерни методи за шифроване и автентикация. Потребителската информация и данни се съхраняват в защитена среда, като се осигурява надежден достъп само за авторизирани потребители. Допълнителните настройки за конфиденциалност позволяват на потребителите да контролират кой може да вижда или редактира тяхната лична информация.

Приятелски семейства и споделяне между тях:

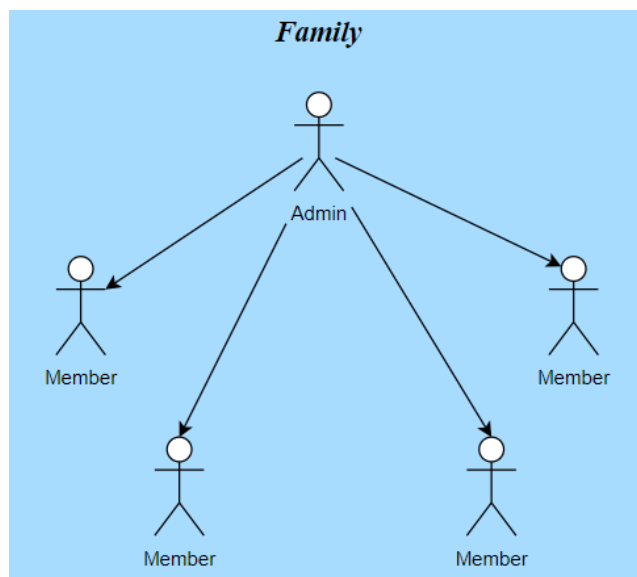
Системата трябва да включва опция за създаване на "приятелски семейства", което позволява на различни семейства да си сътрудничат и споделят информация. Това им позволява да планират съвместни събития, почивки и ангажменти, което допринася за по-активно общуване. Споделянето може да бъде контролирано чрез настройки за поверителност, които определят каква информация може да бъде видяна и редактирана от другите семейства, гарантирайки, че всяко от тях запазва контрол върху своите лични данни.

Сравнение на множество календари:

Функцията за сравнение на множество календари позволява на потребителите да преглеждат и анализират наличните събития от различни календари едновременно. Това осигурява цялостен поглед на всички ангажменти, улеснява идентифицирането на конфликти в графици и помага за намиране на свободно време, което е от съществено значение за координацията на общи планове. Тази функция е изключително полезна за големи семейства или групи, които трябва да си синхронизират програмите за съвместни събития или екскурзии.

Семейна йерархична структура:

Софтуерната система за управление на семейни активности трябва да поддържа йерархична структура на членовете в семейството, която определя различните роли и отговорности. Тя е показана в следната диаграма:



Фиг.5: Примерна структура на членовете в една семейна организация

Всеки потребител трябва да има точно определена роля, като в контекста на семейството, възможните роли са Admin и Member. Потребителите с роля на Администратор имат пълните правомощия в управлението на цялото семейство. Те включват:

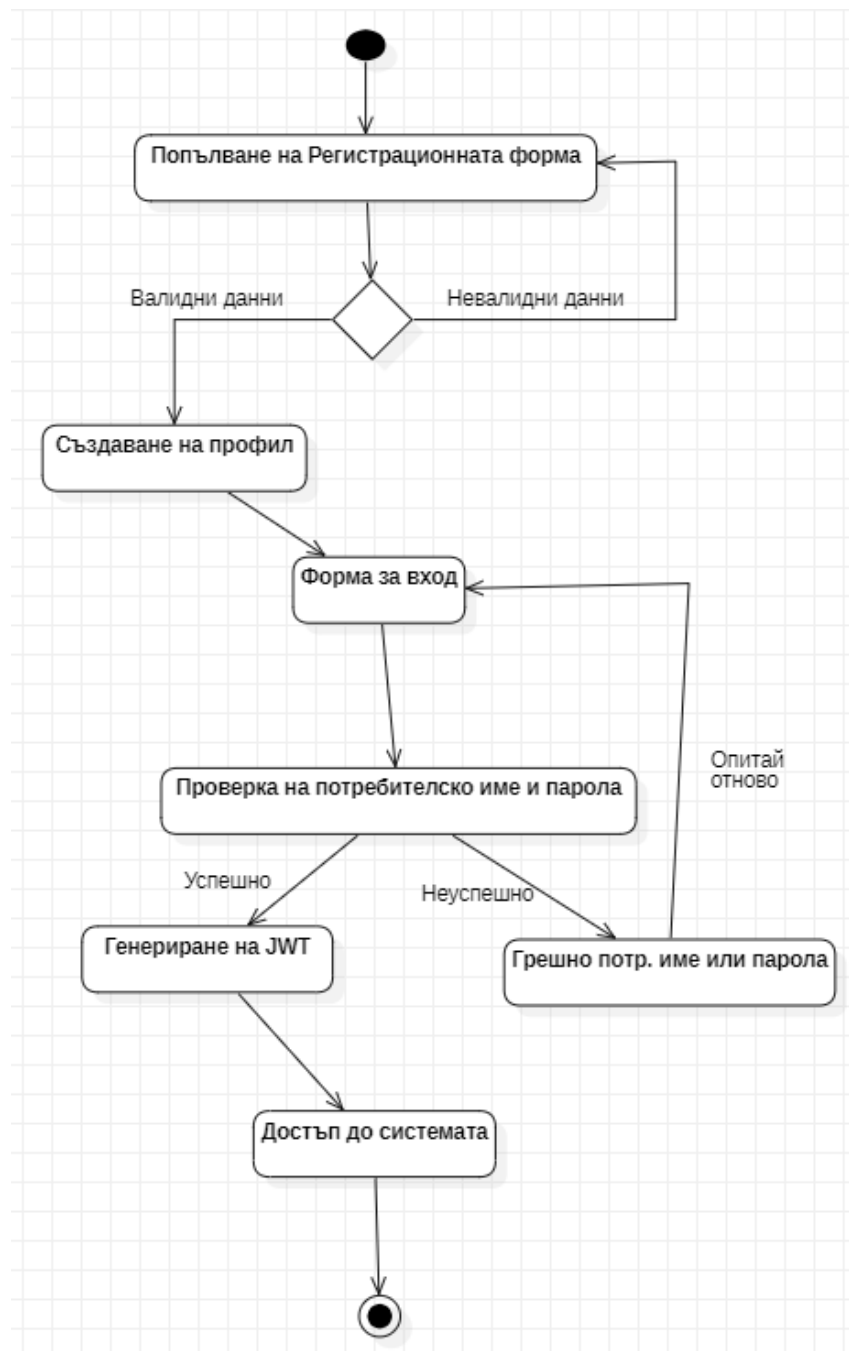
- добавяне или премахване на членове;
- задава или променя правата и достъпа до определени ресурси и календари;
- създаване на задачи и назначаване за изпълнение им;
- управление на семейните календари и общи събития.

Другите потребители с роля на Членове имат следните правомощия:

- добавяне и управление на личните си събития и задачи;
- участие в планирането на дейности;
- управление на зададените им задачи за изпълнение.

Работен процес по регистрация и автентикация:

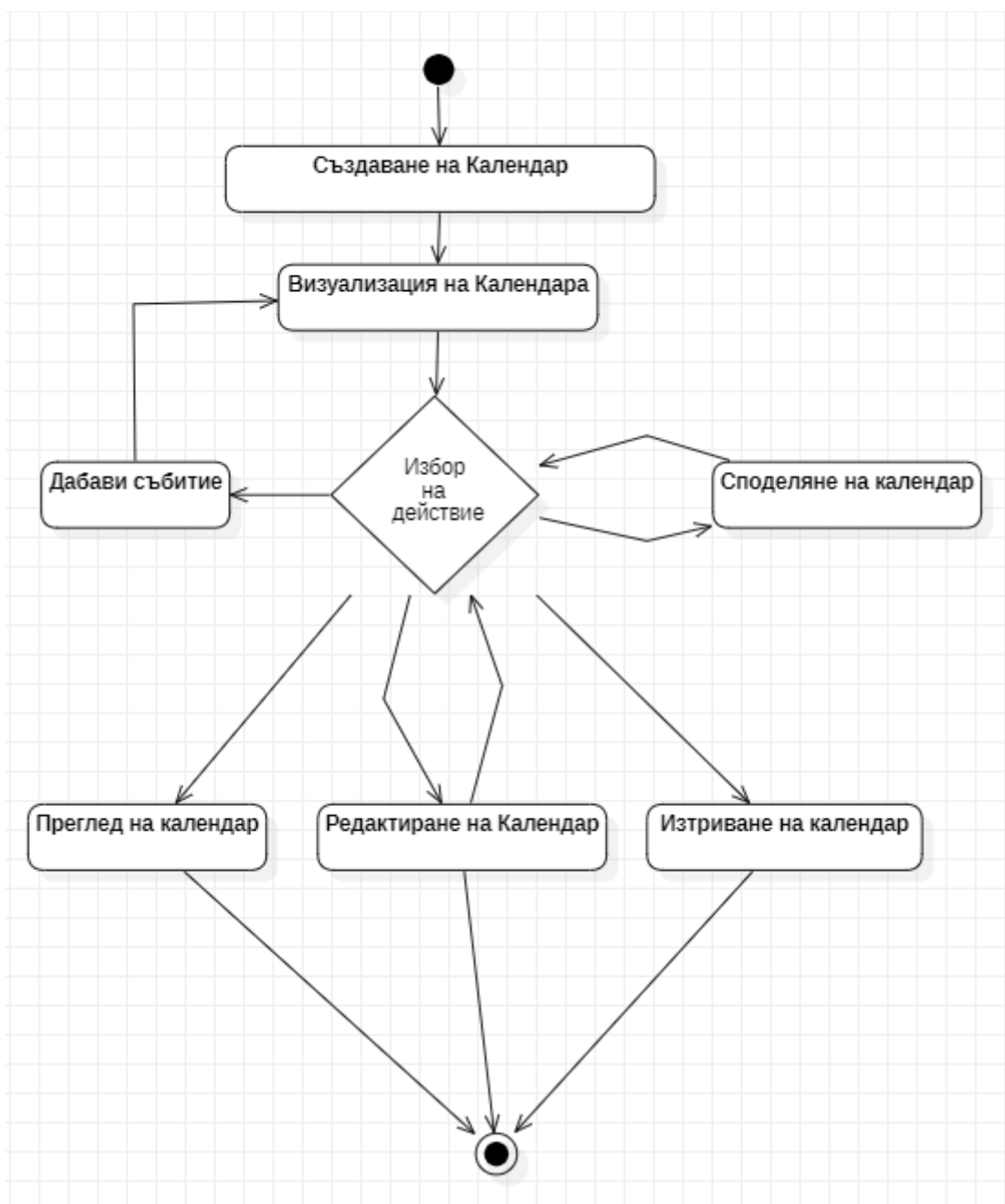
Системата предлага метод за регистрация и автентикация, което улеснява потребителите да създават профили и да влизат в системата чрез безопасен и сигурен начин на удостоверяване. Това се осъществява с помощта на JWT (JSON Web Token), който функционира като начин за потвърждение на достъпа.



Фиг.6: Activity диаграма – Регистрация и Автентикация на потребител

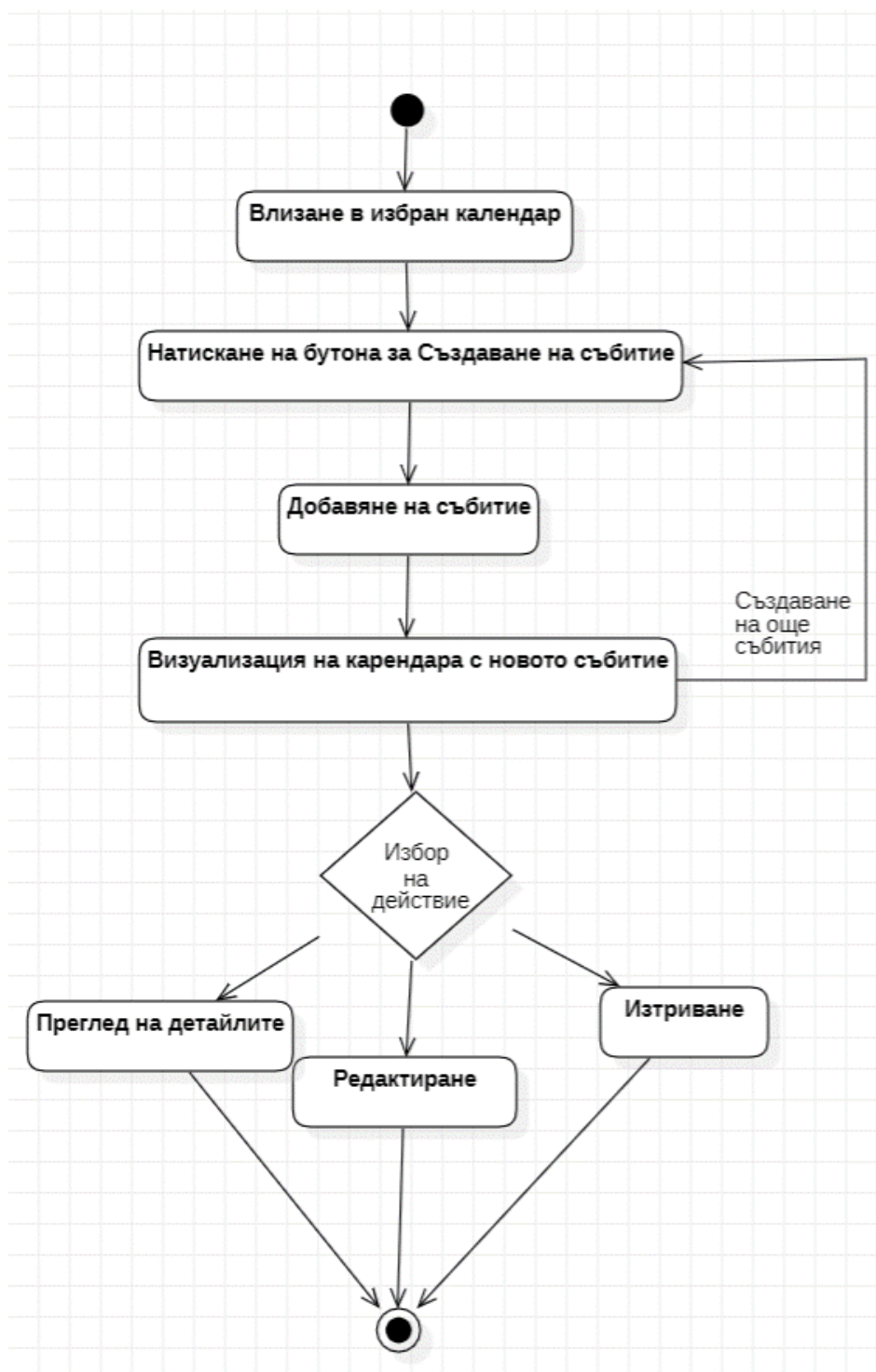
Работен процес по създаване и управление на календари:

Софтуерът трябва да предоставя средства за създаване и контрол на календари. Това включва създаване, редактиране, споделяне, изтриване на избран календар и добавяне на събития към него.



Фиг. 7: Activity диаграма за Управление на Календари

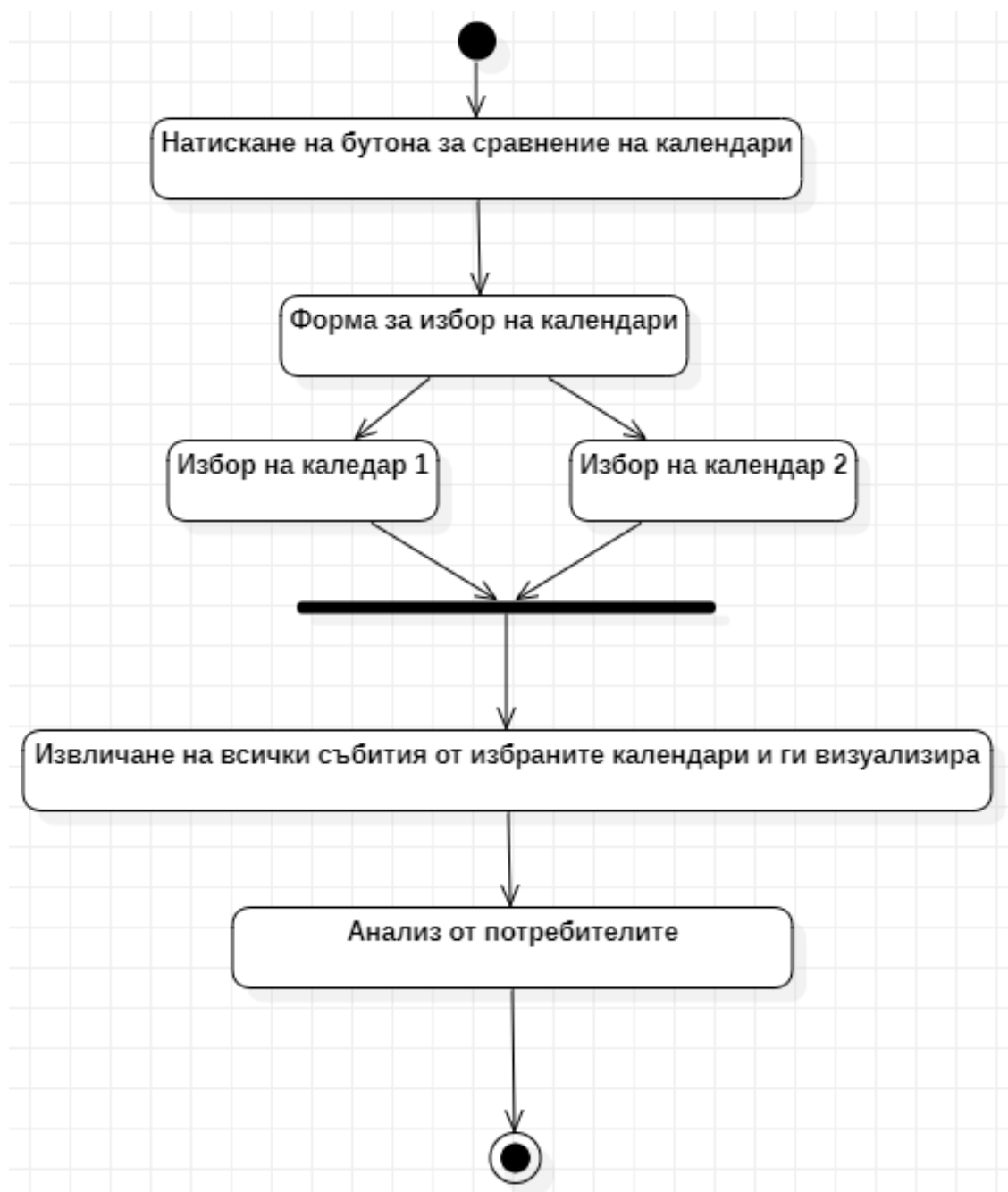
Работен процес по създаване и управление на събития в календара:



Фиг.8: Activity диаграма – Управление на Събития

Работен процес по сравнение на множество календари:

Софтуерната система трябва да може да съпоставя събития между два или повече календара. Тази възможност позволява детайлен анализ на времето на различни потребители. Това улеснява процеса по откриване на свободно време и планиране на бъдещи събития.



Фиг.9: Activity диаграма – Сравнение на Календари

ГЛАВА III

Използвани технологии

За изготвянето на софтуерната платформа и улесненото съхранение на данни, въведени от потребителите, са използвани следните технологии:

1. MS SQL Server



За управление и съхранение на данни в софтуерния продукт е избран MS SQL Server, поради неговата лекота при интегриране с ASP.NET приложения и съвместимост с Windows операционна система.

MS SQL Server е релационна база данни, разработена от Microsoft. Тя е създадена като конкурент на Oracle и MySQL базите от данни и се утвърждава като водеща на пазара за корпоративни приложения в Windows среда. Като релационна база, MS SQL Server поддържа ANSI SQL – стандартния SQL език и включва и T-SQL като специфична SQL имплементация. За удобство на потребителите, има включено SQL Server Management Studio, което улеснява достъпа и управлението на базите от данни. Този инструмент предоставя както скриптов редактори, така и графичен интерфейс за взаимодействие с базата.

Някои от ключовите функционалности на MS SQL Server включват^[1]:

- Управление на бази данни - **Database management**
- Поддръжка и управление на **XML** данни;
- **SQL Server Analysis Services (SSAS)** за анализ на данни;
- **SQL Server Reporting Services (SSRS)** за създаване на отчети;
- **Dynamic management views (DMVs)** за динамични изгледи към управлението на данни;
- **Database mirroring** – за възстановяване на базата при срыв. Това се дължи на двете инстанции на SQL сървъра, които могат да бъдат на една и съща машина или на два различни. Благодарение на тази функционалност се увеличава сигурността.
- **Филтриране и търсене** в базите данни;

MS SQL Server се предлага в няколко версии, вариращи по функционалност и ценови параметри, което позволява на всеки потребител да избере най-подходящия пакет според своите нужди:

- **Enterprise** – Разработен е за големи корпорации със сложни и разнообразни потребности за управлението на данните, включително системи за data warehousing(за отчети и анализ), както и уеб-базирани бази данни. Премиум версията им предлага обширни възможности за използване на големи центрове с изключително бърза производителност, неограничена виртуализация и пълноценни бизнес функции, които гарантират високо ниво на обслужване за критично важни работни натоварвания и достъп на потребителите до важни бизнес анализи. Пакетът включва всички възможности на SQL сървъра и е с най-висока цена спрямо другите версии.
- **Standard** – Подходящ е за малки до средни предприятия, включващ функции за data warehousing и електронна търговия. Предлага основни функции за управление на данни и инструменти за бизнес интелигентност. Поддържа разработката на приложения както на местно ниво, така и в облак, което улеснява управлението на базите данни с минимални IT ресурси.
- **Web** – Този вариант предлага изгодни икономични решения за уеб хостинг и уеб VAPs, които се разширяват и управляват уеб проекти от всякакъв размер. Тази опция е идеална за хора, които използват IaaS на Azure, и търсят достъпност и лекота.
- **Workgroup** – Идеален за малки бизнес организации, този пакет е чудесно средство за управлението на бази данни на малки уеб сървъри или офис среди.
- **Developer** – Позволява на разработчиците да изграждат различни приложения въз основа на SQL Server. Включени са всички функции на версията Enterprise, но е

лицензирана за използване като развиваща и тестова система, а не като продукционен сървър. SQL Server Developer е перфектен за разработчици и тестери на приложения.

- **Express** – Това е базовият безплатен пакет, идеален за учене и създаване на десктоп или малки сървърни приложения. Предлага на потребителите ограничени функционалности, но достатъчни за лична употреба. Пакетът е лимитиран откъм размера на базите данни и броя потребители, но може да се използва като алтернативен вариант на Access базата от данни. Той е предпочитан избор за независими софтуерни доставчици и разработчици, които създават клиентски приложения. **SQL Server Express LocalDB** е по-лека версия на Express, която разполага с всички програмни функции, работи в потребителски режим и предлага бързо инсталиране без нужда от допълнителна конфигурация.

2. *ASP.NET Web API u Entity Framework.Core*



За разработката на този софтуерен продукт е избрана платформата ASP.NET Web API и технологията Entity Framework.Core, който се използва за достъп и управление на базата данни.

ASP.NET Web API и Entity Framework.Core са съществени компоненти за модерната разработка на софтуер, използващи технологията на Microsoft .NET.

ASP.NET Web API е рамка за изграждане на уеб API върху .NET Core платформата, която поддържа създаването на HTTP услуги, които връщат различни формати данни (JSON, XML) и са оптимизирани за ефективност и разширяемост.

Предимства от създаването на API в ASP.NET Core^[1]:

С помощта на ASP.NET можем да създаваме уеб страници и услуги, използвайки една и съща рамка и шаблон. Този подход има следните предимства:

- **Проста сериализация:** ASP.NET е проектиран за модерни уеб изживявания. Крайните точки автоматично сериализират класовете в правилно форматиран JSON. Не е необходима специална конфигурация, като има възможност от персонализирана сериализацията за крайни точки, които имат уникални изисквания.
- **Потвърждаване и упълномощаване:** За допълнителна сигурност, крайните точки на API имат вградена поддръжка за стандартни JSON Web Tokens (JWT). Даването на правомощия, базирано на правила, осигурява гъвкавостта да се дефинират правила за контрол на достъпа.
- **Маршрутизиране:** ASP.NET позволява дефинирането на маршрути, вградени в кода, като се използват атрибути. Данните от пътя на заявката (request path), низа на заявката (query string) и тялото на заявката (request body) автоматично се свързват с параметрите на метода.
- **HTTPS по подразбиране:** HTTPS е важна част от съвременните, професионални уеб API. Той разчита на криптиране от край до край, за да осигури поверителност и да гарантира, че API повикванията няма да бъдат прихванати или променени между клиент и сървър.
- **Инжектиране на зависимости:** Вградената поддръжка за инжектиране на зависимости подобрява модулността и тестваемостта на приложенията.


Entity Framework Core (EF Core) е отворена ORM рамка (*фреймуърк*) от Microsoft, която позволява на разработчиците да работят с база данни, използвайки .NET обекти. Тя елиминира необходимостта от писане на повечето код за достъп до данни, който разработчиците обикновено трябва да пишат.

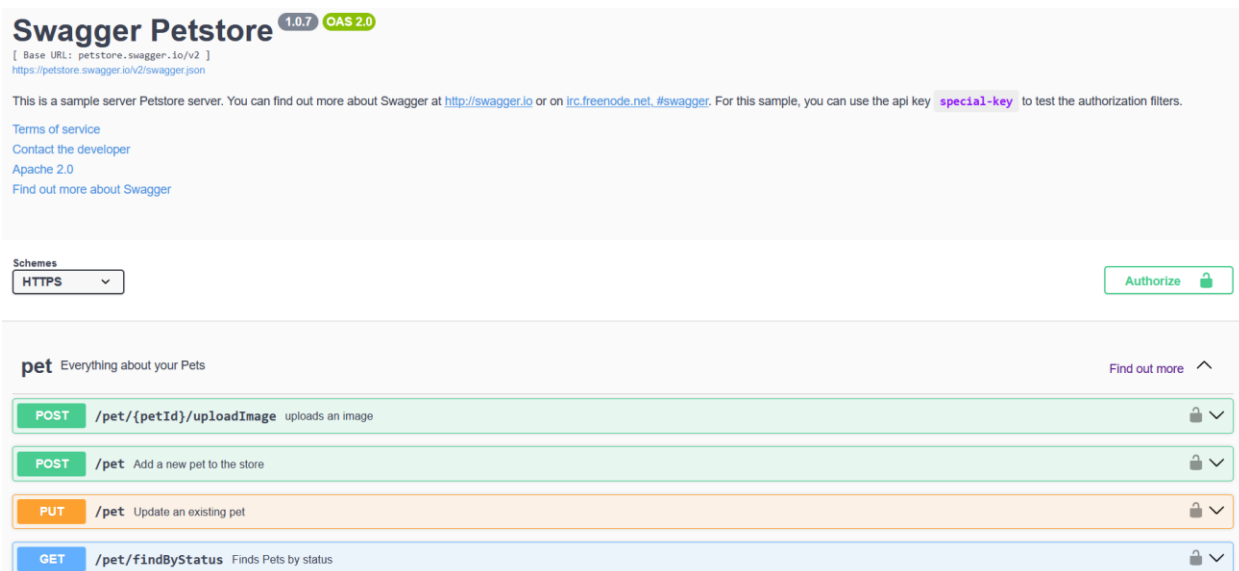
- **Неутралност към базата данни:** EF Core поддържа различни бази данни чрез модел на доставчици. Доставчици са налични за SQL Server, SQLite, PostgreSQL, MySQL и други.
- **Поддръжка на миграции:** EF Core предлага мощна система за управление на промени в схемата на базата данни.

- **Производителност:** Значително е оптимизиран в .NET Core, за да има по-добра производителност.
- **Упростени CRUD операции:** EF Core позволява на разработчиците да извършват CRUD операции без необходимостта от явно писане на SQL заявки, използвайки LINQ за взаимодействие с базата данни.
- **Проследяване на промени:** Автоматично се проследяват промените, направени в инстанциите, което от своя страна - намалява количеството на шаблония код. В практически приложения, комбинирането на ASP.NET Web API с Entity Framework Core позволява на разработчиците бързо да създават сигурни уеб приложения и услуги. Програмистите имат шанса да се съсредоточат повече върху бизнес логиката и по-малко върху достъпа до данни и поддръжката на API. Тази комбинация е особено ефективна за предприятия, където устойчивостта и производителността са от съществено значение.

Заедно, тези инструменти предоставят всеобхватна екосистема за разработка на сложни приложения, като се предлага оптимизиран работен процес от управлението на данни до разработката и разгръщането на API.

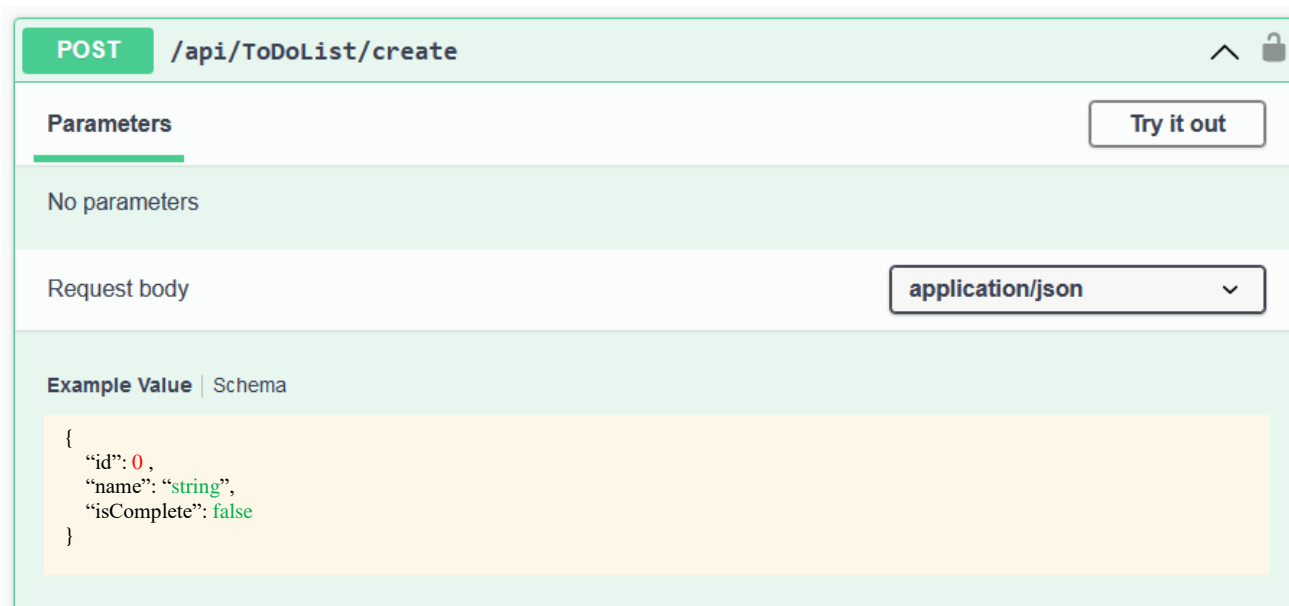
За тестване и проверка на бизнес логиката е използван **Swagger UI**^[8], предоставен от Swagger/ OpenAPI.

Swagger UI  предлага уеб-базиран потребителски интерфейс, който предоставя информация за услугата, използвайки генерираната OpenAPI спецификация. Както Swashbuckle, така и NSwag имат вградена версия на потребителския интерфейс на Swagger, така че да може да бъде хостван директно в ASP.NET Core приложение чрез извикване за регистрация на междинен софтуер. Уеб интерфейсът изглежда така^[9]:



Фиг.10: Интерфейс на Swagger UI

Всеки публичен метод за действие в контролерите може да бъде тестван от потребителския интерфейс. Първо се избира име на метода, който тестваме, като по този начин се разширява секцията. След това се добавят всички необходими параметри, за да се изпълни заявката и се натиска бутона Изпробвайте (Try it out!) ^[10].



Фиг.11: Интерфейс за подаване на заявка

3. *React u Material UI components*



Потребителският интерфейс на софтуерния продукт е изработен с технологията React в комбинация с компонентите на Material UI.

React е гъвкава библиотека за създаване на уеб и интуитивни потребителски интерфейси, позволяваща на разработчиците да създават издръжливи интерфейси от модулни компоненти като Thumbnail, LikeButton и Video. Компонентите в React могат да бъдат разработвани независимо и интегрирани безпроблемно, което прави библиотеката идеална за колаборативни и мащабируеми проекти^[2].

Компонентите на React са предимно JavaScript функции, които могат да включват JSX, разширение на синтаксиса, което ни позволява писането на HTML в рамките на JavaScript. Благодарение на този подход могат да се пишат разметки (markup) и код заедно в един компонент, което улеснява процеса на разработка чрез поддържането на бизнес свързаната логика и рендерирането им заедно^[3].

Дизайнът на React поддържа интерактивен потребителски интерфейс, който актуализира дисплея, когато потребителите взаимодействат с елементите на страницата. Например, той може да реагира на потребителски входове, като актуализира съдържанието динамично без необходимостта от презареждане на цялата страница. Това прави React-а особено полезен за добавяне на интерактивни елементи към съществуващи уеб страници или за разработване на сложни, отзивчиви приложения.

Едно от силните качества на React е неговата способност за лесна интеграция. Разработчиците могат да започнат да използват React в малка част от техния проект, като единичен компонент и постепенно да разширяват. Тази гъвкавост води до експериментиране на React в малки проекти или интегриране в големи мащабни приложения.

Този подход не само подобрява способността за създаване на богат потребителски интерфейс, но също така гарантира, че разработката им е ефективна и поддържана.

Комбинацията на React с Materail UI позволява използването на предварително дефинирани компоненти за бързо и ефективно създаване на естетически приятен и функционален интерфейс. Material UI предоставя голям набор от компоненти, които са в съответствие с принципите на дизайн на материали и са лесни за интегриране с логиката, разработена с React.

Material UI представлява библиотека с отворен код за компоненти на React, която реализира Material Design на Google. Тя е изчерпателна и може да се използва в производство веднага след инсталиране. Основната ѝ цел е да предостави основни компоненти на разработчиците, за да създават отлични потребителски интерфейси, като използват насоките на Material Design като референция^[4].

Предимства на Material UI:

- **Бързо доставяне:** Над 2,500 разработчици допринасят за тази компоненти с отворен код. По този начин се позволява на програмистите да са фокусирани върху логиката на самото приложение, вместо да изработват от начало.
- **Красиви компоненти по подразбиране:** Самата имплементация на Material Design е прецизна, гарантирайки, че всеки компонент отговаря на високи стандарти за форма и функционалност.
- **Възможност за персонализиране:** Библиотеката включва обширен набор от интуитивни функции за персонализация.
- **Междуетипно сътрудничество:** Интуитивният опит за разработчици на Material UI намалява бариерите за вход за разработчици и дизайнери, което улеснява сътрудничеството. Интуитивното изживяване на разработчиците на Material UI дава възможност на екипите да си сътрудничат по-ефективно, независимо дали са бекенд разработчици или технически дизайнери. Дизайнерските комплекти рационализират работният процес и повишават последователността между дизайнери и разработчици.
- **Доверие от хиляди организации:** Material UI има най-голямата UI общност в екосистемата на React. Той е почти толкова стар, колкото и самият React – историята му датира от 2014 г.

Компонентите от Material UI се доставят със съответни стилове по подразбиране и са оптимизирани за работа с **Emotion** (или styled-components). Emotion е библиотека, предназначена за писане на css стилове с JavaScript. Тя осигурява мощна и предвидима стилова композиция в допълнение към страхотно изживяване за разработчици с функции като изходни карти, етикети и помощни програми за тестване. Поддържат се както низови, така и обектни стилове^[15].

Material UI предоставя разнообразие от компоненти, които помагат в разработката на интерфейси заедно с насоките на Material Design. Основните компоненти включват^[15]:

1. **Buttons:** Бутони за различни действия в приложението.
2. **Text fields:** Текстови полета за въвеждане на информация от потребителя.
3. **Icons:** Икони, които могат да бъдат използвани за подобрене на визуалната комуникация.
4. **Cards:** Карти, които организират информацията в компактен и визуално привлекателен формат.
5. **Toolbars:** Инструментални ленти за подреждане.
6. **Menus:** Менюта за навигация и избор от опции.
7. **Dialogs:** Диалогови прозорци за интерактивни предупреждения, потвърждения или допълнителна информация.

Тези компоненти могат да бъдат настройвани и модифицирани, за да отговарят на конкретните изисквания на вашето приложение, като предлагат гъвкавост и ефективност в разработката на потребителския интерфейс.

4. JWT Token и Axios



В рамките на разработката на софтуерния продукт за управлението на потребителските сесии и сигурността, се използва JWT (JSON Web Tokens). Този метод позволява сигурна трансмисия на информация между клиента и сървъра като JSON обекти. JWT се използва широко за автентикация и обмен на информация, като осигурява механизъм за потвърждение на твърденията, които се предоставят между две страни.

За взаимодействие с REST API, разработката на клиентската част включва използването на Axios - популярна JavaScript библиотека за извършване на HTTP заявки. Axios предоставя лесен за използване API и позволява глобално конфигуриране на заявките, което е идеално за работа с автентикации чрез JWT. Благодарение на Axios, приложението ефективно управлява HTTP заявки към сървъра, обработва грешки и осигурява сигурна аутентикация и обработка на сесии.

JWT или JSON Web Token е отворен стандарт, използван за сигурно споделяне на информация между две страни — клиент и сървър. Обикновено JWT съдържа кодиран JSON, който включва набор от твърдения (claims) и подпис. Често се използва в контекста на други механизми за удостоверяване като OAuth, OpenID за споделяне на информация, свързана с потребителя.

Автентикацията чрез JWT е популярен и надежден сценарий. След като веднъж потребителят е влязъл в системата, всяка следваща заявка ще включва JWT, който ще позволи на потребителя да достъпва маршрути, услуги и ресурси, които са разрешени с този токен. Използва се клиент-базирана сесия без състояние, което означава, че сървърът не е зависим от базата данни за запазване на информация за сесията.

JWT могат да бъдат криптирани, но обикновено са кодирани или подписани. Целта на подписания JWT не е да скрие данните, а да гарантира тяхната автентичност. Подписаните токени могат да проверят цялостта на съдържащите се в тях твърдения, докато криптираните токени скриват тези твърдения от други страни. Когато токените са подписани с помощта на двойка публичен/частен ключ, подписът също така удостоверява, че само страната, която държи частния ключ, е тази, която го е подписала^[6].

Структурата на JWT е разделена на три части, разделени с точки (.):

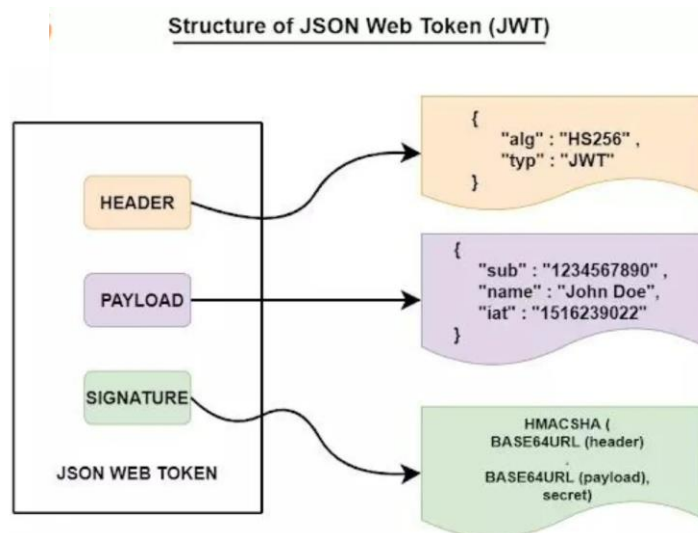
- Заглавие

- Получен резултат



xxxxxx.yyyyyy.zzzzzz

- Подпис



Фиг.11: Структура на JWT^[7]

Заглавна част (Header) се състои от две части:

- Използваният алгоритъм за подписване;
- Типът на токена, който в този случай е предимно „JWT“.

Полученият резултат (Payload) обикновено съдържа твърденията (атрибути на потребителя) и допълнителни данни. Съществуват 3 типа твърдения:

- Регистрирани твърдения - Това са предварително определени твърдения, които не са задължителни, но са препоръчителни, за да се предостави набор от полезни, съвместими твърдения. Някои от тях са: iss (издател), exp (срок на изтичане), sub (тема) или aud (аудитория).
- Публични твърдения - Те могат да бъдат дефинирани по воля от тези, които използват JWT. За да се предотвратят проблеми, трябва да се зададат в

регистъра на IANA JSON Web Token Registry или да бъдат дефинирани като URI с уникално идентификационно пространство.

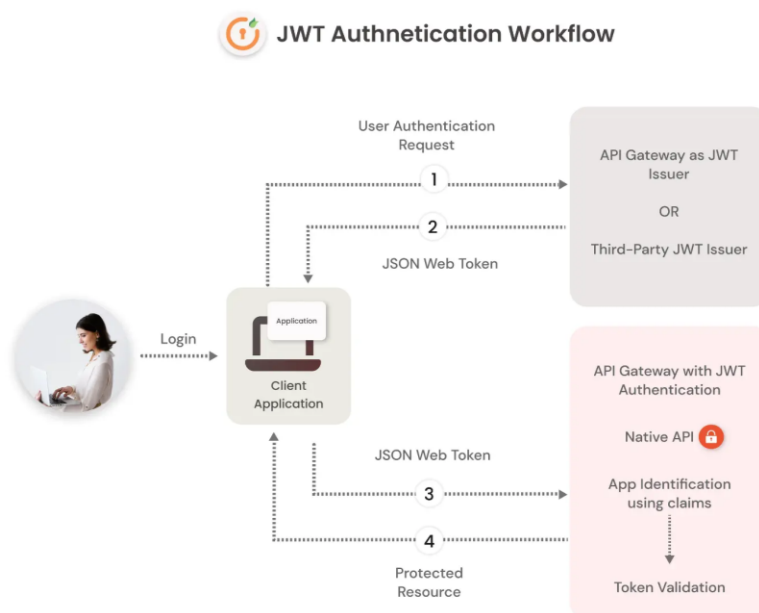
- Частни твърдения - Персонализирани твърдения, създадени за споделяне на информация между участниците, които са се съгласили да ги използват, и не са нито регистрирани, нито публични твърдения.

Подпис: След кодирането на заглавието и получен резултат в Base64, се комбинират с тайна (ключ – известен само на създателя) и алгоритъм за криптиране, споменат в заглавието на токена, като например HS256. Този алгоритъм използва тайната за генериране на уникален подпис.

Подписът се използва за потвърждаване, че токенът JWT не е бил модифициран или променян по време на прехвърлянето. Той може също така да бъде използван за валидиране на изпращача.

Когато става въпрос за автентикация на API и удостоверяване от сървър към сървър, JSON уеб токенът (JWT) е особено полезна технология. В контекста на Single Sign-On (SSO), това означава, че доставчикът на услуги може да получава надеждна информация от сървъра за автентикация.

Чрез споделяне на таен ключ с доставчика на идентичност, доставчикът на услуги може да хешира част от получения токен и да го сравни с подписа на токена. Ако този резултат съвпада с подписа, SP знае, че предоставената информация идва от друга субект, който притежава ключа.



Фиг.12: Диаграма на работния процес за автентикация на JWT

Процесът на автентикация протича по следния начин:

1. Потребителят влиза с потребителско име и парола.
2. Сървърът за автентикация проверява удостоверенията и издава JWT, подписан с частен ключ.
3. Отгук нататък клиентът използва JWT за достъп до защитени ресурси, като прехвърля JWT в HTTP Authorization заглавието.
4. След това сървърът на ресурсите проверява автентичността на токена, използвайки публичен ключ.

Доставчикът на идентичност генерира JWT, удостоверяващ идентичността на потребителя, а сървърът на ресурсите декодира и проверява автентичността на токена, използвайки публичния ключ.

Тъй като токените се използват за удостоверяване и автентикация в бъдещи заявки и API извиквания, трябва да се полагат големи грижи за предотвратяване на критични проблеми. Тези токени не трябва да се съхраняват на общодостъпни места като локалното съхранение на брауъра или бисквитките.

ГЛАВА IV

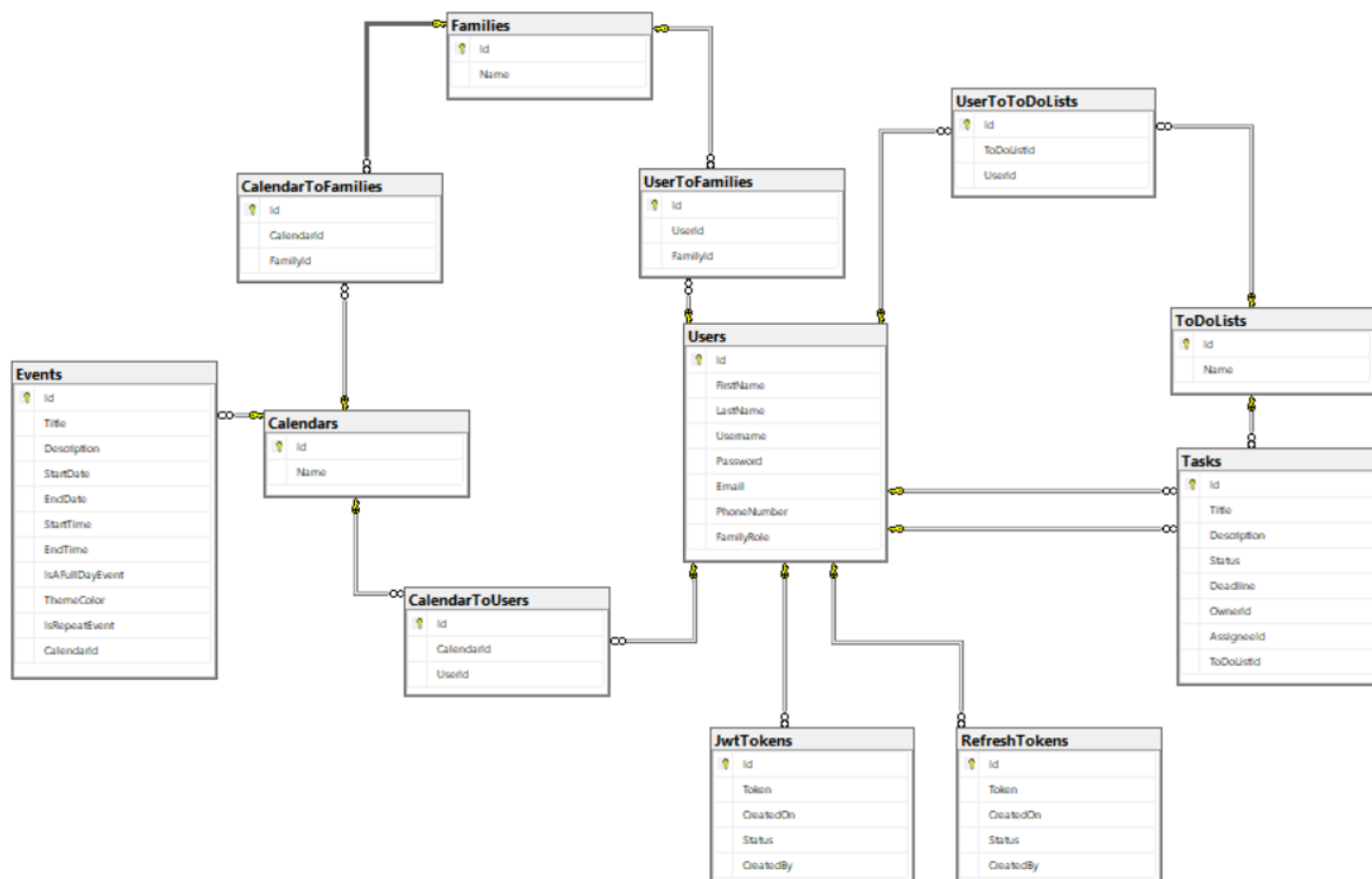
Технологичен дизайн

1. Структура на базата данни

Изградена е логическа структура на източника на данни, която да обслужва системата, така че да отговаря на дефинираните изисквания в **Глава 1**. В системата за управление на семейните активности основният обект е потребителят, който участва в семейни групи и взаимодействия с различни обекти, които репрезентират елементи от ежедневието като календари, събития, задачи за изпълнение.

Изгражда се пълна структура и йерархия на семейството и членове му. Благодарение на интуитивно изградената структура на базата от данни, потребителите могат лесно да управляват своите ангажименти и семейни дейности, а програмистите могат да поддържат без проблем цялата система, като поддръжката се изразява с модифициране, оптимизиране и разширяване на функционалностите.

Структурата на базата е максимално опростена, което улеснява начина на взаимодействие между обектите и разбирането на цялата концепция. Фиг.13 представя диаграма на основните класове в софтуерната система, техните атрибути и връзки.

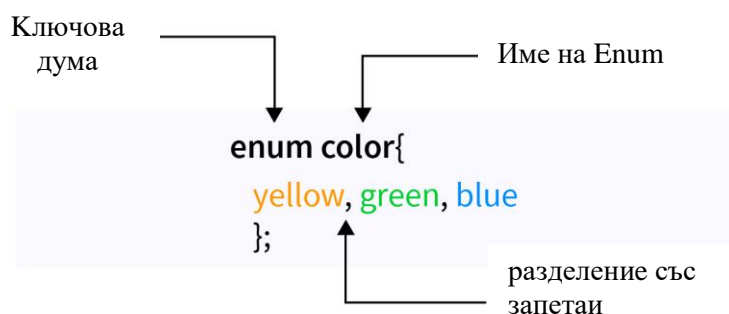


Фиг.13: Клас диаграма на реализираната база данни в MS SQL Server

Както споменах по-горе потребителят е основният фокус, около който се въртят всички дейности. Дори семейството, чиято организация е проблемът ни за решение, се състои от потребители. Затова всеки потребител има следните характеристики:

- **Id** – уникален идентификатор, използван за различаване на всеки потребител;
- **FirstName** – собствено име на потребителя;
- **LastName** – фамилно име на потребителя;
- **Username** – потребителско име, с което потребителя ще влиза в системата;
- **Password** – паролата, с която потребителя ще влиза в системата;
- **Email** – имейл адрес на потребителя, към който ще се изпращат известия и покани;
- **PhoneNumber** – телефонен на потребителя, на които ще се изпращат също известия;

- **Family Role** – ролята на всеки потребител в семейството. Тя може да бъде Admin или Member. Според тази роля се определят правата на достъп на потребителя. Тази характеристика се съхранява като „Enum“ тип стойност, който специфицира група от именувани константи. Това прави кода по-четим и оптимизиран^[13]. На Фиг.14 е представен основния синтаксис на Enum :



Фиг.14 : Основен синтаксис на Enum ^[14]

Както се вижда, има проста дефиницията и основен тип `int` , който започва от 0.

- **Taks: ICollection<Task>** – Всеки потребител има възможност да създава и да бъде назначен за изпълнението на задача. Затова има колекция от таскове(задачи);
- **Clendars: ICollection<Calendar>** – Всеки потребител може да създава календари и ще ги съхраняваме по-този начин;
- **Families: ICollection<Family>** – Потребителят може да участва в много семейства.

Следващият важен обект е Семейството, което има :

- **Id** – уникален идентификатор, използван за различаване на всички семейства;
- **Name** – име на семейството;
- **Clendars: ICollection<Calendar>** – в това поле се съхраняват всички споделени календари с цяло семейство.

Понеже календарите са неотменна част от живота, системата съхранява множество създадени календари от потребителите, независимо дали са лични или споделени. Всеки от тях притежава:

- **Id** – уникален идентификатор, използван за различаване на календарите;
- **Name** – име на календара;
- **Users: ICollection<User>** – това поле съхранява всички потребители, с които е споделен съответният календар;
- **Families: ICollection<Family>** – в това поле се събират всички семейства, с които е споделен календара;
- **Events: ICollection<Event>** – тази колекция включва всички събития от календара.

Какво е календар без събития? След създаването на своя календар, потребителите могат да добавят събития в него. Те съдържат:

- **Id** – уникален идентификатор, използван за различаване на събитията;
- **Title** – заглавие на събитието;
- **Description** – подробно описание на събитието, което включва всички необходими детайли като местоположение, участници, цели;
- **StartDate** – дата, на която събитието започва ;
- **EndDate** – дата, на която приключва събитието;
- **StartTime** – точен час на започване на събитието;
- **EndTime** – точен час на края на събитието;
- **IsAllDayEvent** – булев атрибут, който маркира събитието за целодневно;
- **ThemeColor** – цвят на събитието. Това се използва за визуалното разграничаване на събитията в календара.

Този атрибут е от тип *Enum* и възможните избори са:

- Red (0) – Червено
- Blue – Синьо
- Green – Зелено
- Yellow – Жълто
- Orange – Оранжево
- Purple - Лилаво
- Pink - Розово
- Gray - Сиво
- Black - Черно

- **IsRepeatEvent** – Потребителя може да избере типа на повторение. Този атрибут предоставя възможността да се зададат различни честоти на повтаряне, което от своя страна е изключително полезно за събития като рождени дни, годишнини или чести семейни срещи. Възможните стойности са:
 - None (0) – не се повтаря;
 - Daily – повтаря се ежедневно;
 - Weekly – повтаря се веднъж седмично;
 - Monthly – повтаря се веднъж месечно;
 - Yearly – повтаря се всяка година.
- **CalendarId** – Идентификатор на календара, към който принадлежи събитието.

Списъкът със задачи играе голямо влияние при организацията и изпълнението на ежедневните задължения. Потребителите имат шанса да създават списъци, с които да си помагат при планирането на важни събития или домакинската работа. Те са значими за ефективното управление на времето и ресурсите. Списъци имат:

- **Id** – уникален идентификатор, използван за различаване на списъка;
- **Name** – име на списъка;
- **Tasks: ICollection<Task>** – колекция от задачи, която позволява всяка задача да бъде лесно идентифицирана и свързана със съответния ѝ списък. Елемента е от съществено значение за управлението на множество задачи, като ни гарантира, че всяка от тях може да бъде намерена, актуализирана или отбелязвана като завършена в рамките на своя списък.

Всяка зададена задача в списъка се дефинира със специфични характеристики, които помагат при планирането и изпълнението на важни действия. Характеристиките на задачата в платформата включват:

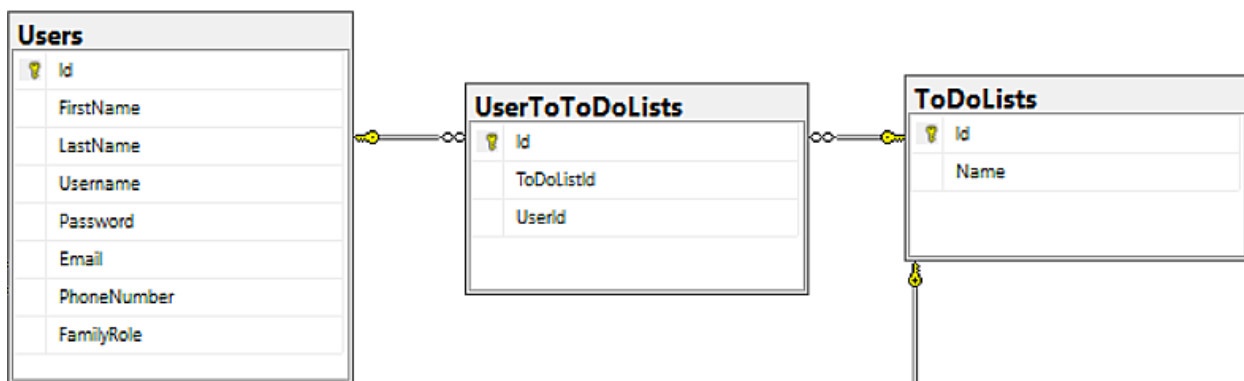
- **Id** – уникален идентификатор, използван за различаване на задачите;
- **Title** – заглавие на задачата;
- **Description** – подробно описание на задачата, което изяснява детайлите и очакванията от нея;

- **Status** – текущо състояние на задачата, което е от тип Enum и позволява следенето на прогреса. Възможните стойности са:
 - **TODO** –задачата е създадена, но не е започната
 - **InProgress** – задачата е в процес на изпълнение
 - **Done** – задачата е завършена
- **Deadline** – крайният срок за изпълнение на задачата
- **OwnerId** – идентификатор на създателя на задачата
- **AssigneeId** – идентификатор на изпълнителя на задачата
- **ToDoListId** – идентификатор на списъка, към който принадлежи задачата

Съчетанието на всички тези обекти и атрибути в модела на базата от данни позволява създаването на динамична и гъвкава система, която отговаря на разнообразните нужди на всяко семейство.

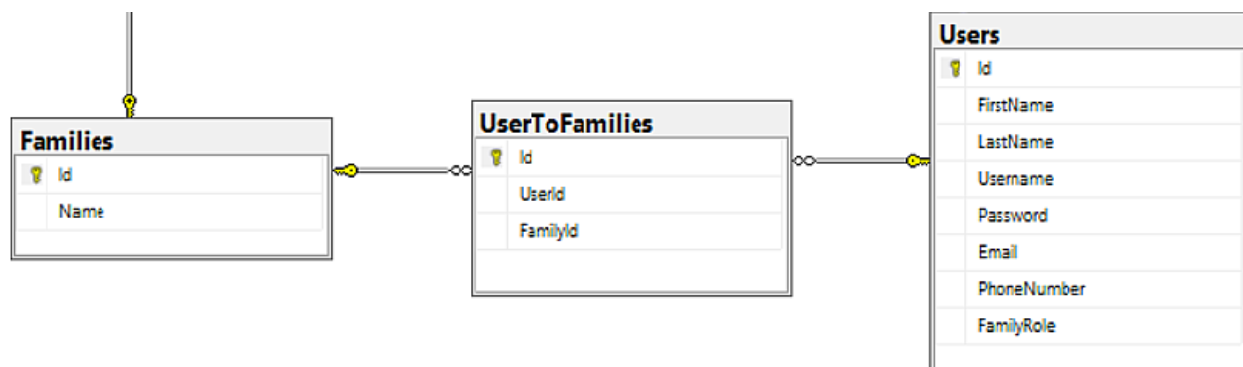
Междинните таблици са от съществено значение в системата за управление на семейните активности. По този начин се улесняват сложните връзки между потребителите, семействата, календарите и списъците със задачи. Чрез използването на тези междинни структури, системата гарантира, че потребителите могат да взаимодействат безпроблемно със споделени и лични ресурси.

Таблицата **UserToDoList** свързва потребителите с техните създадени списъци със задачи, позволявайки на множество потребители да имат достъп и да управляват споделени задачи. Структурата ѝ е:



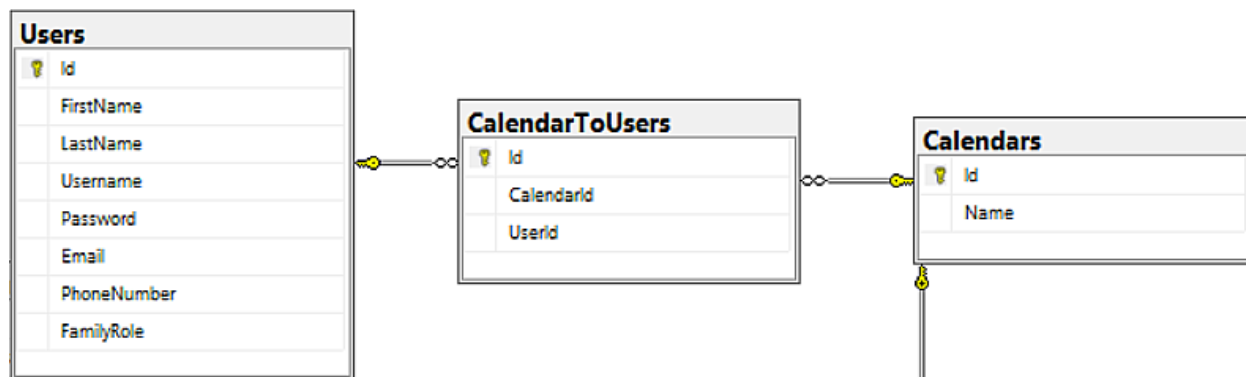
Фиг.15: Клас диаграма на връзката(M:M) между User и ToDoList

UserToFamily таблицата установява връзките между потребителите и техните семейства, позволявайки на потребителите да бъдат членове на няколко семейства.



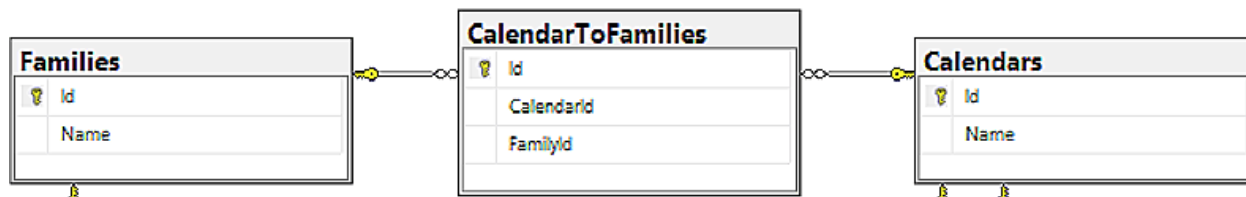
Фиг.16: Клас диаграма на връзката(M:M) между User и Family

Таблицата **CalendarToUser** позволява на потребителите да имат лични и споделени календари, подобрявайки процеса на планиране.



Фиг.17: Клас диаграма на връзката(M:M) между User и Calendar

CalendarToFamily оптимизира планирането на семейни събития, което улеснява синхронизацията между различните семейства. Връзката между календар и семейство, позволява на всички членове от съответното семейство да преглеждат и управляват семейните събития вътре.



Фиг.18: Клас диаграма на връзката(M:M) между Family и Calendar

Всяка от тези междинни връзки е от съществено значение за пълната структура на системата, предоставяйки плавен поток между всички участници и техните активности.

2. Сървиси и тяхната значимост при организирането на архитектурата в рамките на проекта

Сървисите в системата са създадени с цел да изолират голяма част от логиката, която се намира в контролерите. Тяхната отговорност е да съхраняват информация за класовете в системата и да изпълняват основни операции като извличане, обработка, редактиране и изтриване на данните от базата. В тези сървиси се извикват репоситори класове, с помощта на които се отделя основната логика. Благодарение тази комбинация от сървис с репоситори се намалява и предотвратява повторението на код.

Методите дефинирани в сървисите могат да се извикат единствено в контролерите, където не се изисква допълнителна информация за данните и връзката им в базата. Така за всеки клас има създаден отделен собствен сървис, създавайки се капсулация на обекта и неговата обработка. Затова, когато един обект има специфична логика за редактиране или изтриване, тя се дефинира в сървис класа. Това ни гарантира простота на контролерите и лесна поддръжка.

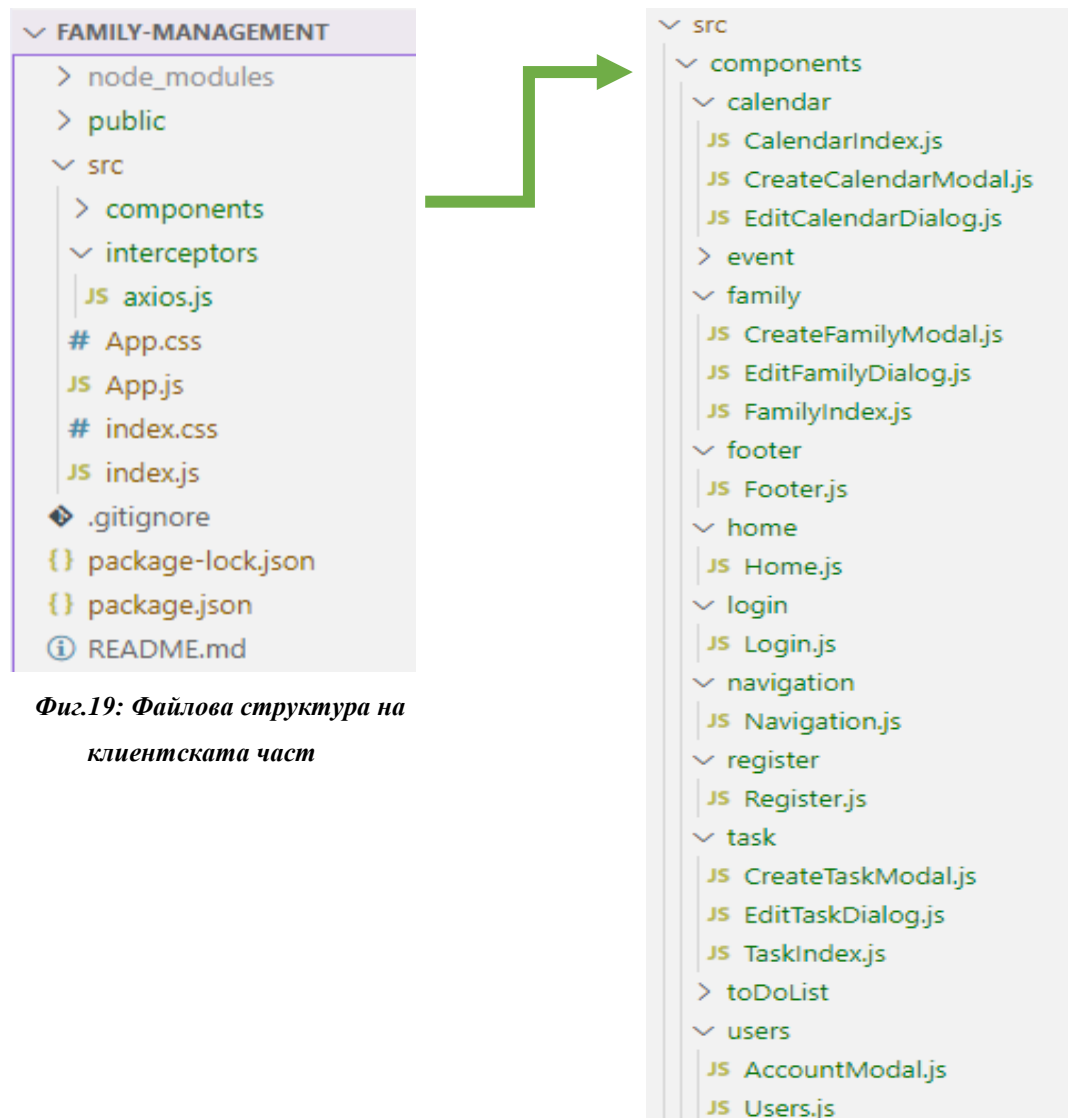
3. Модули в системата

Структурата на софтуерния продукт е разделена на два основни модула: frontend и backend, които осигуряват плавно и безпроблемно използване от потребителите.

- **Frontend (UI/UX)** – Клиентската част е разработена с React и е отговорна изцяло за потребителското изживяване и интерфейс. Всички компоненти, използвани за

визуализацията, се съхраняват в папката “src”. Те са организирани в индивидуални папки, всяка съдържаща JavaScript файлове, които от своя страна осигуряват тяхната функционалност и визуална презентация. Например, в папката “calendar” се намират файловете “CalendarIndex.js”, “CreateCalendarModal.js” и “EditCalendarModal.js”, които управляват всички аспекти на календарната функционалност в приложението. Отделно извън “src”, имам папка “interception” с файл “axios.js” за управление на HTTP заявките .

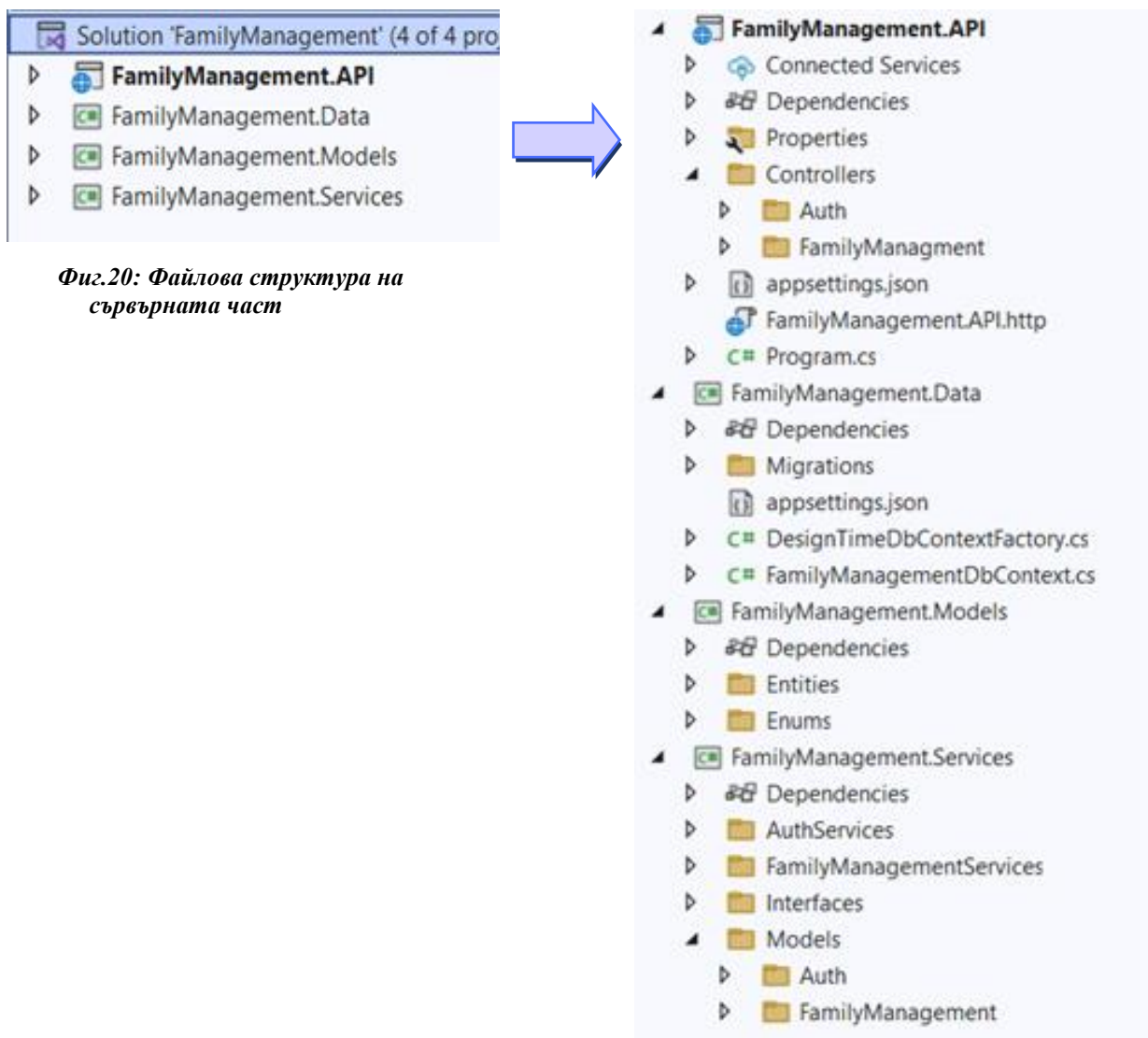
А централният App.js играе ключова роля в координацията на всички подкомпоненти, като гарантира гладко потребителско изживяване и интегрирана пълната функционалност между различните части на интерфейса.



Фиг.19: Файлова структура на клиентската част

- **Backend** – Сървърната част е изградена с ASP.NET Web API и структурирана в четири основни проекта, които поддържат ясно разделение на отговорностите.

Проектът **FamilyManagement.API** съдържа основните API контролери, които обработват входящите заявки от потребителите. **FamilyManagement.Data** отговаря за управлението на всички операции, свързани с данните. Това включва връзката с базата данни и миграциите за обновлението ѝ. В **FamilyManagement.Models** се съдържат дефинираните модели, които представят таблиците в базата данни. И накрая **FamilyManagement.Services** е отговорен за бизнес логиката на приложението.



Фиг.20: Файлова структура на сървърната част

4. REST API

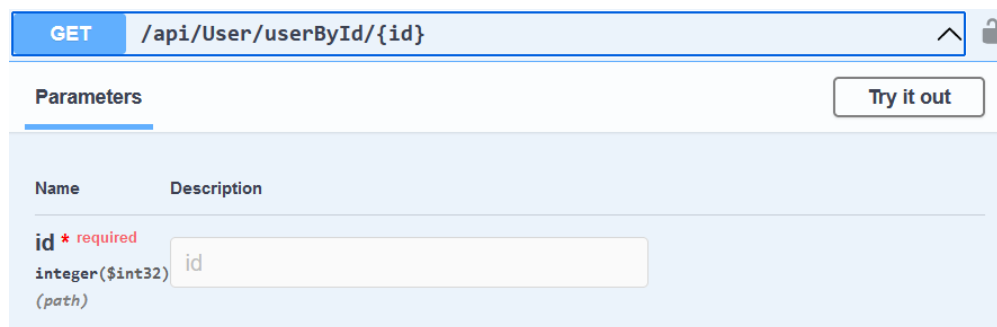
Representational State Transfer (REST) е архитектурен стил за изграждане на уеб услуги. Заявките REST се извършват чрез HTTP, същите HTTP заявки, които уеб браузърите използват за извличане на уеб страници и изпращане на данни към сървъри. Основните четири HTTP метода могат да бъдат съпоставени с CRUD операциите^[12], както следва:

- **GET** – извличане на данните
- **POST** – създаване на нов елемент
- **PUT** – актуализиране на елемент от данни
- **PATCH** – актуализиране на елемент от базата данни чрез описание на набор от инструкции за това как елементът трябва да бъде модифициран.
- **DELETE** – изтриване на елемент от данните

Използването на REST API за реализацията на софтуерният продукт за управление на семейни активности ни осигурява редица ползи, които улесняват разработката, мащабирането и поддръжката му. В следните няколко реда са разгледани някои важни имплементации на REST в системата:

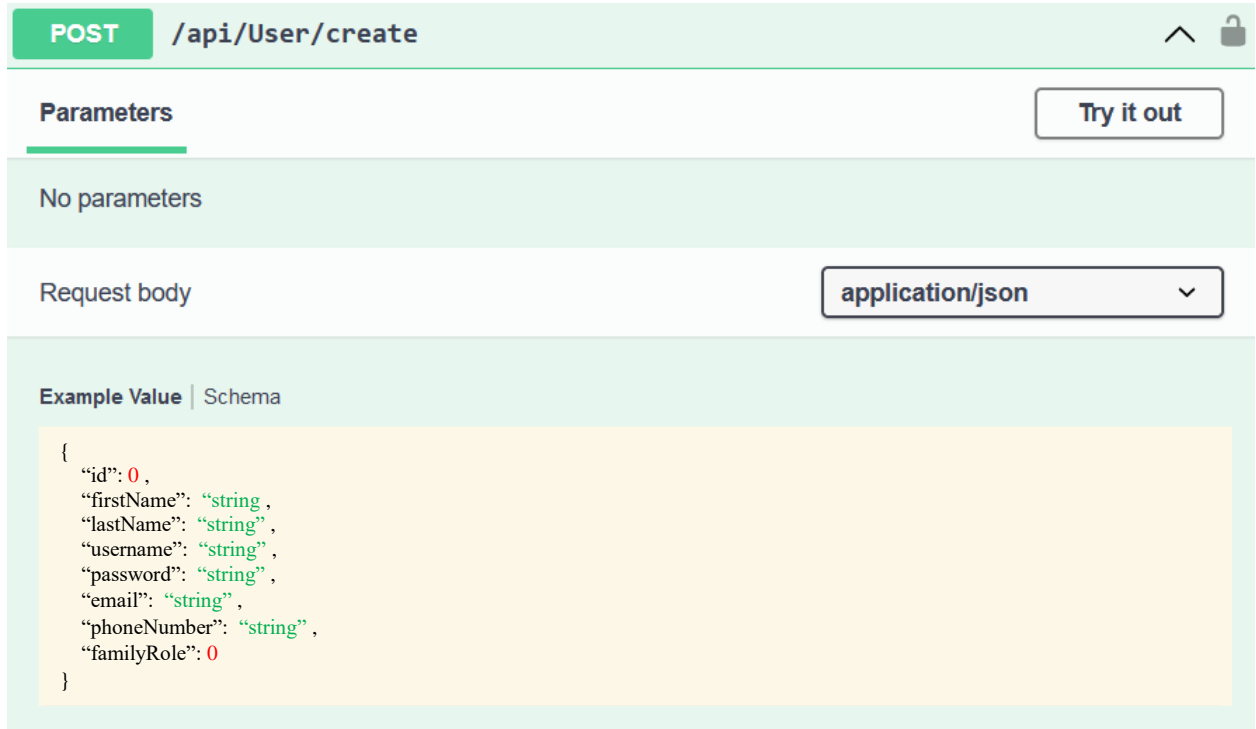
Управление на Потребителски Профил:

- REST Endpoint: ‘GET /api/User/userById/{id}’ - извлича детайли за потребителския профил, посредством неговото Id. Това позволява на потребителя да разгледа и да редактира своя профил след успешно влизане.



Фиг.21: Swagger UI за GET заявка на потребител по ID

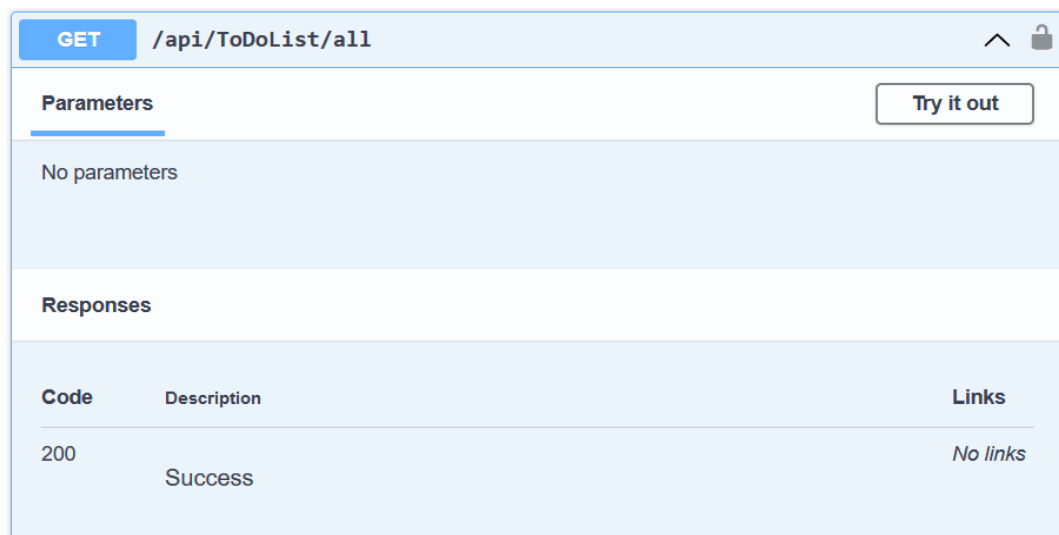
- REST Endpoint: **'POST /api/User/create '** - регистрира нов потребител в системата след подаване на необходимите данни като име, електронна поща и парола.



Фиг.22: Създаване на потребител чрез POST заявка

Управление на Списъци:

- REST Endpoint: **'GET /api/ToDoList/all'** - извлича всички списъци със задачи, асоциирани с потребителя или семейството.



Фиг.23: Извличане на всички списъци чрез заявка GET

- REST Endpoint: **'PATCH /api/ToDoList/edit/{id}'** - частично обновление на елемента. Метод PATCH позволява на потребителите да изпратят само тези части от данните, които искат да бъдат актуализирани, вместо всички данни за обекта.

PATCH /api/ToDoList/edit/{id}

Parameters Try it out

Name	Description
id * required integer(\$int32) (path)	id

Request body application/json

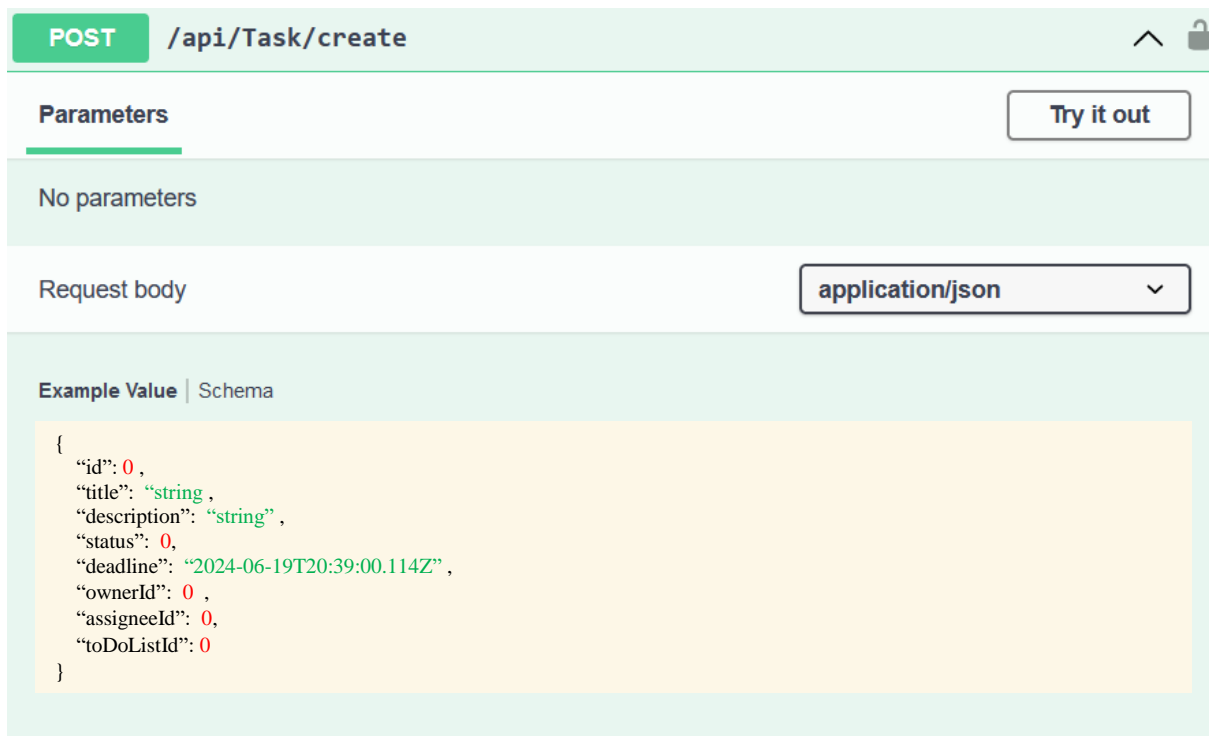
Example Value | Schema

```
{
  "id": 0,
  "name": "string",
  "ownerId": "string"
}
```

Фиг.24: Редактиране на списък с PATCH заявка

Управление на Задачи:

- REST Endpoint: **'POST /api/Task/create '** - създава нова задача в системата след подаване на необходимите данни.



POST /api/Task/create

Parameters Try it out

No parameters

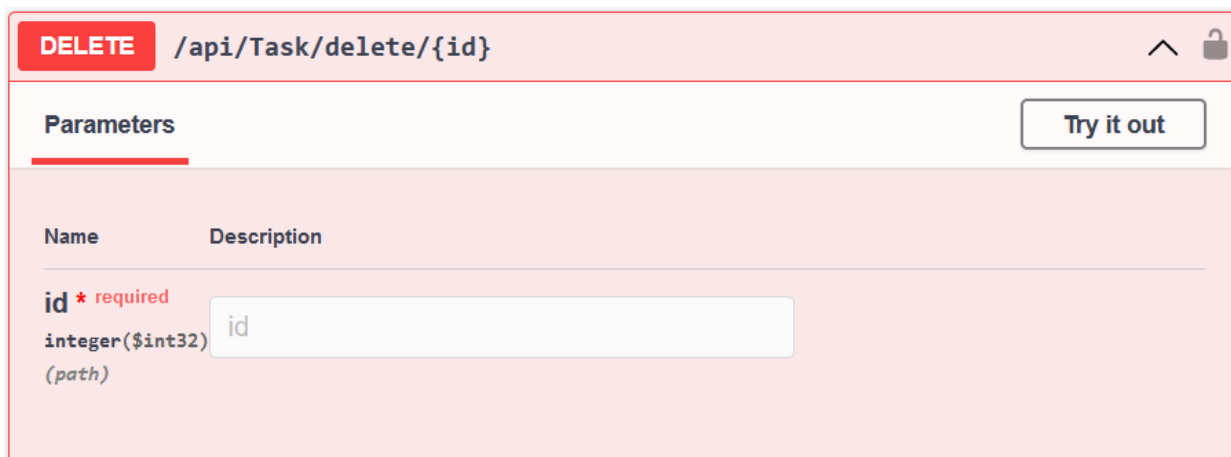
Request body application/json

Example Value | Schema

```
{
  "id": 0,
  "title": "string",
  "description": "string",
  "status": 0,
  "deadline": "2024-06-19T20:39:00.114Z",
  "ownerId": 0,
  "assigneeId": 0,
  "toDoListId": 0
}
```

Фиг.25: Създаване на задача списък с POST заявка

- REST Endpoint: ‘GET /users/{userId}’ - извлича детайли за потребителския профил, посредством неговото Id. Това позволява на потребителя да разгледа и да редактира своя профил след успешно влизане.



DELETE /api/Task/delete/{id} Try it out

Parameters

Name	Description
id * required integer(\$int32) (path)	id

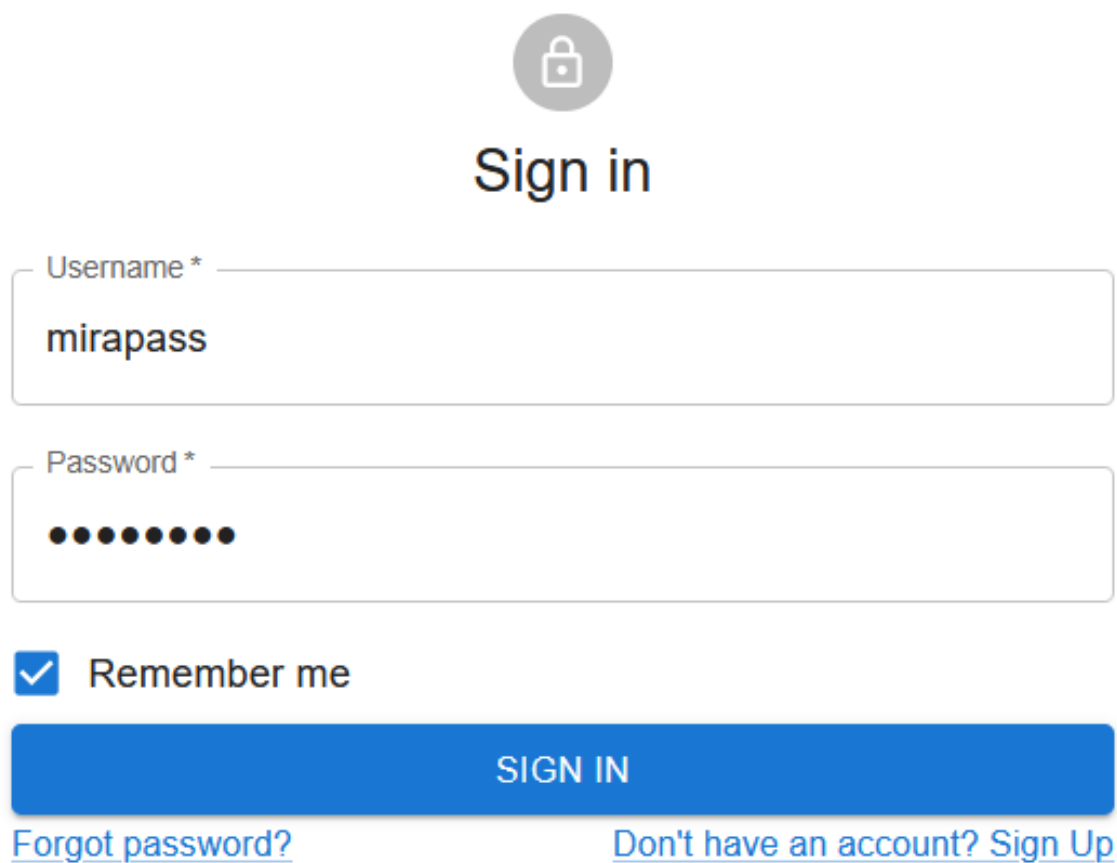
Фиг.26: Изтриване на задача чрез заявка Delete

ГЛАВА V

Наръчник на потребителя

1. Вход и Регистрация в системата

Системата поддържа вход на потребителите с автентикация. Всеки потребител, който вече има профил, може да влезе в системата, въвеждайки своето потребителско име и парола във формата за вход. На *Фиг.27* е показана формата за вход, където потребителя трябва да въведе своите данни за автентикация.



The image shows a 'Sign in' form. At the top is a lock icon inside a circle, followed by the text 'Sign in'. Below this are two input fields: 'Username *' containing 'mirapass' and 'Password *' containing ten dots. There is a checked checkbox labeled 'Remember me'. A blue 'SIGN IN' button is below the inputs. At the bottom are two links: 'Forgot password?' and 'Don't have an account? Sign Up'. Two blue arrows point from the bottom towards these links.

Username *

mirapass

Password *

●●●●●●●●●●

☒ Remember me

SIGN IN

[Forgot password?](#) [Don't have an account? Sign Up](#)

Фиг.27: Форма за Автентикация

Потребителят има възможност да смени паролата си, чрез натискане на бутона „**Забравена парола?** “ Ако потребителя няма акаунт в системата, веднага може да се направи регистрация с избиране на бутона „**Нямате акаунт? Регистрирайте се** “.

Формата за вход се състои от:

- Поле за потребителското име;
- Поле за паролата;
- Поле с отметка за запомняне на потребителския вход от браузъра;
- Бутон за вход.

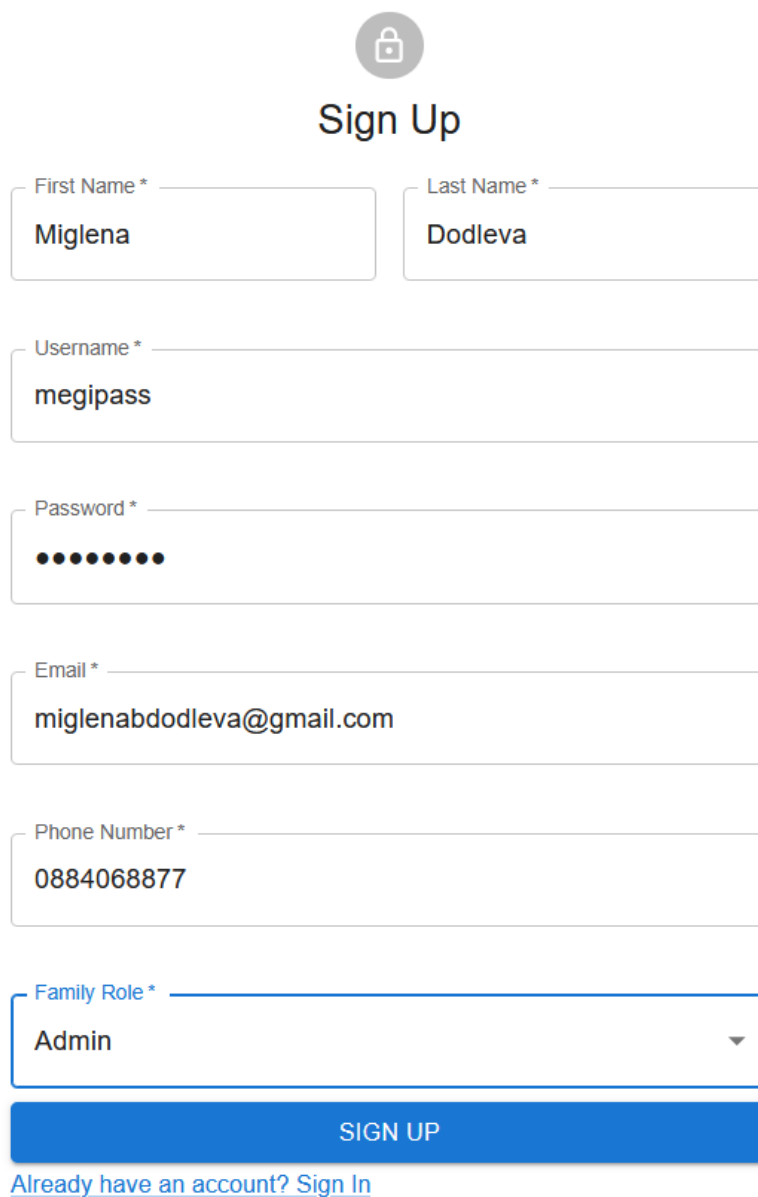
След въвеждане на необходимите данни и натискане на бутона за Вход, се стартира бизнес логика, която проверява за потребител с такова потребителско име и парола в базата данни. При успешен вход в системата се зарежда началната страница, а при грешна комбинация – страницата се презарежда със съобщение за грешка.

За новите клиенти на системата е разработена регистрационна форма, която може да се достъпи чрез натискане върху текста бутона „**Нямате акаунт? Регистрирайте се** “ от Входа.

Формата за регистрация включва:

- Поле за собствено име;
- Поле за фамилно име;
- Поле за потребителско име;
- Поле за паролата;
- Поле за имейл адрес;
- Поле за телефонен номер;
- Поле с избор на ролята на потребителя. Възможни роли: Админ(Admin) и Член(Member);
- Бутон за регистрация.

На следващата *Фиг.28* е представена подробна визуализация на формата:



The image shows a 'Sign Up' form with a lock icon at the top. The form contains several input fields: 'First Name *' with the value 'Miglena', 'Last Name *' with the value 'Dodleva', 'Username *' with the value 'megipass', 'Password *' with masked characters, 'Email *' with the value 'miglenabdodleva@gmail.com', 'Phone Number *' with the value '0884068877', and a 'Family Role *' dropdown menu currently set to 'Admin'. Below the form is a blue 'SIGN UP' button and a link that says 'Already have an account? Sign In'.

Sign Up

First Name *
Miglena

Last Name *
Dodleva

Username *
megipass

Password *
●●●●●●●●

Email *
miglenabdodleva@gmail.com

Phone Number *
0884068877

Family Role *
Admin

SIGN UP

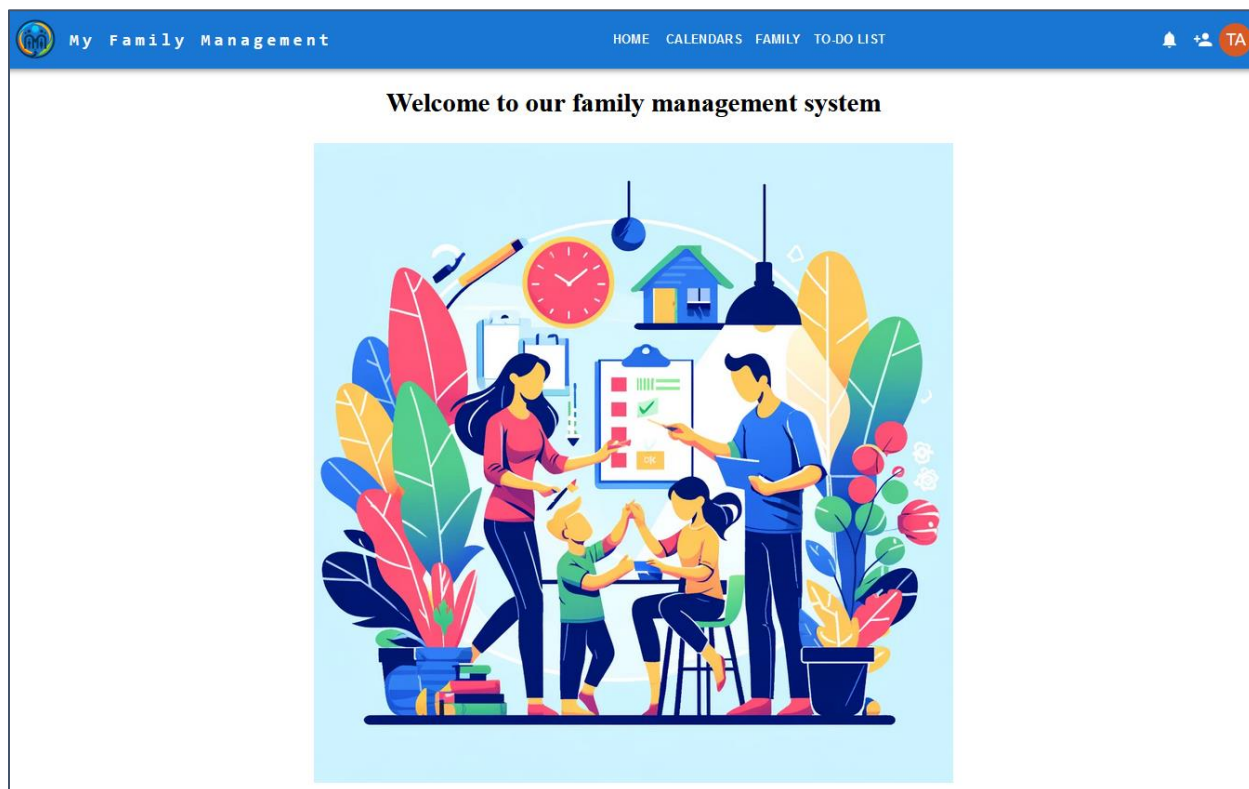
[Already have an account? Sign In](#)

Фиг.28: Форма за Регистрация в системата

След попълването на всички необходими данни, системата ги проверява и връща съобщение за грешка, ако са некоректни. При успешна регистрация, потребителския профил се активира и новият потребител може да използва удостоверенията си за вход в системата.

2. Начална страница

След успешен вход, потребителят се пренасочва към началната страница. Тя е проектирана да предоставя бърз достъп до всички основни функции на системата за управление на семейните активности.



Фиг.29: Визуализация на началната страница в системата

Както е изобразено на Фиг.29, в центъра на началната страница е разположена илюстрация на семейство, ангажирано с планирането на тяхното време и разпределяне на задачи. Тя представя главните цели на софтуерния продукт. В горната част на страницата се намира навигационната лента, чрез която потребителите могат да достъпват всичко необходимо. Състои се от следните елементи:

Фиг.30: Навигационна лента



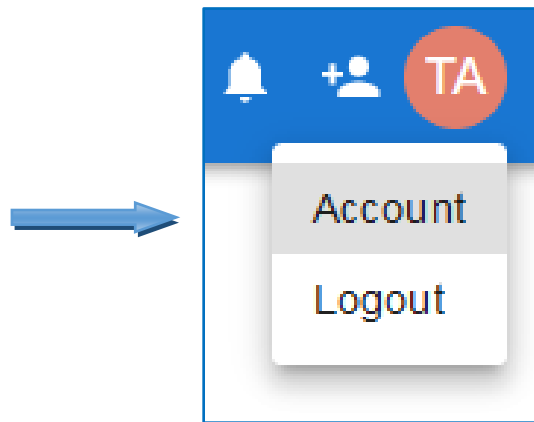
Лого: Всяко кликане върху него връща потребителя на началната страница

В средата на лентата имаме разположени главните секции на приложението. Чрез тях потребителите бързо могат да достигнат до нужната им информация. Всяка от секциите предоставя специфични функции и инструменти за нуждите на семейното управление.

Икони за потребителя: Те съдържат бързи връзки за известия, добавяне на нов член и персонализирани настройки на профила

3. Управление на профила

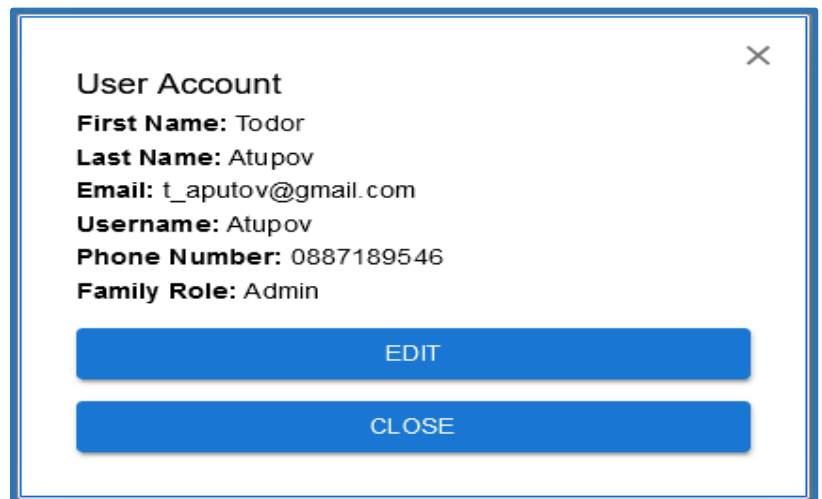
Всеки потребител може да преглежда и редактира своите лични данни, като име, фамилия, електронна поща, телефонен номер и роля в системата. Настройките на профила са достъпни чрез натискане на иконата на профила в горния десен ъгъл на интерфейса, където се разкриват следните две опции: "Account" (Профил) и "Logout" (Изход). Аватарът на потребителя в горния десен ъгъл е изработен от първите букви на собствено име и фамилията. Това улеснява разпознаването на профила и предава индивидуалност.



Фиг.31: Настройки на профила

Когато потребителят кликне върху опцията "Account", се отваря прозорец, който показва текущите регистрирани данни за профила:

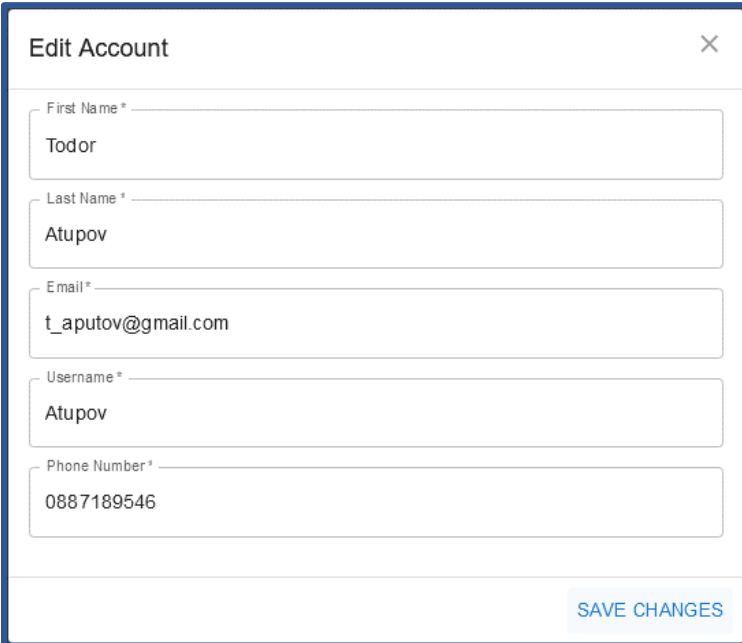
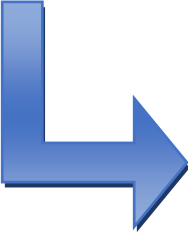
- Собствено име
- Фамилно име
- Електронна поща
- Потребителско име
- Телефонен номер
- Роля в системата



Фиг.32: Изглед на потребителските данни

За да редактира профилните си данни, потребителят трябва да натисне бутона "Edit" (Редактирай), което отваря нов прозорец с форма за редактиране:

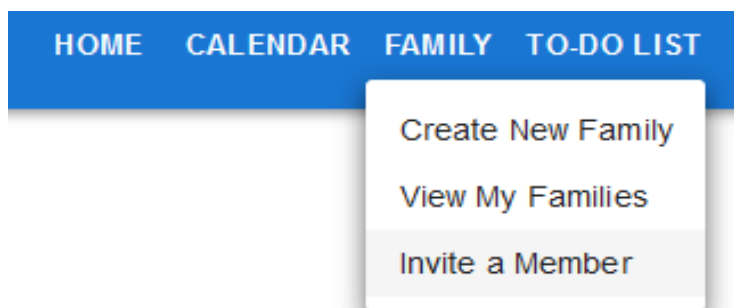
- Полетата са предварително попълнени с текущата информация.
- Потребителят може да промени която и да е от предоставената информация.
- След направените промени на данните, трябва да бъде натиснат бутона "Save Changes"(Запази Промените).



Фиг.33: Форма за редактиране на информацията за акаунта

4. Управление на Семейства

Потребителите в системата могат да създават, преглеждат, редактират и управляват различни семейни профили. Този модул е изключително полезен за координиране на задачи и събития, специфични за всяко семейство. При натискане на „Family“ от навигационната лента, за потребителя се разкриват няколко опции: Създай Семейство, Разгледай всички семейства, към които принадлежи акаунта и възможността да се изпрати покана за присъединяване:



Фиг.34: Падащо меню към управлението на семейства










За да създадете ново семейство, потребителя трябва да кликне върху бутона „Create New Family“ (Създай Ново Семейство). Появява се форма за създаване и бутон „Create“:

Create New Family

CREATE

Фиг.35: Форма за създаване на семейство

След успешно създаване, потребителя се препраща към главната страница за управление на семейства. Тя се състои от списък със семейства и информация за тях. Налични са следните полета: Име на семейство, брой членове и действия, които потребителя може да извърши като редактиране, изтриване или преглед на членовете.













CREATE FAMILY		
Name	Members Count	Actions
Dodlevi	3	  
Gadzhevi	2	  
Todorovi	6	  

Фиг.36: Таблица с наличните семейства


5. Управление на Списъци със задачи




Системата предоставя на потребителите удобен начин за организиране и управление на различни задачи, свързани с ежедневието. В тази секция потребителите могат да създават, преглеждат и редактират списъци със задачи, като всяка задача може да бъде детайлно описана и планирана. След натискането на “To Do List” от навигационния бар, пред потребителя се отваря таблица пълна със създадени от него списъци , или списъци в които участва. В горния ляв ъгъл се намира бутона за създаване на нов списък. Таблицата в центъра на страницата се състои от полета, пълни с информация за списъка, като име,

брой задачи и действия, които могат да се извършат с него: редактиране, изтриване и преглед на задачите вътре.

CREATE TO-DO LIST		
Name	Task Count	Actions
Trip to Venice	0	  
Birthday Party	0	  
Homework	0	  
Weekly Shopping	0	  

Фиг.37: Таблица със списъци

Както се вижда от *Фиг.35* списъците, които имаме са празни. За да се създаде нова задача, първо потребителя трябва да кликне върху бутона за преглед на задачите →  на избора от него списък, към който ще принадлежи задачата. Новата заредена страница, в която се визуализират задачите и информацията за тях е празна, понеже още нямаме създадена задача. Това ще се коригира бързо, когато потребителя натисне върху бутона „Create Task“ (Създай Задача).

 My Family Management HOME CALENDAR FAMILY TO-DO LIST   TA						
CREATE TASK						
Title	Description	Status	Deadline	Owner	Assignee	Actions

Фиг.38: Визуализация на празен списък

След натискане на този бутон, пред потребителя ще се появи форма за попълване на данни, нужни за създаването на задача.

Трябва да се попълнят следните полета:


- Заглавие
- Подробно описание
- Статус:
 - За Изпълнение
 - В процес
 - Изпълнена
- Краен срок на изпълнение
- Създател
- Изпълнител на задачата



Фиг.39 : Форма за създаване на задача

Фиг.40: Визуализация на избор за дата за краен срок

След попълването на всички необходими данни и натискането на бутона "Create Task" (Създай задача), задачата се записва в системата. Потребителят може веднага да види новосъздадената от него задача в общата таблица, където се отразява нейният статус и детайли (Фиг.41). Това позволява лесно проследяване на напредъка по задачата и управлението на отговорностите както от страна на създателя, така и от изпълнителя.




My Family Management


HOME

CALENDAR

FAMILY



TO-DO LIST





TA










CREATE TASK

Title	Description	Status	Deadline	Owner	Assignee	Actions
Book Hotel	Research best places to sty in Venice.	To Do	1/7/2024	Todor Atupov	Miglena Dodleva	<div></div>


Фиг.41: Таблица със задачи към даден списък

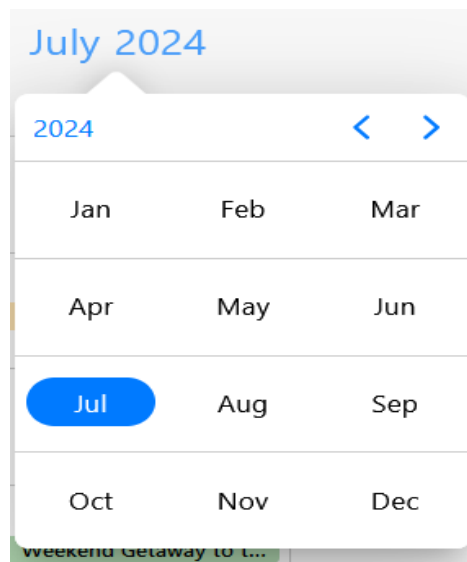
6. Управление на Календари и Събития

Системата предлага интуитивен интерфейс за управление на календари и събития, което позволява на потребителите да организират своите ангажименти ефективно. В тази секция, потребителите могат да добавят, преглеждат и редактират календари заедно с техните събития. След натискане върху “**Calendar**” от менюто за навигация, потребителят вижда таблица със създадени от него календари. В горния ляв ъгъл е разположен бутона за създаване на нов календар, а в центъра на страницата се визуализира общата им информация, като име, брой събития и възможни действия (редактиране, изтриване, преглед).

My Family Management			
HOME		CALENDAR	FAMILY TO-DO LIST
CREATE CALENDAR			
Name	Event Count	Actions	
Work	5	  	
School Calendar	0	  	
Travels and Vacations	4	  	

Фиг.42: Таблица с календари

След избор на календар и натискане върху опцията за преглед  , на екрана се зарежда календарен изглед, който показва всички текущи събития и ангажименти в него. В горната част на страницата се намират бутони за навигация между месеците, както и бутон за добавяне на ново събитие.



Фиг.43: Навигация между месеци и години

CREATE EVENT						
July 2024						
< Today >						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	Jul 1	2	3	4	5	6
				Cultural Festival Visit		
7	8	9	10	11	12	13
Cultural Festival Visit						
14	15	16	17	18	19	20
					Weekend Getaway to t...	
21	22	23	24	25	26	27
Weekend Getaway to t...						
28	29	30	31	Aug 1	2	3
4	5	6	7	8	9	10
Girls' Trip to Venice						

Фиг.44: Календарен изглед на събития

За добавяне на събитие, потребителят трябва да кликне върху бутона "Create Event" (Създай Събитие). Това отваря форма, където се въвеждат детайли за събитието:

- Заглавие
- Подробно описание
- Начална дата: дата + час
- Крайна дата: дата + час
- Дали събитието е цял ден?
 - Възможни отговори: Да или Не
- 7. Избор на цвят за събитието
- 8. Избор за повтораемост
 - Не се повтаря
 - Ежедневно
 - Ежеседмично
 - Ежемесечно
 - Ежегодишно

Create Event

Event Title *

Event Description *

Start Date *
mm / dd / yyyy , -- : -- --

End Date *
mm / dd / yyyy , -- : -- --

Is a Full Day Event?
No

Theme Color

Repeat Type

CREATE EVENT

Create Event

Event Title *
One-day Family Beach Vacation

Event Description *
Spend quality time with family on the sandy beaches. Build sandcastles, surf, and relax under the sun.

Start Date *
07 / 01 / 2024 , 12 : 00 PM

End Date *
07 / 01 / 2024 , 09 : 00 PM

July 2024

Sun Mon Tue Wed Thu Fri Sat

30 1 2 3 4 5 6

7 8 9 10 11 12 13

14 15 16 17 18 19 20

21 22 23 24 25 26 27

28 29 30 31 1 2 3

4 5 6 7 8 9 10

Clear

CREATE EVENT

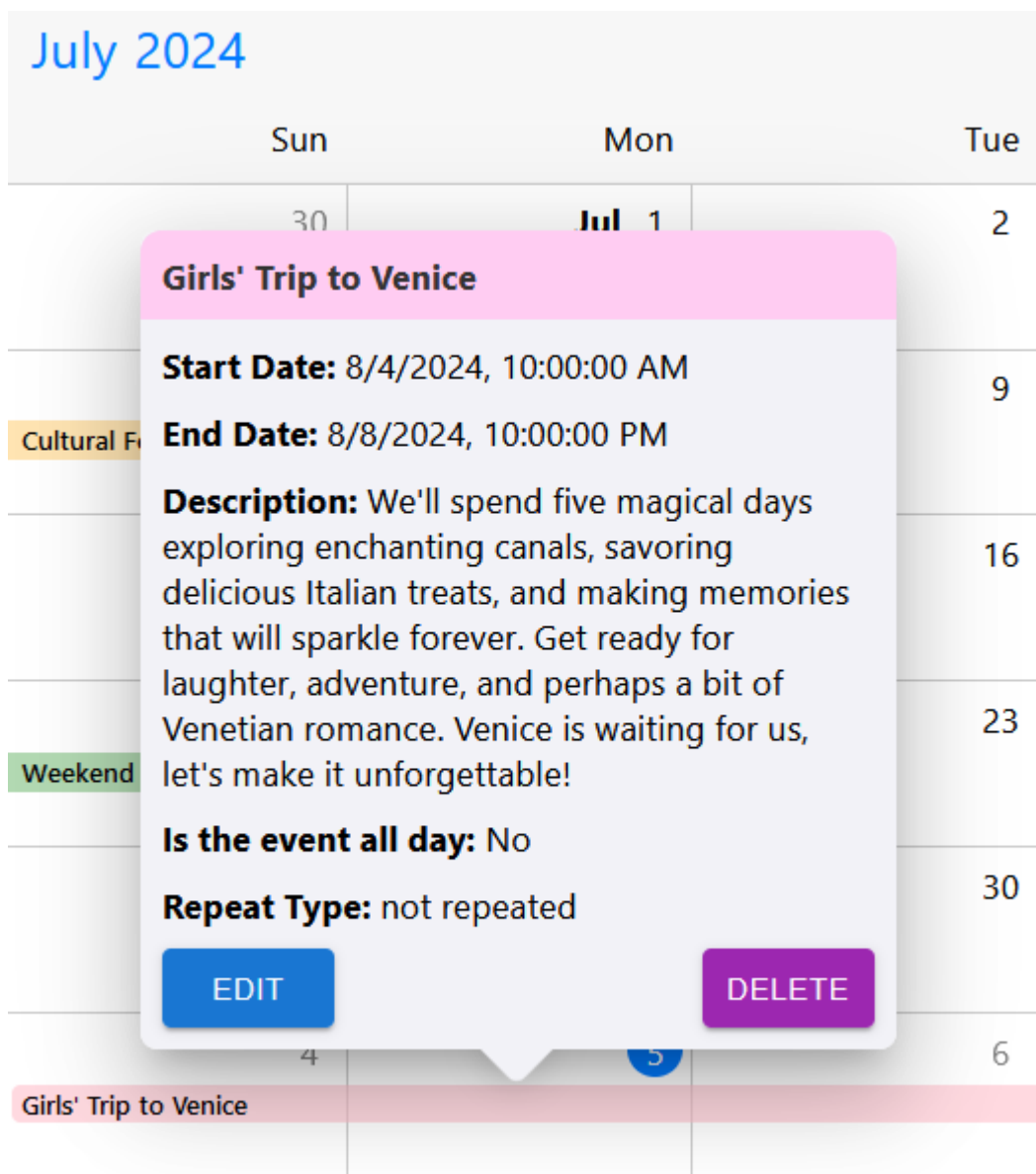
Фиг.45: Форма за създаване на ново събитие
(празна форма => попълнена ✓)

След попълването на всички необходими полета и натискането на бутона "Create Event", събитието автоматично се добавя в календара и става видимо на избраната дата.

July 2024		
Sun	Mon	Tue
30	Jul 1 One-day Family Beach ...	2
7 Cultural Festival Visit	8	9
14	15	16

Фиг.44: Визуализация на новосъздаденото събитие

Потребителите също така могат да редактират, изтриват и преглеждат детайли към съществуващи събития. За подробен преглед на събитие, потребителят трябва да натисне веднъж върху него. Това действие ще отвори прозорец с пълна информация:



Фиг.45: Преглед на детайлите за събитие

В този изглед могат да се използват бутоните "Edit" (Редактирай) и "Delete" (Изтрий) за съответните действия. Прозорецът може да се затвори с двойно кликване върху събитието. Това предоставя на потребителите по-голям контрол и улеснява организирането на техните ангажименти.

Заключение

В рамките на настоящата дипломната работа е разработена система, която улеснява и оптимизира ежедневието на семействата. Системата предоставя лесен и иновативен подход за планиране, управление и проследяване на разнородни задачи, събития и дейности. Софтуерният продукт е целесъобразен с нуждите на малки и големи семейства.

Потребителите на системата могат да създават и контролират задачи, да планират събития в лични или общи календари и да разпределят равномерно отговорностите между членовете на семейството. Продуктът улеснява въвеждането и прегледа на информация за предстоящи и изпълнени задължения и ангажименти, което подобрява комуникацията и координацията в рамките на семейството.

Софтуерната система успешно постигна поставените пред нея цели и отговаря на всички изисквания, като демонстрира ефективност и практичност в управлението на семейни активности. За да бъде още по-цялостна и гъвкава към непрестанно променящите се нужди на съвременните семейства, бъдещото ѝ развитие може да се фокусира върху следните аспекти:

- Мобилна версия;
- Модул за управление на финансите;
- Модул за семейни форуми и чат групи;
- Игрови елементи.

Използвана литература:

- [1] [Microsoft Learn, "SQL Server 2019", Nov/2023](#)
- [2] [React.dev, Overview](#)
- [3] [Reactjs.org, Documentation "Introducing JSX"](#)
- [4] [Material UI - Overview, Introduction](#)
- [5] [Material-UI components, Documantation](#)
- [6] [JWT.IO, 'Introduction to JSON Web Tokens'](#)
- [7] [MiniOrange, "What is JWT \(JSON Web Token\)? How does JWT Authentication work?"](#)
- [8] [Microsof Learn, ASP.NET Core web API documentation with Swagger / OpenAPI, June/2024](#)
- [9] [Swagger.io, Ttools => swagger-ui](#)
- [10] [Swagger.io, Documentation](#)
- [11] [Microsof Learn, REST in ASP.NET Core](#)
- [12] [Microsof Learn, "Enabling CRUD Operations in ASP.NET Web API", Nov/2022](#)
- [13] [Bytehide.com/blog , "How To Use Enum in C#", Jun/2023](#)
- [14] [Atharva Mahamuni, "C Enumeration \(enum\)", Nov/2023](#)
- [15] [Emotion.sh, Docomentation, "Introduction"](#)