**ДОКУМЕНТАЦИЯ**

НА ПРОЕКТ **“RECIPEBOOK”**



СЪСТАВЕНА ОТ:

**МЕЛАНИ АНГЕЛОВА**

**ДАНИЕЛ БУМБАЛОВ**

**ДАНИИЛ ШИРЯЕВ**

ДОПЪЛНИТЕЛЕН МАТЕРИАЛ ЗА ИЗПИТ КЪМ МОДУЛ **“РАЗРАБОТКА НА СОФТУЕР”**

ПРОГРАМА “ОБУЧЕНИЕ ЗА IT КАРИЕРА”

**СЪДЪРЖАНИЕ**

* За проекта
* Цели на проекта
* Етапи на създаване
* Разпределение на задачите
* Model
* View
* Controller
* Изображения
* Заключение

**ЗА ПРОЕКТА**

* RecipeBook е групов проект, който представлява приложение с трислойна архитектура, реализирано за работа в Desktop среда за Windows 10
* Приложението е създадено върху основата на UWP и е написано с C# и XAML код
* Проектът е с разширими възможности и предразполага към подобрения и нововъведения

**ЦЕЛИ НА ПРОЕКТА**

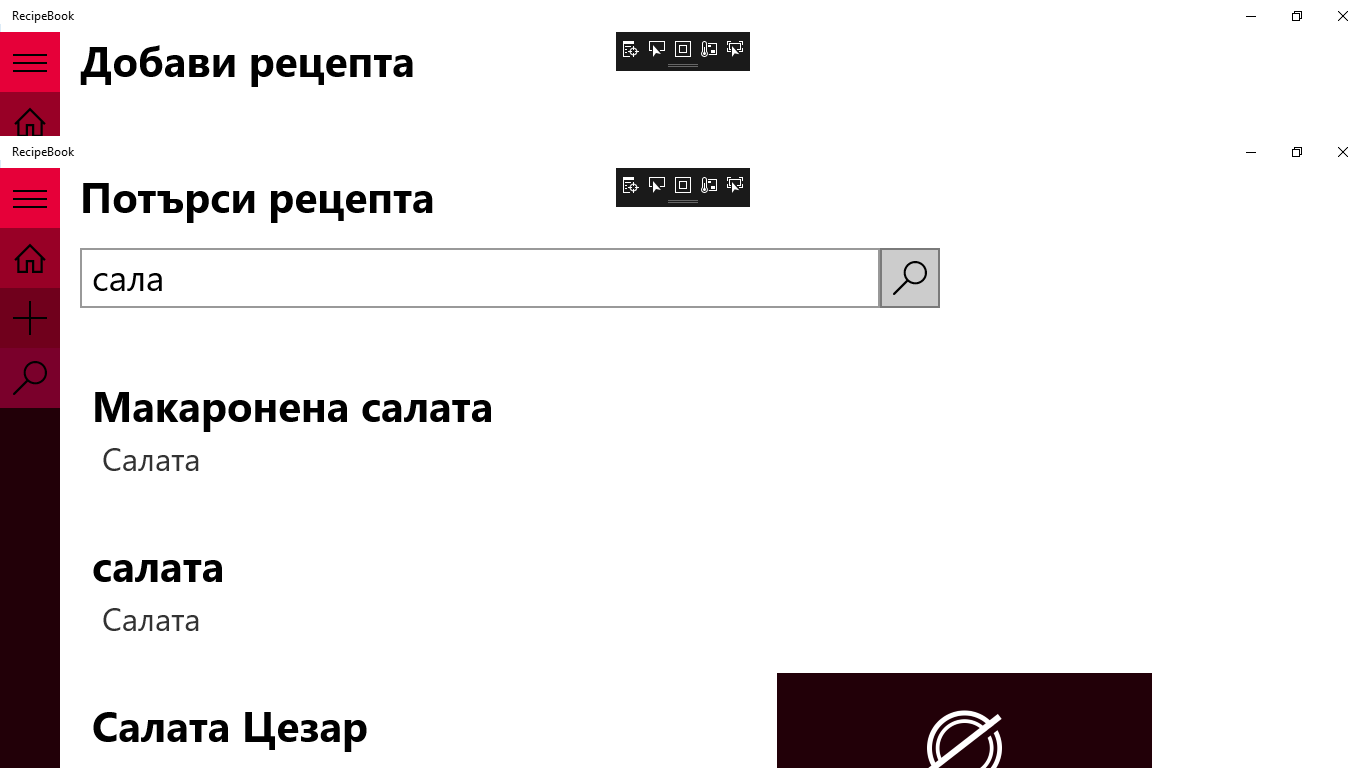
* Да се създаде функционално приложение, което да бъде ползотворно и смислено
* Да се създаде интуитивен потребителски интерфейс, който не изисква никакви специални умения, за да бъде използван

от потребителите

* Да се създаде опростен и лек дизайн, който не напряга потребителите с вида си и носи подходящо впечатление за проекта и неговата функционалност
* Проектът да бъде foolproof - потребителят във всеки момент от манипулирането на базата данни има възможността да редактира и промени информацията, която влага в нея; проектът се стреми да обхване възможни случайни и умишлени грешки и да предотврати нарушаване на дейността си, вследствие на такива, или да обработи неочакван вход от потребителя
* Проектът да бъде разбираем и приятен за използване

**ЕТАПИ НА СЪЗДАВАНЕ**

* Избиране на тема на проекта
* Определяне на идеята
* Обмисляне на функционалността
* Разпределяне на елементите на проекта между членовете на групата
* **//** Един от членовете на екипа напусна групата
* Избиране на типа проект (UWP)
* **//** Техническо време за осигуряване на Windows 10 на всички компютри в групата
* Начало на същинската разработка
* **//** Техническо време за усвояване работата с Git
* Установяване на общо хранилище на проекта в Git
* Създаване на първичната база данни
* Работа по дизайна
* **//** Техническо време за отстраняване на проблеми със съвместимостта на проекта за всички компютри, довело до пълно пренаписване на проекта и откриване на нов клон в Git
* Scaffolding и начало на осъществяване взаимодействията между приложението и базата данни
* **//** Техническо време в неуспешни опити за осъществяване на връзка към сървър от различни мрежи
* Създаване на контролера, осъществяване функционалността на програмата, взаимоотношенията между през. слой и базата данни



**РАЗПРЕДЕЛЕНИЕ**

**НА ЗАДАЧИТЕ**

**<МЕЛАНИ АНГЕЛОВА/>**

* Работа по базата данни и поддръжка
* Предлагане идеи по дизайна на проекта
* Поддържане на хранилището на проекта

**<ДАНИЕЛ БУМБАЛОВ/>**

* Работа по контролера и асистенция в дизайна
* Предлагане идеята на проекта
* Поддържане функционалността и изправността на проекта

**<ДАНИИЛ ШИРЯЕВ/>**

* Основна работа по дизайна
* Предлагане идеи по функционалността на проекта
* Организиране на структурата на страниците

**MODEL (БАЗА ДАННИ)**

* Базата данни е направена с **MySQL Workbench 8.0**
* Състои се от 5 таблици, определящи съдържанието на проекта
  + Recipes – рецептите
  + Products – продуктите
  + Categories – категория на ястието
  + Types – групата, към която продуктът спада
  + ProductsRecipes – „мост“ м/у “продукт” и “рецепта”
* **Таблица ‘recipes’**: представлява рецептите, който ще търсите или добавяте. Съдържа колони ‘id’(идентификационен номер), ‘name’(име), ‘description’(описание на рецептата), ‘category’(категорията, към която принадлежи рецептата) и ‘img\_name’(име на снимка за референция). Таблицата осъществява връзка към **‘categories’**.
* **Таблица ‘products’**: включва продуктите, които са необходими за рецептата. Състои се от ‘id’ и ‘name’, ‘type’(типът на продукта) и ‘price’(цената на определено количество от продукта). Таблицата осъществява връзка към **‘types’**.
* **Таблица ‘categories’**: Състои се от колона “id”, “name”- името на категорията.
* **Таблица** **‘types’**: съдържа видът продукт: плод, зеленчук, месо, риба, морски дарове, яйца, млечни продукти, хлебни изделия, подправки, ядки и други; Колони: “id” и “name”.
* **Таблица ‘products\_recipes’**: Пази отношенията между всяка рецепта и продуктите, участващи в нея, заедно с количеството(quantity) нужно за самия продукт на рецептата.
* **Scaffolding** – Свързването на приложението с базата данни (Entity Framework) Контекстът осъществява връзката.

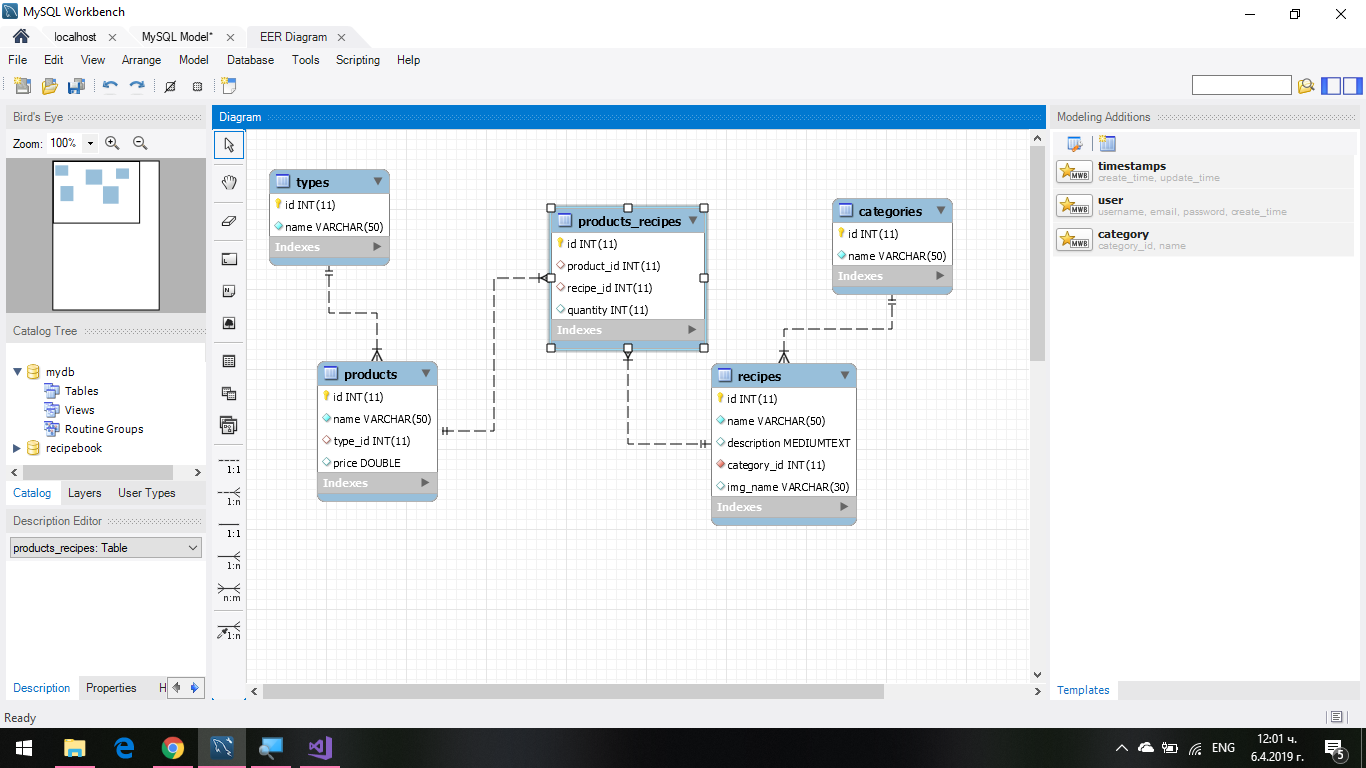
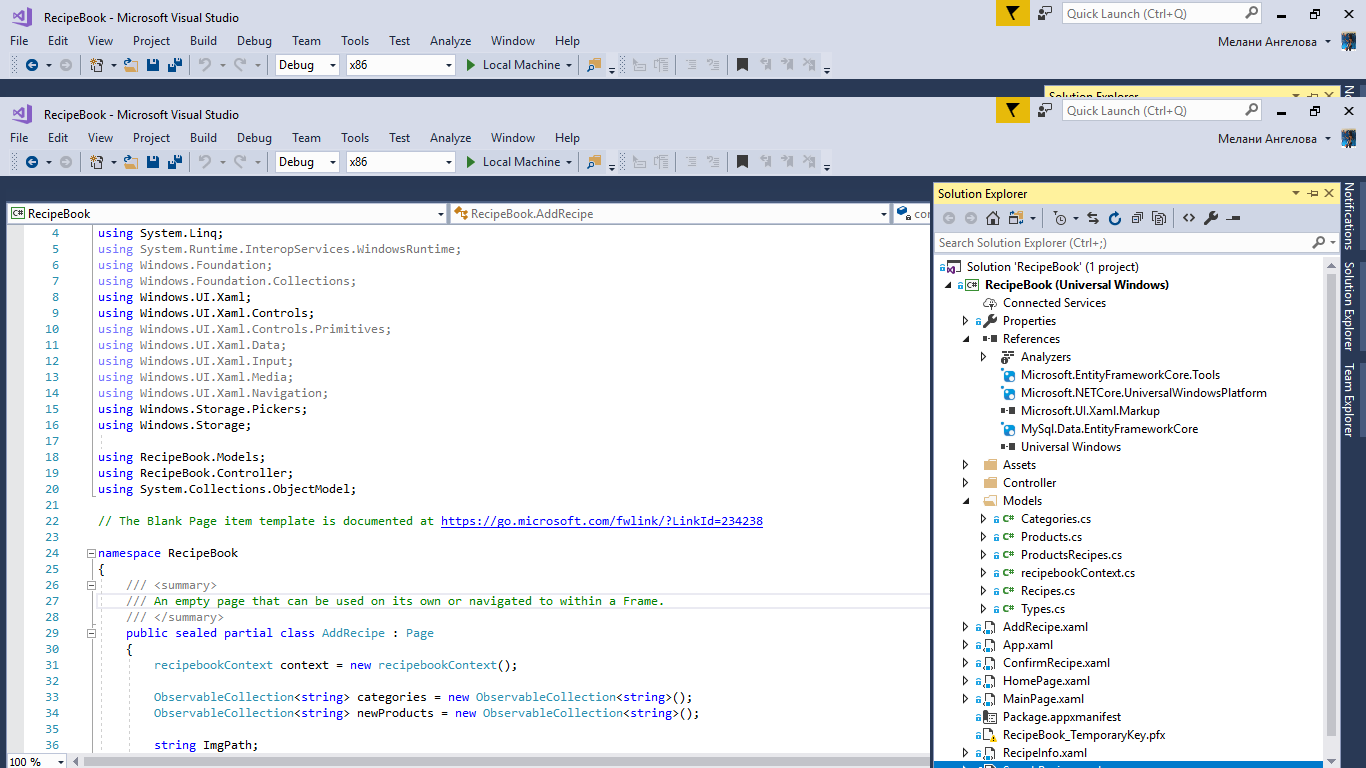
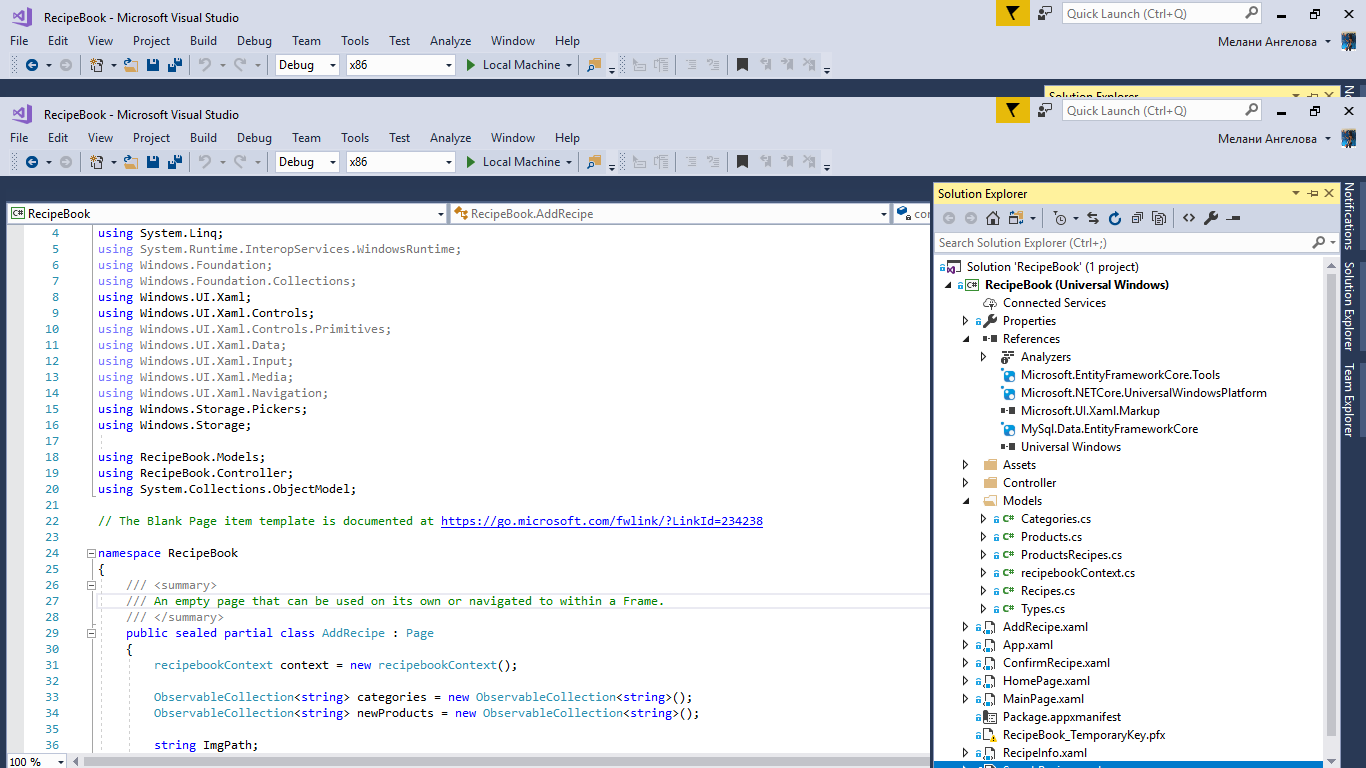
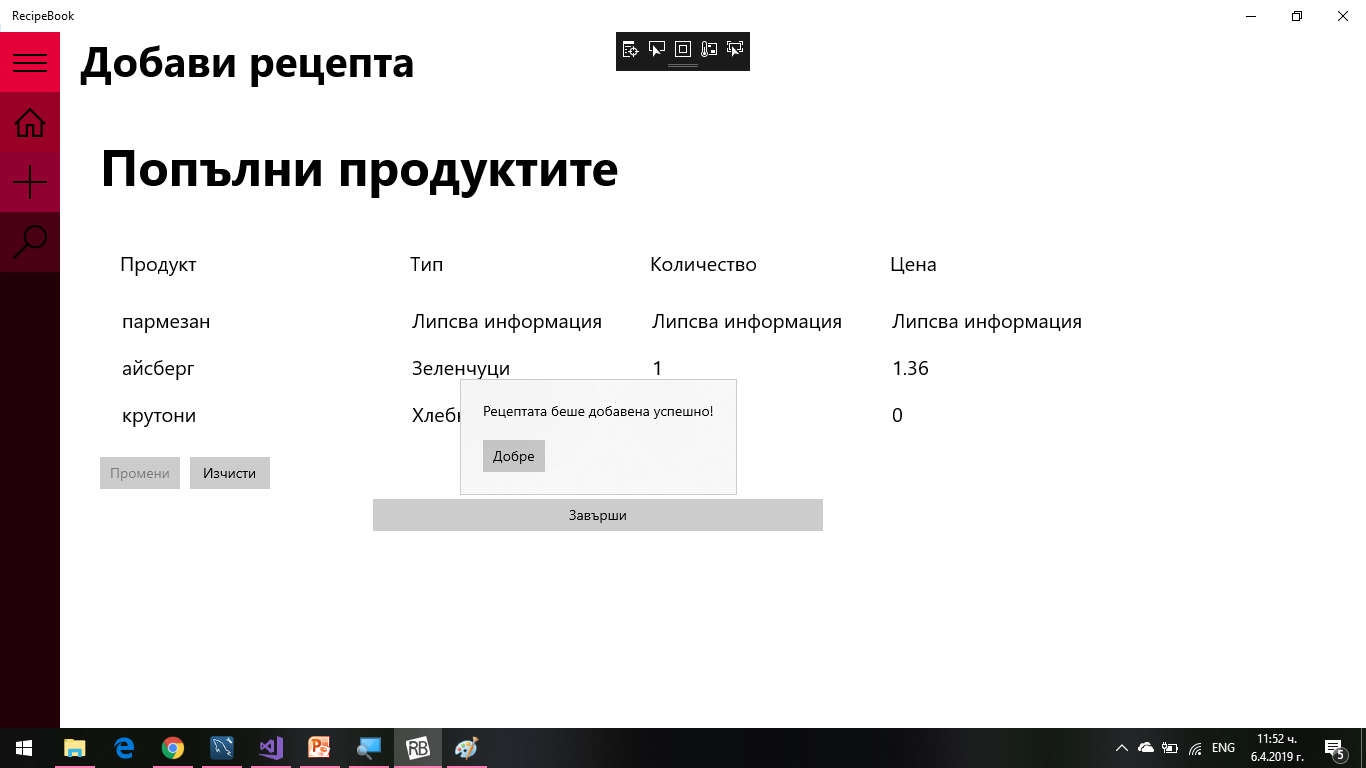
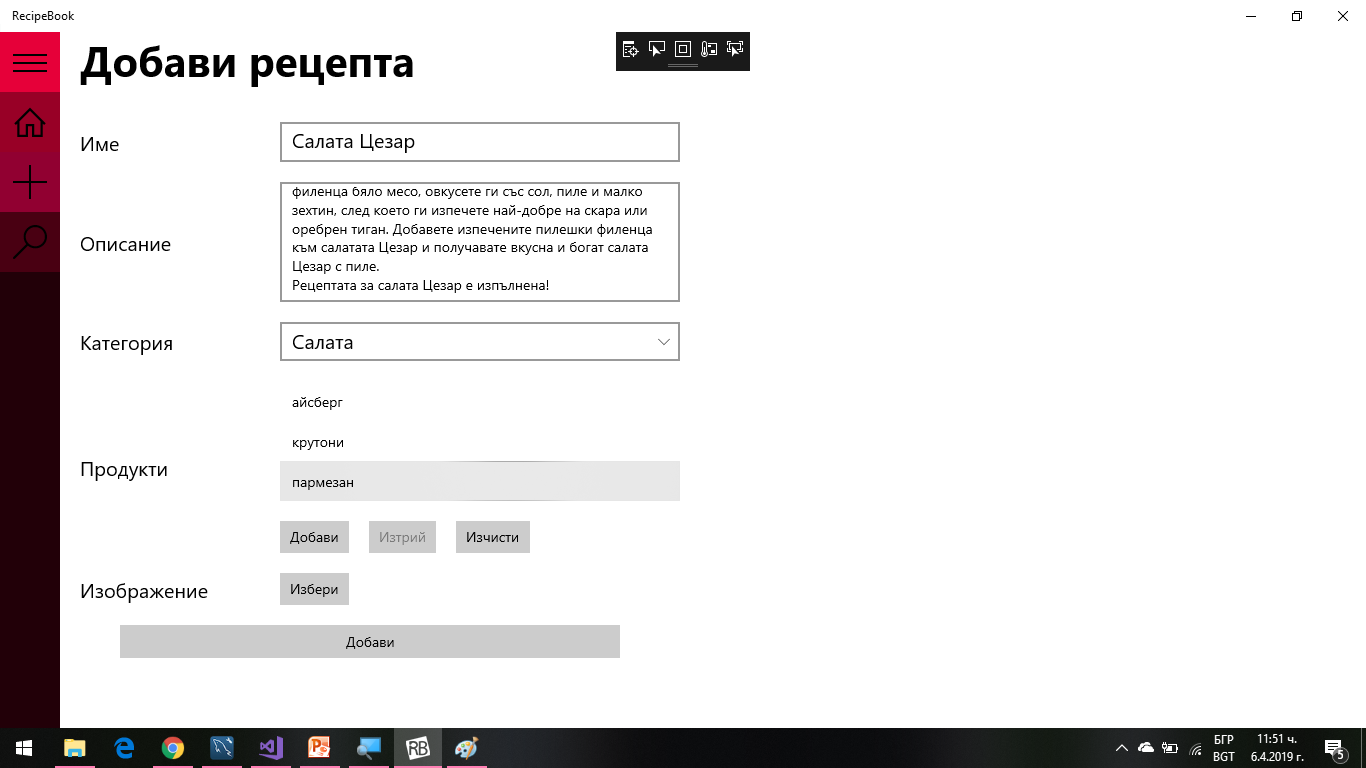
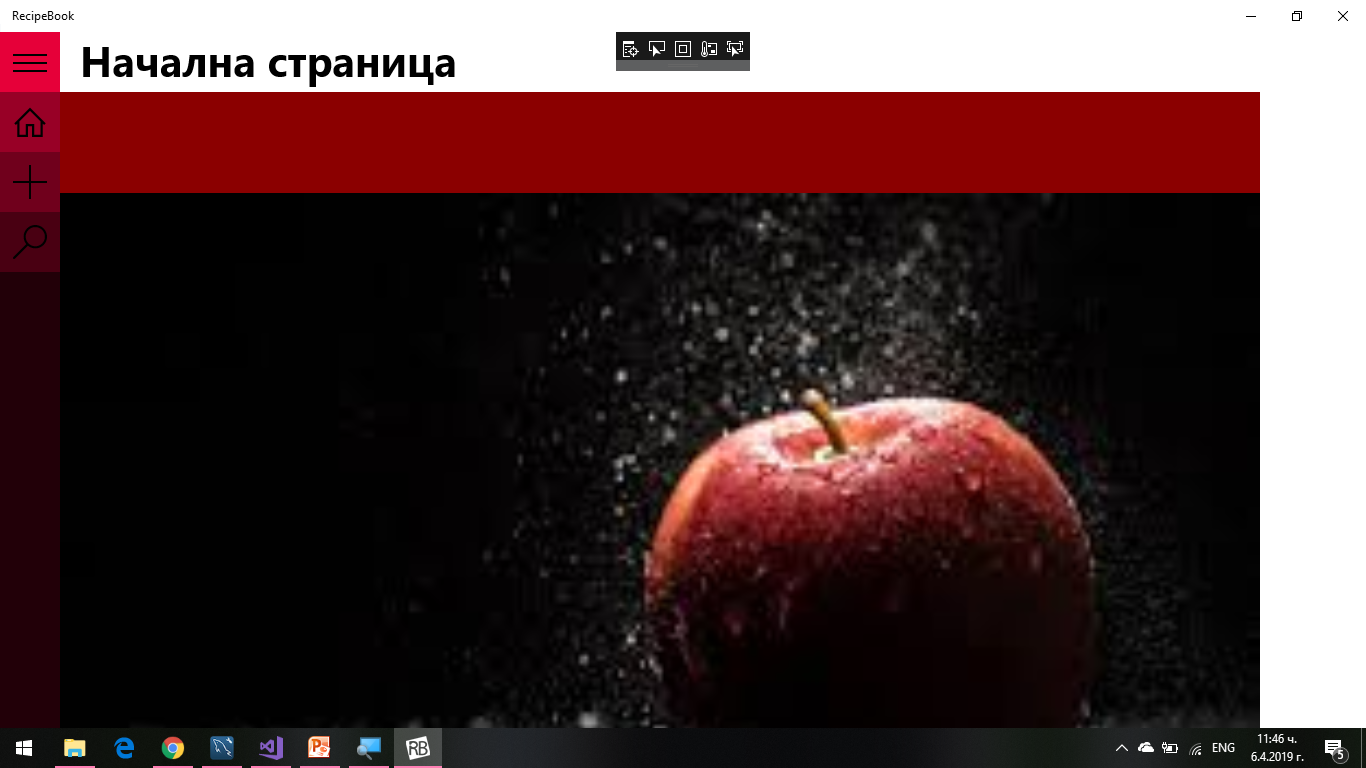
**VIEW**

* Дизайнът на приложението е направен с **UWP - Universal Windows Platform  и XAML** - език за front-end.
* Приложението се състои от няколко екрана
* Главния екран се състои от изображение и вертикално навигационно меню, намиращи се в SplitView
* При промяна на избора Frame, разположен в SplitView.Content зарежда търсената страница
* Menu\_Button се намира в RelativePanel, след натискането на бутона, SplitView.Pane се отваря и се четат имената на разделите в ListBox:
  + начален екран (**Начало**)
  + добавяне на рецепти (**Добави**)
  + търсене на рецепти по име (**Потърси**)
* Прозорецът **“Добави”** е предназначен за добавяне на лични рецепти; в него се избират името на рецептата, продуктите, нужни за приготвянето ѝ, категорията на ястието, пълно описание и изображение.
* Прозорецът **“Потърси рецепта”** е предназначен за намиране на рецепта от базата данни по име
* Прозорецът **“Попълни продуктите”** е предназначен за обогатяване/обновяване на информацията за вече добавени в данните продукти и добавянето на нови. Прозорецът е естествено продължение на **“Добави”** и се достига само през него.
* Прозорецът **“Разгледай рецепта”** е предназначен единствено за показване на избрана в **“Потърси рецепта”** рецепта.
* Дизайнът се свързва с базата данни посредством контролера. Стойностите на елементите в дизайна се получават от базата данни.

**CONTROLLER**

* Контролерът е съвкупността от методи и класове, които обработват потребителския вход и предоставят на последния достъп до базата данни.
* За възприемане на входа от потребителя се използват .cs файловете, стоящи зад XAML страниците.
* Добавянето на рецепта протича на 2 етапа:
  + Инициализиране на конкретните данни на рецептата
  + Потвърждаване на информацията за продуктите и управление на ново постъпилата информация за нови продукти
* Процесът на добавяне на рецепта преминава през следните файлове, които извличат информация съответно:
  + **CreatePage.cs** => **AddRecipe.xaml.cs**
  + **ConfirmPage.cs** => **ConfirmRecipe.cs**
  + **Controller.cs** => **ConfirmPage.cs**
* Процесът на извличане на информация от базата данни се реализира локално за всеки случай чрез **LINQ** и ламбда изрази
* Някои по-специални XAML елементи като **AutoSuggestBox**, **Flyout**, **ListView** налагаха реализирането на специфични методи, като за визуализацията на **ListView** са създадени локални класове, аналози на тези от контекста
* Определени извлечения и обновявания на информацията с базата данни налагаха валидация, за да се възприема по-добре от потребителя извлечената информация (такъв е случаят с **„Липсва информация“,** заместваща липсващи числови и текстови данни в списъка с продукти от “**Потвърди продуктите**”)

**ИЗОБРАЖЕНИЯ**



**ЗАКЛЮЧЕНИЕ**

* Проектът е в своето развитие и работата по него ще продължи
* Операциите в него ще бъдат оптимизирани
* Функционалността ще бъде разширена
* Колаборацията ще продължи

СЛИВЕН**;**

**{06.04.2019}**