**Апликација за работа во кафуле**

Апликацијата што нашата група ја направи е едноставна и лесна програма за работење во кафуле.

Со цел да се зголеми ефикасноста на конобарите оваа е едноставна програма која многу лесно се учи, така што ќе им ја олесни работата и на нив, а и управителот ќе биде среќен, бидејќи сите маси ќе бидат послужени и заведени на време.

**Упатство за користење и функционалности:**

Прво вклучување на програмата:

* При прво вклучување на програмата се вклучува прозорец за регистрација на администраторот, после оваа регистрација администраторот не може да ги смени ни корисничкото име ни лозинката. Додека за конобар корисничкото име и лозинка се статички.

Нов конобар:

* Нов конобар се додава со најава како администратор каде што администраторот има привилегии да додаде конобар сo уникатна шифра.

Најава на конобар:

* Секој конобар се најавува на апликацијата најпрво со корисничкото име и лозинката за конобари, а потоа за внес на маси и артикли со доделената шифра од страна на администраторот.

Заведување на маси:

* Секој конобар може да додаде нова маса при клик на копчето “Add“ при што се појавува прозорец преку кој се доделува име на масата. Откако ќе се додаде масата, конобарот може да внесува ставки(нарачки) кој се прикажуваат на отворениот прозорец.

Принтање нарачка/сметка:

* При клик на копчето – “Order/Bill“ за селектираната маса се печати нарачка или сметка, преку принтерот кој се наоѓа во шанк. При клик на копчето Bill масата се пребришува од листата со маси и се зачувува во дневниот извештај.

Печатење на дневен извештај:

* При клик на копчето “Daily report“ доколку сме најавени како администратор(само во тој случај е дозволено копчето) се печати дневниот извештај за сите конобари за наплатените маси во одредениот временски период.

Останати функционалности за администраторот:

* Додавање и бришење на нов артикл – При клик на копчето “Add new article” се отвара прозорец во кој треба да се внесе име на артиклот, цена и шифра. Додека при клик на “View/Delete article” се отвара нов прозорец каде е прикажана целосната листа на артикли во која може да се селиктира одреден артикл и да се пребрише. Во оваа листа артиклите се сортирани според шифра.
* “Delete waiter” – е копче преку кое се отвара нов прозорец со листа од заведените конобари, и при клик на одреден конобар може да го селектираме и пребиршеме.
* “All waiters” – е копче преку кое може да ги видеме сите моментални конобари додадени од страна на администраторот и при селекција на некој од конобарите со клик на копчето “View waiter” во отворениот прозорец ќе ни се прикажат сите негови моментални маси.

Graphical user interface

Description automatically generatedGraphical user interface, text, application, email

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, text, application, email

Description automatically generatedText

Description automatically generated

**Податочни структури:**

Главните податоци за апликацијата се чуваат во класите public class SavedData (Article/Tables/DailyReport), додека пак главните функции се чуваат во класата public class Main.

Класата Main е главниот прозорец преку кој се вршат сите функции:

Graphical user interface

Description automatically generated

**Серијализација:**

За некои податоци да бидат достапни и по исклучување на програмата искористена е бинарна серијализација. Кога корисникот прв пат ја вклучува апликацијата на својот компјутер се појавува прозорец за регистрација на администраторот, после таа регистрација се креира скриен документ во самиот директориум на апликацијата. Тој документ содржи корисничко име и хеширана лозинка за администраторот, корисничко име и лозинка за конобарите кое се креира статички. При додавање на нов конобар од страна на администраторот тој конобар исто така се додава во серијализацијата. На ист принцип е изведено зачувувањето на артиклите, масите како и дневниот извештај. За еден објект да можеме да го серијализираме, потребно е класата од која е инстанциран да биде серијализабилна.

[Serializable]   
public class Admin  
[Serializable]  
public class Waiter

За читање на веќе внесените податоци во овие фајлови се повикува методот Deserialize(): со FileStream од содржината на серијализираниот документ како аргумент. Излезот од овој метод се кастира како соодветен објект и се доделува на веќе инстанциран null објект од иста класа.  
За претставување на податоците искористени се ListBox и dataGridView.

**Опис на класата AddTableArticle**

Оваа класа најпрво содржи променлива од типот ArticleItem како и конструктор преку кои се иницијализира формата и објектот од ArticleItem се иницијализира на нов објект.  
- Функцијата “txtName\_TextChanged“ е функција која при пребрарување во textbox полето ја исполнува листата со елементи кои се повржани со тоа внес, во оваа функција се проверува дали полето е празно или не, потоа се вчитува серијализираната дата за артиклите, се проверува дали таа постои, во случај да нема и потоа се пребарува за секој елемент дали неогвото име содржи дел од внесениот текст. Ако во случај нема внесено ништо листата останува празна.  
- Потоа функцијата за додавање на тој артикл “btnAdd\_Click“, во оваа функција најпрво се проверува дали е селектиран некој артикл од листата на пребарани артикли, доколку е селектиран тој артикл и неговата количина се доделуваат на променливата за да може да се пристапи до неа преку главната форма.  
- Последна функција е “btnBack\_Click“ и таа го враќа корисникот назад на претходната(главната) форма.

Graphical user interface, application, Word

Description automatically generatedText

Description automatically generated