# Bevezető

A záródolgozatom fő témája a szoftverfejlesztés. Azért ragaszkodom ehhez, mivel szeretek programozni és nagyon tetszik a C# nyelve, felépítése. Viszonylag könnyű vele programokat készíteni, illetve szinte bármilyen elképzelést meg lehet vele valósítani. Először játékfejlesztés mellett döntöttem Unity programban, majd ezt az ötletet elvetettem, mivel attól tartottam, hogy nem készülök el határidőre, mivel beláttam, hogy rengeteg plusz felkészülést igényel. Elkezdtem tanulmányozni és úgy éreztem, hogy néhány hónap nem lenne elég, hogy átlássam a működését. Ezért esett a választásom az önkiszolgáló programra, amit azért tartottam jobb ötletnek, mert ezt a mindennapokban is használjuk, akár Tesco-ban, akár Auchan-ben.

Az önkiszolgálós pultok lényege összességében, hogy a vásárló nem a kasszához megy és futószalagon egy ott dolgozó ember húzza le a megvásárolt termékeket, hanem magunknak kell kiválasztani és lehúzni a termékeket. Ami vonalkód nélküli termék azt az érintőképernyőn lehet kiválasztani, és ha nem darab a mértékegysége, akkor a pult leméri nekünk, ha a mérlegre rakjuk. Ha ezeket elvégeztük, akkor a pult kéri tőlünk, hogy rakjuk az árucikkeket a csomagolóhelyre. Ott úgy ellenőrzi, hogy minden odaraktunk-e, hogy egy mérleg összehasonlítja a termék súlyát a már bevitt súllyal. Ha megegyezik, akkor pakolhatunk tovább.

Ezzel a programmal az a célom, hogy egyszerűbbé, letisztultabbá és egyben korszerűbb megjelenítést valósítsak meg neki a Tescoshoz képest. Modernebbé szeretném megvalósítani, hogy akár telefonnal is be lehessen olvasni a terméket, ha az a felhasználónak kézre esőbb. Tervem, hogy kibővítsem új funkciókkal, amely szintén könnyebb felhasználást jelentene a felhasználónak. Mindenféleképp teljesértékű program lenne, amelyet pár hiányosság kivételével fel lehetne használni boltokban.

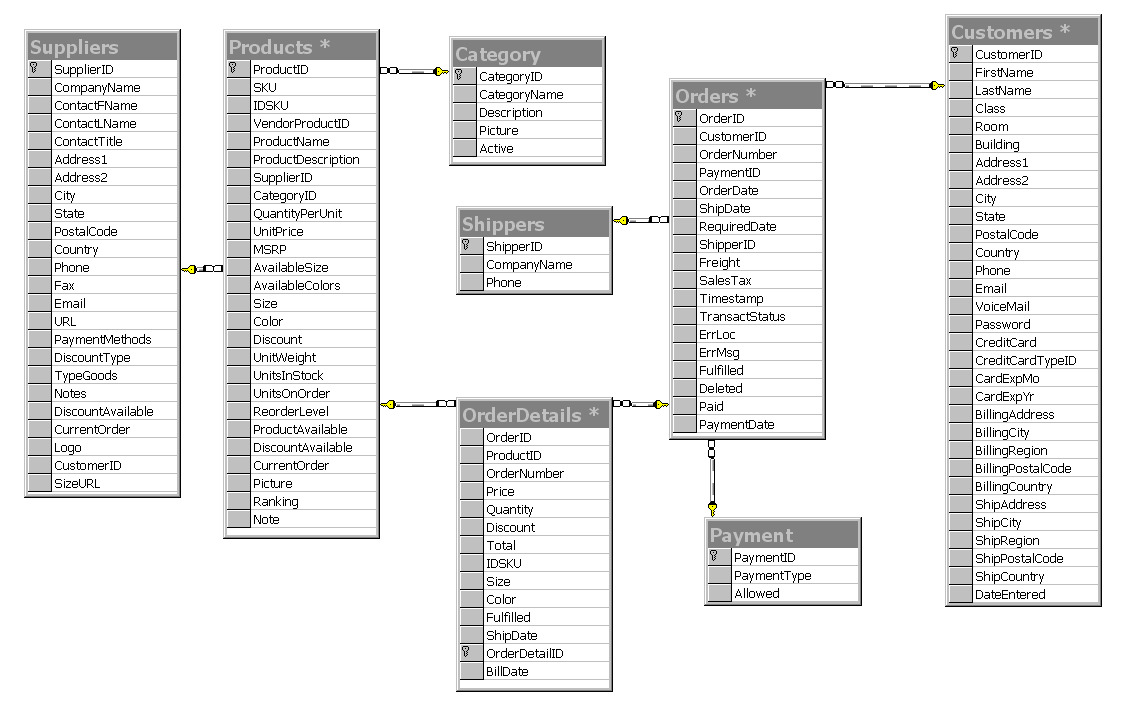
# Tervezés folyamata

Eleinte a [www.figma.com](http://www.figma.com) oldalon megterveztem a programot a Tanár úr javaslatára. Nagyon tetszett az oldal, mivel elképesztően egyszerűen meg tudtam valósítani azt, amit elképzeltem. A főoldalát meg is terveztem a programomnak, majd hozzá kezdtem a megvalósításához, Visual Studio-ban. C# form-hoz folyamodtam, mivel ez nekem egyszerűbb, mint a WPF. De mielőtt nekikezdtem volna, a dizájnnal kapcsolatban azonnal felmerült két dolog. Az egyik, hogy hogyan tudnám reszponzívvá tenni, a másik, hogy alapvető animációkat bele tudnék-e tenni. Youtube-on utána is néztem, majd egy számomra új form-os eszközzel ismerkedtem meg, a TableLayoutPanel-lal. Valójában ugyan az, mint a WPF-es StackPanel, viszont itt egyszerűbb, mivel csak be kell húzni, és anchor-ral ki tudtam választani, hogy melyik oldalhoz „tapadjon”, amellyel reszponzív is lett a form-om. Ezután a gombokat, label-eket és a képeket már csak bele kellett rakosgatnom és meg kellett formáznom. Ezt követően hirtelen megtoppantam, mivel egy TableLayoutPanel-be be akartam húzni egy másikat, aminek másmilyen hátteret adtam, majd a szélei beljebb voltak, nem töltötték ki a mögötte lévő TableLayoutPanel-t. Azt hittem, hogy tényleg WPF-hez kell folyamodnom, mert nem jött össze a dizájn, de eszembe jutott, hogy nem-e a margó miatt van. Megtaláltam, az éréküket 0-ra állítottam, így ez az akadály megoldódott. Amint behelyeztem mindent, mint ahogy megterveztem a Figma oldalán, nem tetszett a program kinézete, mivel nem tudtam árnyékokat és lekerekítéseket alkalmazni. Utána néztem mivel lehetne ezt megoldani, a https://stackoverflow.com/ oldalán találtam egy megoldást, amellyel a System.Drawing-al meg lehetett valósítani, viszont nagyon pixeles lett a lekerekítés, így ezek után az árnyékkal nem is próbálkoztam, inkább kerestem egy alternatívát. Egy kis keresgélés után találtam egy nagyon szuper kiegészítőt Form-os alkalmazásokhoz, ennek a neve Guna.UI2. Hozzá is csatoltam a programomhoz, majd a Toolbox-ban rengeteg eszközét meg is találtam. A gombokat és a paneleket ki is cseréltem, ahol tudtam, mivel rendelkeznek lekerekítéssel és a kiegészítőnek van egy „ShadowPanel” nevezetű panelje, amely már alapértelmezetten rendelkezik árnyékkal. Amint beleraktam őket a form-ba és beállítottam őket kisebb ügyeskedéssel, majdnem hogy úgy is nézett ki, mint amilyenre megterveztem. Már csak a betűtípus nem volt egyforma, így a Figma-n beállított betűtípust letöltöttem <https://fonts.google.com/> oldaláról. A „Segítségkérés egy munkatársunktól” gombra csináltam egy egyszerű form-ot, amely a gomb kattintásakor jelenik meg. Ezek után létrehoztam egy másik formot is a Dolgozói felületet, amely azt fogja utánozni, ami a Tesco-ban is megtalálható. Az ott dolgozók látni fogják a pultokat és le fogják tudni követni, amik a pultokon megjelenik. A „Segítségkérés egy munkatársunktól” gomb is összeköttetésben lesz ezzel a form-mal, hisz ha egy vásárló megnyomná azt a gombot, ezen a felületen fog látszódni, hogy melyik pultnál kértek segítséget és ott lehet eltüntetni a felugró ablakot, amely megjelenik a vásárló pultján. Az önkiszolgálót és a dolgozói felületet úgy szerettem volna megvalósítani, hogy azok külön-külön képernyőn jelenjenek meg, így imitálná azt a működést, amely a Tescoban is megtalálható. A stackoverflow segítségével sikerült beállítanom, ha 2 monitor van rákötve a számítógépre, akkor azesetben a 2 form 2 külön monitoron jelenik meg teljes képernyőben a program elindulásakor. Ha nincs képernyő, akkor csak kis megjelenésben indulnak el egy monitoron. A segítségkérés form-ját úgy állítottam be, hogy az önkiszolgáló form-ján jelenjen meg, mint egy „pop-up” ablak. Amint megjelenik, az ablak fixen van, és nem lehet mozdítani. Ez után már csak az összefűzésük maradt hátra, hogy a Segítségkérés gomb megnyomásakor az engedélyezés gomb elérhetővé váljon a dolgozói felületen, így az engedélyezés gombra kattintva eltűnik az önkiszolgálónál az ablak és újra elérhetővé válik az önkiszolgáló kezelhetősége. Mivel mindent az Onkiszolgalo.cs-ben hoztam létre, úgy szerettem volna, hogy ez legyen a fő ablak, ami mindent működtet, így ez legyen az összeköttető is a többi 2 form közt. Számomra nagyon nehezen ment, hogy az „Engedélyezés” gomb eltüntesse a segítségkérés ablakot. Órákig jártam utána, hogy miképpen lehetne ezt az akadályt orvosolni, de sikertelenül, így a Tanár úrhoz fordultam segítségül, aki sikeresen megfejtette az akadályomat. Egy hatalmas segítség volt ez számomra, mivel egy olyan dolgot mutatott meg, amelyeket később is fel tudok használni.

Amint kész lett a formok megjelenése és működése, a 2 felül található, magyar és angol gombokat állítottam be, hogy kattintásukra megváltozzon a felületek nyelve. Itt egy szimpla fordítást kellett elvégezni, hogy a Click eseményre megváltozzon a label-ek és gomb-ok felirata vagy magyarra, vagy angolra.

Ezek után beállítottam az időt. Egy szimpla Timer-rel oldottam meg, majd a label szövegjét beállítottam DateTime-al, hogy a jelenlegi időt mutassa a Timer Tick eseményénél. Beállítottam azt is, hogy a magyar nyelven magyar formátumú, angol nyelven pedig angol formátumú legyen az idő. Ezzel el is készültem az alapmegjelenítéssel.

A következő lépés az adatbázis elkészítése MySql-el, amelyhez találtam egy sablont, hogy miképp készítsem el az adatbázisomat.



A képen látható adatbázis szerkezetből kinyertem azokat, amelyeket az én programomban is fel tudom használni. Az adatbázisomat „termekek” címen hoztam létre, majd „arucikkek” néven létrehoztam egy táblát, melyben tárolódnak: „aruid”, „vonalkod\_szam”, „nev”, „mennyiseg”, „mertekegyseg”, „ara”. A táblát fel is töltöttem egy pár termékkel, ami a szemem elé került.  
Szerencsémre akkor kezdtük el tanulmányozni a MySql összekapcsolását a Visual Studio-val, amikor ide eljutottam, így a frissen tanultakat fel is tudtam használni a programomban, így össze is kapcsoltam a programomat az adatbázissal.

Az ezt követő funkció megvalósítása a vonalkód beolvasása volt, amely egy kisebb fejtörést okozott, mivel nekem nincs vonalkódolvasóm, mint senki másnak az ismertségemben. Azon tűnődtem, hogy veszek egyet, de sajna a legolcsóbb minőségi vonalkódolvasó is drága volt ahhoz, amit a keretem megengedett volna. Így kreatívnak kellett lennem és más ötlethez kellett folyamodnom. Egy kis gondolkodás után megnéztem, hogy egy ilyen funkciót hogyan tudnék megvalósítani a telefonommal.