

Meta-Tactics

Databasteknik och webbaserade system

Projektrapport

Utförd av: Petter Näsman ,William Danielsson & Samuel Hannu Datum: 2021-03-29
Första inlämningen

Kursansvarig: Per Kvarnbrink

Övriga lärare: Stefan Berglund, Ole Norberg & Stig Byström

Godkänd

Retur

Rättad av:

Datum:

Innehåll

1. Inledning	2
2. Problembeskrivning	2
3. Systembeskrivning	3
4. Implementation	4
5. Gränssnitt	7
6. Lösningens begränsningar	15
7. Problem och reflektioner	16
8. Referenser	16

1. Inledning

Denna rapport redogör för projektarbetet Meta-Tactics inom kursen Databasteknik och webbaserade system. Projektet har gått ut på att utveckla en webbaserad databasdriven applikation och har i detta fall utförts i en grupp bestående av Petter Näsman, William Danielsson och Samuel Hannu. Arbetet har haft som syfte att omsätta de kunskaper som anskaffats under tidigare laborationer i kursen.

Meta-Tactics är en databas-baserad applikation som skapats för att underlätta för spelare av spelet Teamfight Tactics. Teamfight Tactics är ett rund-baserat strategispel som ställer åtta spelare mot varandra i en alla mot alla match. Spelet går ut på att bygga ett lag bestående av Champions som släss för spelarens räkning, för att vinna spelet gäller det att vara den sista personen kvar. Varje champion har olika Traits som kan kombineras för olika effekter till ditt lag. En champion kan även tilldelas upp till tre Items vilket gör karaktären starkare. Meta-Tactics finns till för att samla information om alla Champions och Items från spelet på ett ställe och även visa de bästa kompositionerna av lag för att ge spelare som använder applikationen en extra egg i matchen.

2. Problembeskrivning

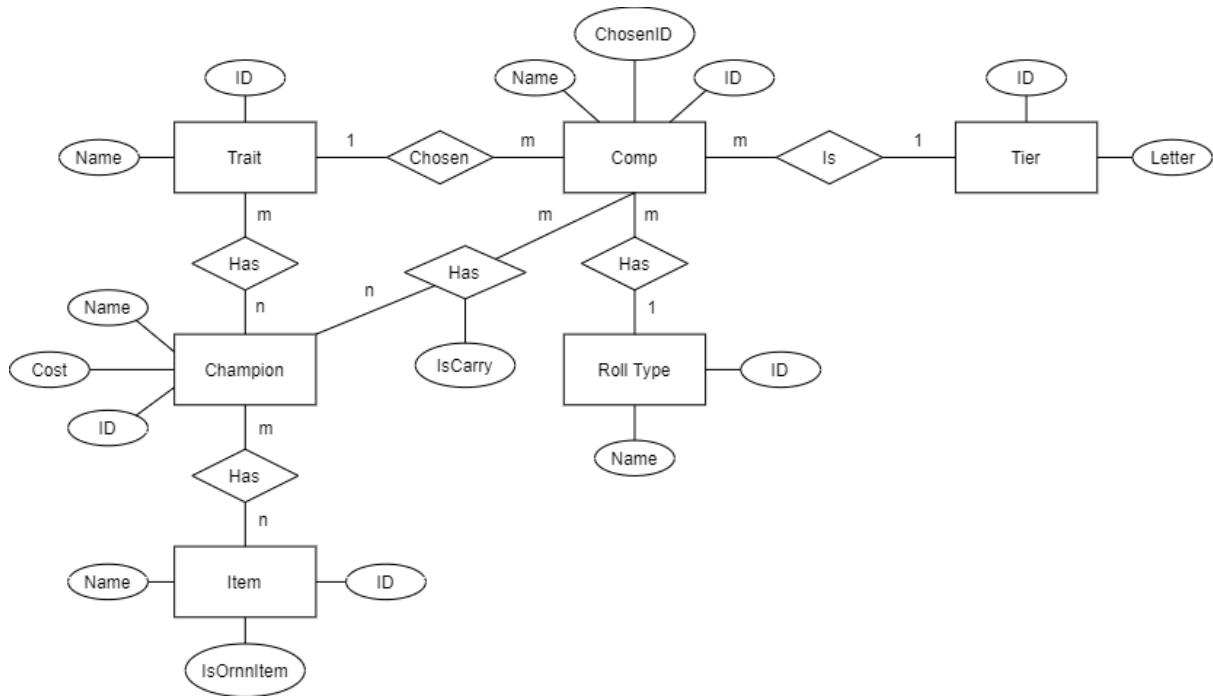
Den webbaserade databasdrivna applikationen som utvecklats under projektet har utformats utifrån ett antal krav. Dessa krav har på ett eller annat sätt applicerats och handlar främst om funktionalitet som bör finnas i applikationen.

Det databasdrivna systemet ska vara baserat på ASP.NET MVC, där användaren genom ett webbgränssnitt ska kunna hämta, ändra och lägga till data i underliggande databas. Routingen på applikationen ska vara inställd för att starta där den ska när annat inte är specificerat i URLen. Applikationen ska ha ett inloggningssystem och ska använda sessions, sessionStorage eller cookies för att spara information från en sida till en annan.

Webbgränssnittet i applikationen ska på ett eller annat sätt presentera data från databasen som om detta var unika webbsidor samt innehålla minst ett inmatningsformulär. System ska också lagra metadata, t.ex ”keywords” och ”description”.

Resulterande autogenererad html-kod ska också analyseras enligt W3C:s rekommendationer.

3. Systembeskrivning



Databasen består av 6 stycken relaterade tabeller. Varav tre stycken har en många till många kopplingstabell. I databasen finns det comps (Compositions/Teams) som består av Champions. Champions kan ses som spelpjäser som i sin tur har sina egna attribut och egenskaper som Items, Traits, Cost osv. Många Items kan tillhöra många Champions och många Champions kan ha många items, samma sak med traits.

Comps är rankade med Tier där A är högst följt av B, C, D, E, F. Ett comp har även en Roll Type, vilket är en taktik som bör appliceras när ett specifikt comp används, till exempel slowroll, hyperroll m.m. Relationerna till de olika tabellerna är ganska komplicerade då de alla hör ihop på ett eller annat sätt. Det finns även vissa specialfall som att Champions kan vara "Carry" eller att vissa speciella Items kan vara "OrnnItems" som är en viss typ av Items från en Champion vid namn Ornn. Där av "isOrnnItem" attributet i Item tabellen.

Utvecklingsmiljön som användes var Visual Studio Community 2019. Programeringsstrukturen på applikationen är ASP.net MVC Core och Google Chrome användes som standardwebbläsare för att visa hemsidan.

Databasen är en SQL-server Database och operativsystemet som användes var Windows 10.

4. Implementation

Implementationen har gjorts i Visual Studio Community med hjälp av ASP.net MVC och är skriven i C# och html. Programmet använder sig av Razor som slår ihop C# och html. Applikationen har körts på Windows 10 i Google Chrome men borde också fungera i andra webbläsare och operativsystem.

En av de första viktiga funktionerna som skapades till applikationen var ett sätt att hantera inläsning och uppvisning av sammansatt information mellan två tabeller med en många till många relation.

Detta visade sig vara mindre smidigt än önskat så som visat i bilden nedan är det svårt att göra detta på ett bra sätt direkt i en SQL sats. I detta fall ska en Champion visas upp samt dennes Traits, Items, och Cost. Då en Champion har flera Traits och Items visas blir dessa uppvisade genom att repeteras för varje "kombination" som finns i deras relationer. I detta fall var önskan att lyckas visa all information för en Champion på en rad med fler kolumner istället för denna repetition.

	Trait	Champion	Item	Cost
1	Assassin	Akali	Blue buff	3
2	Assassin	Akali	Infinity edge	3
3	Assassin	Akali	Rapid firecannon	3
4	Ninja	Akali	Blue buff	3
5	Ninja	Akali	Infinity edge	3
6	Ninja	Akali	Rapid firecannon	3
7	Mage	Annie	Gargoyle stoneplate	2
8	Mage	Annie	Ionic spark	2
9	Mage	Annie	Surfire cape	2
10	Fortune	Annie	Gargoyle stoneplate	2
11	Fortune	Annie	Ionic spark	2

Exempel på tabell från SQL-sats

Till en början experimenteras det med olika SQL-satser för att lösa problemet, men detta verkade hopplöst. På grund av detta valdes det istället att skapa en algoritm i applikationens kod för att hantera repetitionen av samma data. Ett objekt kallat ChampionInformation skapades där informationen om en Champion kan sparas in, vilket inkluderar namn, Trait 1-3, Item 1-3, Cost och Ability, men i detta fall är ej de två sistnämnda relevant. När tabellen sedan ska läsas in skapas ett temporärt sträng objekt där Champion namnet på första raden sparas in. Detta kunde sedan användas för att lägga in ett villkor i en while-loop som kollar om namnet som läses in på varje rad är detsamma som föregående rad. Om detta är fallet kommer samma ChampionInformation fortsättas användas för att lägga in ny data i form av Traits och Items. Genom if/else-satser säkerställs sedan att Traits 1-3 samt Items 1-3 aldrig fylls i med dubbletter och att de fylls i rätt ordning. När while-loopen märker att ett nytt Champion-namn dyker upp läggs ChampionInformation objektet till i ett ChampionInformationList objekt för att sedan skapa ett nytt ChampionInformation objekt och börja loopen på nytt. Detta gör att varje Champion

Projektrapport

separerar från varandra och innehåller ingen repeterande data, vilket gör det väldigt enkelt att sedan visa upp i vyn. Värt att notera är att olika Champions kan ha olika många traits, vissa har två medan andra har tre. Detta har hanterats genom att Trait3 är null om inget tredje Trait hittas, vilket sedan kan användas i vyn för att skapa ett villkor som endast visar Trait3 om denne inte är null. Nedan visas koden för denna implementering samt resulterande lista.

```

while (i < count)
{
    ChampionInformation ci = new ChampionInformation();

    while (i < count && (temp == null || temp.Equals(myDataSet.Tables["champions"].Rows[i]["Champion"].ToString())))
    {

        ci.Name = myDataSet.Tables["champions"].Rows[i]["Champion"].ToString();
        ci.Cost = Convert.ToInt32(myDataSet.Tables["champions"].Rows[i]["Cost"]);

        if (ci.Trait1 == null)
        {
            ci.Trait1 = myDataSet.Tables["champions"].Rows[i]["Trait"].ToString();
        }
        else if (ci.Trait1 != (myDataSet.Tables["champions"].Rows[i]["Trait"].ToString()) && ci.Trait2 == null)
        {
            ci.Trait2 = myDataSet.Tables["champions"].Rows[i]["Trait"].ToString();
            ci.Trait3 = null;
        }
        else if(ci.Trait2 != myDataSet.Tables["champions"].Rows[i]["Trait"].ToString() && ci.Trait1 != myDataSet.Tables["champions"].Rows[i]["Trait"].ToString())
        {
            ci.Trait3 = myDataSet.Tables["champions"].Rows[i]["Trait"].ToString();
        }

        if (ci.Items1 == null)
        {
            ci.Items1 = myDataSet.Tables["champions"].Rows[i]["Item"].ToString();
        }
        else if(ci.Items2 != myDataSet.Tables["champions"].Rows[i]["Item"].ToString() && ci.Items2 == null)
        {
            ci.Items2 = myDataSet.Tables["champions"].Rows[i]["Item"].ToString();
        }
        else if(ci.Items2 != myDataSet.Tables["champions"].Rows[i]["Item"].ToString() && ci.Items3 == null)
        {
            ci.Items3 = myDataSet.Tables["champions"].Rows[i]["Item"].ToString();
        }

        temp = myDataSet.Tables["champions"].Rows[i]["Champion"].ToString();

        i++;
    }
    temp = null;
    ChampionList.Add(ci);
}

```

Kod för hantering av hämtad data från databasen

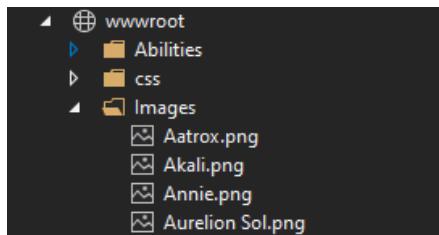
Name ↑	Cost ↑	Traits	Items		
Akali	3	Assassin Ninja	Blue buff	Infinity edge	Rapid firecannon
Annie	2	Mage Fortune	Gargoyle stoneplate	Ionic spark	Sunfire cape
Aurelion sol	4	Dragonsoul Mage	Guardian angel	Hextech gunblade	Jeweled gauntlet

Lista med ChampionInformation objekt visat i vyn

Projektrapport

När all information om en Champion sedan ska visas i vyn gjordes valet att inkludera bilder för varje Champion och Trait. Följande är en genomgång på hur visningen av listan fungerar samt hur bilderna implementeras.

Till en början laddades relevanta bilder ner och lades i mappar under rotmappen. Dessa namngavs efter respektive namn för Champion, Trait o.s.v för enkel hantering i koden.



Importerade bilder namngivna efter Champions

För att sedan visa Champions med deras medföljande data loopas ChampionInformation listan igenom och i vyn används innehållet i varje ChampionInformation för att bilda en rad. Här kommer namngivningen av bilderna till hjälp. Med tecknet ~ i src strängen kommer vi automatiskt till rotmappen och kan därifrån välja rätt mapp med bilder. För att sedan visa rätt bild till t.ex rätt Champion används namnet i en ChampionList, där nu endast strängen “.png” behövs läggas till i slutet. Denna metod används för Champions, Traits samt Items. Ett exempel som visar hur detta sker för en Champion samt resultatet av detta visas nedan.

```
@model Teamfight_Tactics.Models.DisplayLists
@foreach (Teamfight_Tactics.Models.ChampionInformation ChampionInformation in Model.ChampionInformationList)
{
    <div style="margin-top:3rem;">
        <div style="background-color: #1F1A24; opacity:0.95;">
            <h4 style="margin-left:1rem; padding-top:1rem; padding-bottom:1rem;">Champion information - @ChampionInformation.Name</h4>
        </div>
        <div style="display:grid; background-color:#1F1A24; grid-gap:0; margin-top:-0.5rem; grid-template-columns: repeat(5, 1fr);">
            <div>
                
            </div>
            <div>
                <h5 style="position:relative; top:4rem;"> @((ChampionInformation.Cost)</h5>
            </div>
        </div>
    </div>
}
```

Kod som visar en champion med tillhörande bild i vyn

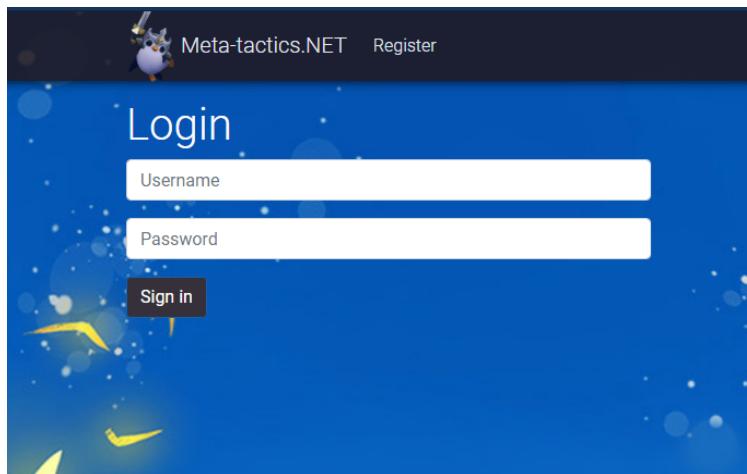
Name	Cost	Traits	Items		
Akali	3	Assassin Ninja	Blue buff	Infinity edge	Rapid firecannon
Annie	2	Mage Fortune	Gargoyle stoneplate	Ionic spark	Sunfire cape
Aurelion sol	4	Dragonsoul Mage	Guardian angel	Hextech gunblade	Jeweled gauntlet
Azir	5	Emperor Keeper	Guardian angel	Guinsoo's rageblade	Spear of shojin

Resulterande visning av Champions i vyn

5. Gränssnitt

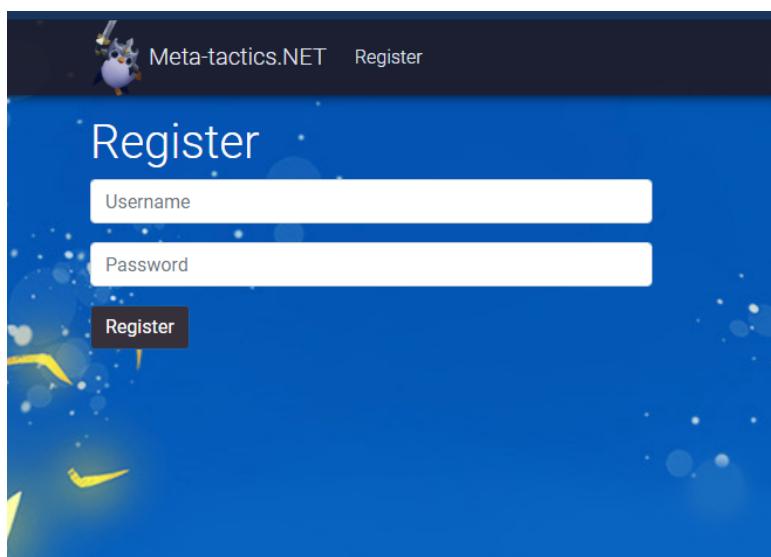
Här presenteras webbapplikationens gränssnitt och funktioner med hjälp av bilder och förklarande text. De mest triviala vyer som Champions, Items och Compositions kommer att gås igenom för att ge läsaren en känsla av webbsidan.

Det första användaren ser när hen startar upp sidan är en inloggningsskärm. För att få tillgång till sidan måste hen vara inloggad på ett konto. Ifall användaren inte har ett konto klickar hen på Register.



Startsidan för webbapplikationen

När användaren har skrivit in sitt konto klickar hen på Register knappen och användaren registreras då i databasen och det går nu att logga in med det angivna kontot.



Register vyn för att skapa ett konto

Projektrapport

Den första vyn som automatiskt visas när användaren är inloggad är Compositions, Compositions är trots allt den viktigaste delen av spelet och därför är det den som det är mest vikt på. På Compositions kan man se vilka Champions olika comps har, vilka Items som använts och vilken Rank/Tier compet har. Denna vy har även en sorteringsfunktion för att lätt kunna ordna alla Compositions från sämst till bäst, eller bäst till sämst.

Tier ↑	Name	Chosen Trait	Roll type	Comp
A	Kayle	◆ Executioner	Level 8	
A	Samira	❖ Slayer	Level 8	
A	Legendary	❖ Slayer	Level 8	
B	Keepers	❖ Keeper	Level 8	
B	Mage's	◆ Mage	Level 8	
B	Brawler's	◆ Brawler	Level 8	
C	Tristana	◆ Sharpshooter	Slowroll Level 5	

Start-vyn när användaren är inloggad, här kan hen sa alla de vanligaste compositions rankade från A - F

Projektrapport

Förutom Comps finns också Champions där alla Champions i spelet finns listade. Där kan man se hur mycket varje Champion kostar, vilka de bästa Items till en Champion är och vilka Traits dem har. På denna sida kan användaren söka på vilken champion som helst samt filtrera på Traits såsom Duelist, Hunter, Assassin osv. Alla rubriker som har en $\downarrow\uparrow$ icon bredvid sig går att sortera på. Klickar användaren på Name sorteras listan i stigande respektive fallande ordning.

Name	Cost	Traits	Items		
Aatrox	4	Vanguard Cultists	Gargoyle stoneplate	Gargoyle stoneplate	Sunfire cape
Akali	3	Assassin Ninja	Blue buff	Blue buff	Infinity edge
Annie	2	Mage Fortune	Gargoyle stoneplate	Gargoyle stoneplate	Ionic spark
Aurelion sol	4	Dragonsoul Mage	Guardian angel	Guardian angel	Hextech gunblade
Azir	5	Warlord Emperor	Guardian angel	Guardian angel	Guinsoo's rageblade
Brand	1	Mage Dragonsoul	Blue buff	Blue buff	Infinity edge
Braum	2	Vanguard Dragonsoul	Bramble vest	Bramble vest	Gargoyle stoneplate
Cho'Gath	4	Brawler Fable	Guardian angel	Guardian angel	Morellonomicon
Darius	3	Slayer Fortune	Hextech gunblade	Hextech gunblade	Ionic spark

Champions vyn där alla Champions samt information om dem är listade

Projektrapport

The screenshot shows the 'Champions' section of the Meta-tactics.NET website. At the top, there is a search bar with the word 'Assassin' selected from a dropdown menu. Below the search bar are two buttons: 'Search' and 'Filter'. The main area displays a table of champions with the following columns: Name, Cost, Traits, and Items. The items column lists three items for each champion. The champions listed are Akali, Diana, Katarina, Pyke, and Talon.

Name	Cost	Traits	Items
Akali	3	Assassin	Blue buff, Infinity edge, Rapid firecannon
Diana	1	Assassin	Infinity edge, Hextech gunblade, Titans resolve
Katarina	3	Assassin	Guardian angel, Hextech gunblade, Quicksilver
Pyke	2	Assassin	Blue buff, Morellonomicon, Spear of shojin
Talon	4	Assassin	Giant slayer, Guardian angel, Infinity edge

Champions vyn, Champions filtrerade på Assassin

The screenshot shows the 'Champions' section of the Meta-tactics.NET website. At the top, there is a search bar with the word 'Sharpshooter' selected from a dropdown menu. Below the search bar are two buttons: 'Search' and 'Filter'. The main area displays a table of champions with the following columns: Name, Cost, Traits, and Items. The items column lists three items for each champion. The champions listed are Nidalee, Samira, Sivir, Teemo, and Tristana.

Name	Cost	Traits	Items
Nidalee	1	Sharpshooter	Infinity edge, Jeweled gauntlet, Spear of shojin
Samira	5	Sharpshooter	Guardian angel, Infinity edge, Last whisper
Sivir	3	Sharpshooter	Last whisper, Quicksilver, Rapid firecannon
Teemo	2	Sharpshooter	Guinsoo's rageblade, Morellonomicon, Spear of shojin
Tristana	1	Sharpshooter	Hand of justice, Last whisper, Quicksilver

Champions vyn, Champions filtrerade på Sharpshooter

Projektrapport

Name	Cost	Traits	Items
Zilean	5	Cultists, Mystic	Blue buff, Blue buff, Guardian angel
Zed	2	Ninja, Slayer	Quicksilver, Quicksilver, Rapid firecannon
Yuumi	3	Mystic, Spirit	Blue buff, Blue buff, Spear of shojin
Yone	5	Adept, Exile	Blue buff, Blue buff, Hextech gunblade
Yasuo	1	Duelist, Exile	Infinity edge, Infinity edge, Quicksilver
Xayah	4	Executioner, Keeper	Elderwood, Hand of justice, Hand of justice, Guardian angel
Wukong	1	Vanguard, Divine	Deathblade, Deathblade, Guinsoo's

Champions vyn, Champions sorterade i fallande ordning

Name	Cost	Traits	Items
Akali	3	Assassin, Ninja	Blue buff, Infinity edge, Rapid firecannon
Annie	2	Mage, Fortune	Gargoyle stoneplate, Ionic spark, Sunfire cape
Aurelion sol	4	Dragon soul, Mage	Guardian angel, Hextech gunblade, Jeweled gauntlet
Azir	5	Emperor, Keeper	Guardian angel, Guinsoo's rageblade, Spear of shojin
Brand	1	Mage, Dragon soul	Blue buff, Infinity edge, Jeweled gauntlet
Braum	2	Vanguard, Dragon soul	Bramble vest, Gargoyle stoneplate, Sunfire cape

Champions vyn, Champions sorterade i stigande ordning

Projektrapport

Ifall man vill ha en ännu mera detaljerad vy eller ta reda på mer om en Champion kan man klicka på en specifik champion vilket tar en till dess skräddarsydda Champion Page. Denna sida är en och samma för alla champions men den är dynamisk på det vis att den information som hämtas från databasen är olika beroende på vilken Champion som man klickar på. Detta gör att det upplevs som att alla Champions har sin egen sida utan att man behöver koda 59 olika vyer i html. På denna sidan syns vilken Ability "egenskap" en Champion har samt mera detaljerade information med beskrivning på vad dess ability gör samt bilder på Traits och Items istället för bara text vilket gör det mera lättläsligt och förståeligt.

Champion information - Aurelion sol

4 Dragonsoul Mage

Active-ability

Voice of Lightning
Aurelion Sol breathes lightning in a line towards the farthest enemy, dealing magic damage and Overcharging all enemies hit for 10 seconds. If an enemy is already overcharged, Aurelion Sol consumes the effect and deals 50% bonus damage.

Best items for Aurelion sol

Guardian angel Hextech gunblade Jeweled gauntlet

Champion-information sida för Aurelian Sol

Champion information - Azir

5 Emperor Keeper

Active-ability

Warlord's Divide
Azir charges forward with a wall of soldiers, knocking back enemies hit and dealing magic damage. After their charge, the wall of soldiers remains in place in front of Azir for 3 seconds as impassable terrain. Any spearmen from the General trait will accompany Azir for the charge and position themselves to protect him afterwards.

Best items for Azir

Guardian angel Guinsoo's rageblade Spear of shojin

Champion-information sida för Azir

Projektrapport

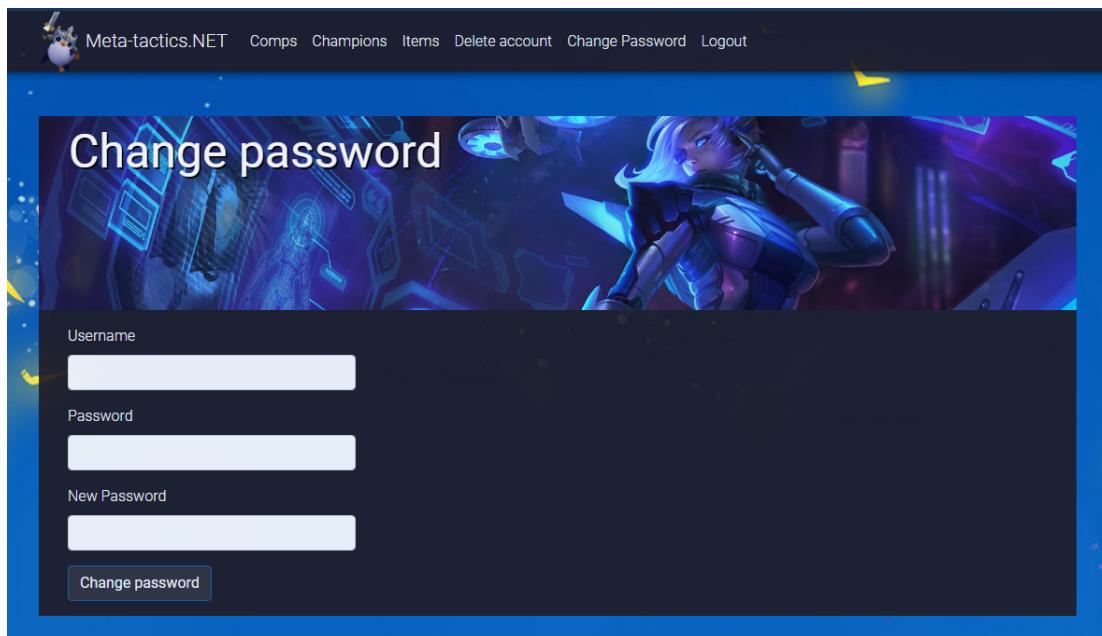
En annan vy som är viktig är Items sidan. Här kan användaren enkelt se med tillhörande bilder och text vad alla items gör och även söka på specifika Items.

Name	Description
Anima visage	The wearer regenerates 8% of their maximum Health each second.
Bloodthirster	Basic Attacks heal the holder for 40% of the damage dealt.
Blue buff	[Unique] After casting their spell, the wearer's Mana is set to 20.
Bramble vest	Negates bonus damage from incoming critical hits. On being hit by a Basic Attack, deal 80/100/150★ magic damage to all nearby enemies (once every 2.5 seconds).
Chalice of power	When combat begins, the wearer and all allies within 1 hex in the same row gain 30 Spell Power for the rest of combat.
Collector	The wearer executes enemies below 10% on hit. Executions generate 1 gold.
Deathblade	Contributing to a kill grants the holder +10 Attack Damage for the rest of combat. This effect can stack any number of times (starting at 3).
Death's defiance	60% of the damage the wearer receives is instead dealt over 5 seconds as non-lethal damage.

Items vyn, beskrivning av alla befintliga items samt en sökfunktion för specifika Items

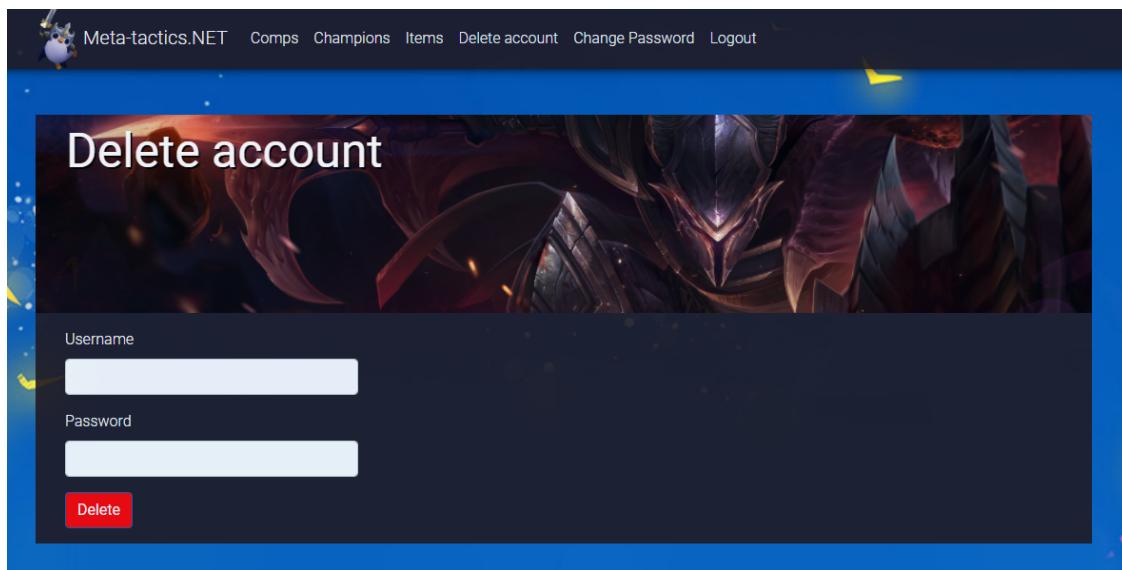
Projektrapport

När användaren är inloggad kan hen uppdatera sitt lösenord och ta bort kontot ifall så önskas. För att byta lösenord måste hen skriva in sitt lösenord och användarnamn igen som en säkerhetsåtgärd och sen skriva in sitt nya lösenord. När hen klickar på Change Password knappen ändras det befintliga lösenordet i databasen.



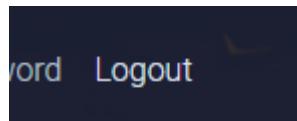
Gränssnitt för att ändra lösenord på profilsidan

På Delete Account vyn är det ungefär samma sak. Här måste också användaren skriva in sitt kontonamn och lösenord igen för att konfirmera att hen är säker på att kontot ska tas bort. När hen klickar på Delete tas användaren bort helt ur databasen och tillgången till sidan försvinner och hen kommer inte kunna logga in med den gamla informationen längre.



Gränssnitt för att ta bort kontot på profilsidan

Slutligen kan hen logga ut genom att när som helst klicka på Logout knappen längst uppe till höger. Då skickas användaren tillbaka till inloggningssidan och kan inte få tillgång till sidan något mera förens hen loggar in igen.



Logga ut knappen

6. Lösningens begränsningar

Implementation är inte helt felfri, till exempel blev några planerade funktioner och metoder uteblivna. En av dessa är att ifall man söker på något t.ex item eller en champion så försvinner sorteringen på listan ifall man hade sorterat sen innan. Vi satt länge och försökte fixa detta men tillslut fick vi ge upp då vi märkte att tiden började rinna ut.

En annan begränsning var att vi ville ha en “Comp Builder” där man själv grafiskt kan få bygga sitt eget Composition av Champions. Men detta visade sig vara för komplicerat och fungerade inte med hur vi hade strukturerat upp databasen.

7. Problem och reflektioner

Detta projektet har verkligen varit lärrikt och utmanat oss i gruppen om våra databaskunskaper. Vi har fått lösa många problem längs vägen men i slutändan överträffade vi våra förväntningar. Det var ett flertal gånger då vi var redo att avgränsa projektet ytterligare och ta bort vitala funktioner därför att det inte gick att få de att fungera. Men vi lyckades alltid att överkomma dessa problem och i samband med det lära oss mycket mer än vad vi hade gjort ifall vi tog den lätta vägen.

Ett exempel på detta är de olika Champion Information sidorna som kommer upp när man klickar in på en specifik Champion. Dessa ska vara individuellt för varje champion och inte vara hårdkodade, istället skulle all data hämtas direkt från databasen och generera en specifik sida för varje Champion. Detta var svårt på flera sätt och särskilt på grund av hur databasen var uppbyggd som gjorde det svårt att hämta rätt data ur tabellerna. SQL-koden var också klurig då SQL gärna vill visa datat på rader medan vi ville visa upp den som kolumner. För detta gjordes en workaround genom att ta in SQL datat och formatera om det i en lista som sen skickas till kontrollen och upp till vyn.

I sin helhet har projektet gått relativt bra förutom att det var brist på tid, det blev tidigt uppenbart att uppgiften aldrig skulle bli inlämnad i tid. Men vi i gruppen känner att vi har lärt oss mycket under denna kurs och projektet var ett bra sista "test" för att visa våra kunskaper inom SQL, MVC dot.net.core och Razer.

8. Referenser

Filmer från kursen:

http://www2.tfe.umu.se/systemteknik/Webbteknik/DoW/MVCintro/content_med_core.html

W3Schools:

<https://www.w3schools.com/tags/default.asp>