

Freeelo

Grupp 32

Designdokument

V. 3.0

2016-05-25



Författare:

Christoffer Nilsson

Sigvard Nilsson

Alexander Johansson

Dokumenthistorik

Version	Datum	Ändring	Av
1.0	4/4	Grov layout, överblickande innehåll	Christoffer, Sigvard
1.1	19/4	Layout- och innehållsändringar	Christoffer
2.0	11/5	Utbyte av alla bilder mot ny design, textrevidering och mer ingående förklaringar, klassdiagram uppdaterat, ordlista uppdaterad	Alexander
2.1	23/5	Byte av vissa bilder, uppdaterad text	Alexander
3.0	25/5	Förbättringar gällande layout och ett par ändringar gjordes från feedback. Use-case-diagram gjordes på dator istället för på papper. Korrekturläsning och ordlista uppdaterat	Christoffer, Alexander

Innehåll

Dokumenthistorik	2
Ordlista	4
1. Dokumentets syfte	5
2. Systemdiagram	5
3.1 Scenarion	6
4. Klassdiagram	7
5. UI-skisser	8
5.1 LoginActivity	8
5.2 MainActivity	8

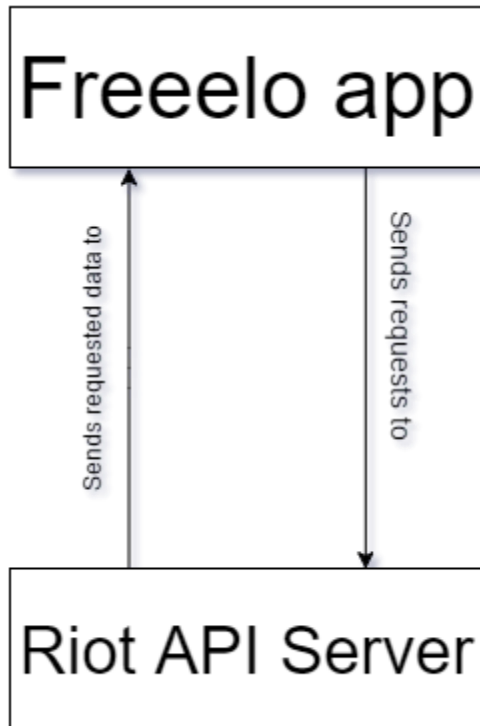
Ordlista

API	i detta sammanhang en server som tillåter tredje part att skicka specifika requests om olika information angående spelet.
Drop-down-spinner	ett element i Android som representerar en meny vars alternativ visas i en textbox med drop-down-funktionalitet.
Mastery	en av totalt trettio egenskaper i ett egenskapsträd som påverkar en spelares förmågor och egenskaper i en match.
Rank	en spelares rang
Request	en förfrågan om någon data från en server.
REST	ett begrepp som beskriver hur kommunikation mellan maskiner kan tillhandahållas.
Rune	en av totalt trettio runor som påverkar en spelares egenskaper i en match.
Summoner name	en spelares namn i spelet.
Summoner spell	en extra offensiv eller defensiv attack som alla karaktärer har.
Ultimate	varje spelare har en ultimat förmåga som är mer kraftfull än de andra, "vanliga" attackerna.
View pager	element i Android som tillåter användaren att navigera mellan olika "block" av data, exempelvis mellan fragment i en activity.

1. Dokumentets syfte

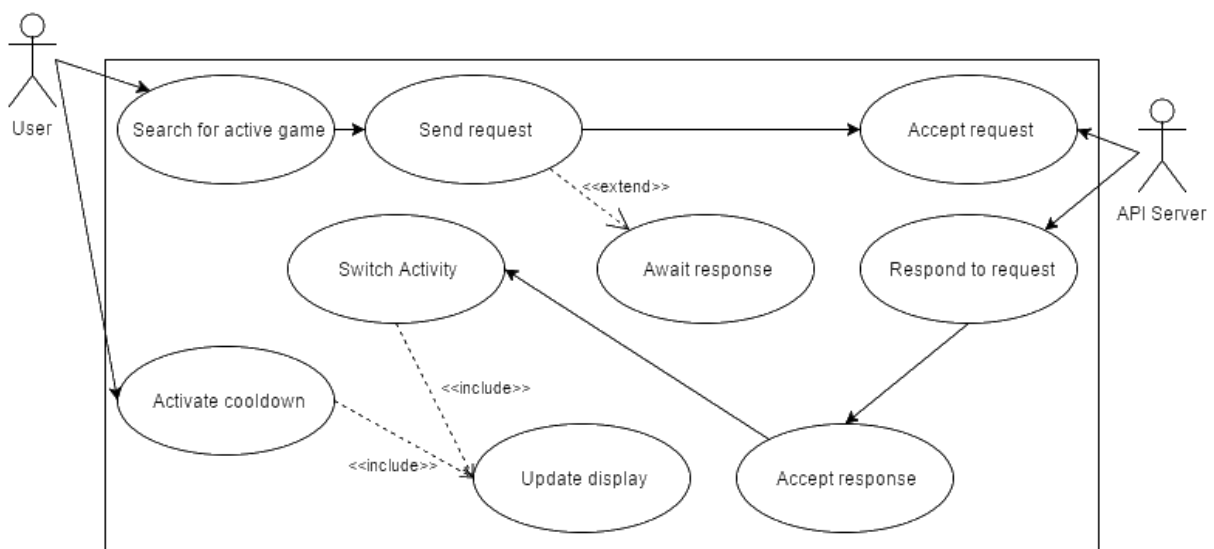
Detta dokument innehåller ett systemdiagram för överblick av systemets begränsningar och interfaces, användningsfallsdiagram (use-case-diagram) och klassdiagram för en mer ingående syn i applikationen.

2. Systemdiagram



Diagrammet till vänster beskriver hur vår Androidapplikation kommunicerar med Riot Games API-server. Appen skickar en request till servern som sedan svarar med motsvarande data (om requesten är korrekt specificerad). API-servern är baserad på RESTful arkitektur och därför måste våra requests också vara av REST-typ.

3. Användningsfallsdiagram

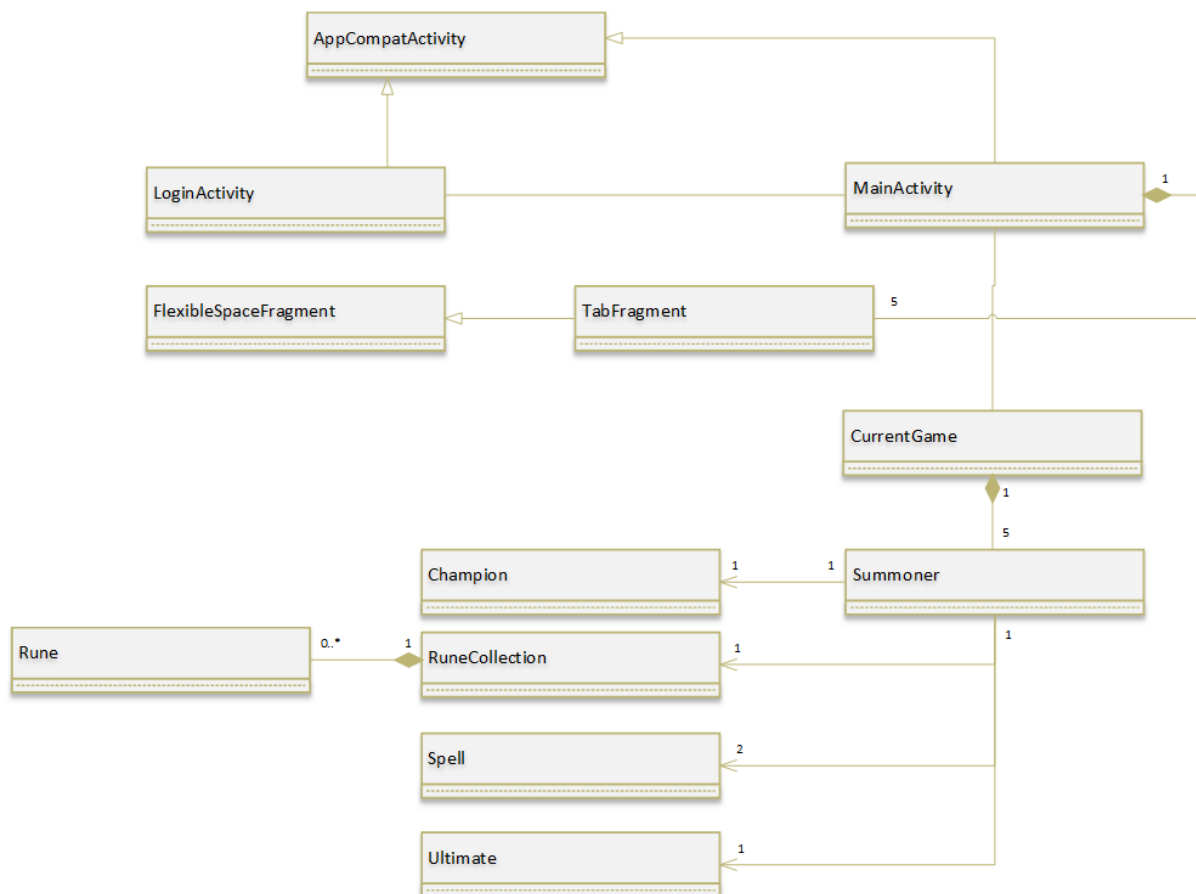


3.1 Scenarion

- *Sökning efter aktivt spel:* Christoffer skriver in sitt *summoner name* och väntar på att appen ska ladda klart. I bakgrunden skickar appen en data-request till servern och väntar därefter på svar tillbaka från servern. Servern skickar tillbaka efterfrågad data efter viss tid och appen tar emot datan. Datan är korrekt, så appen uppdaterar displayen och Christoffer kan nu se information om matchen i fråga.
- *Sökning efter aktivt spel (felstavat summoner name):* Alexander skriver in sitt *summoner name* och väntar på att appen ska ladda klart. Appen skickar en data-request till servern och väntar på svar. Servern skickar tillbaka ett error-meddelande (404) eftersom användarnamnet som skrevs in stavades fel av Alexander. Appen visar ett error-meddelande på displayen och Alexander uppmanas antingen skriva in ett annat *summoner name* eller korrigera det tidigare inskrivna.
- *Aktivering av en cooldown:* Sigvard trycker på knappen för *summoner spell* ett. Appen börjar räkna ned från viss tid, som beror på tidigare hämtad data från servern. Samtidigt uppdaterar appen displayen kontinuerligt med aktuell tid för nedräkningen.

4. Klassdiagram

- LoginActivity och MainActivity ärver båda AppCompatActivity, detta för att vi ska ha tillgång till supportbiblioteket, så appen stöds även på äldre API-nivåer (bakåt till nivå 21 från nivå 23).
- MainActivity visar nya TabFragments när det byts flik (view pager). MainActivity håller i logiken för fragment-hantering och vidarebefordran av CurrentGames data.
- TabFragment ärver från FlexibleSpaceFragment som ger upphov till effekten där bilden i MainActivity:n blir täckt av de olika flikarna när man skrollar ner. TabFragment och dess privata klasser håller i logiken för allt byte av bilder, logik för spelare och dess datarepresentation, samt även för hantering av cooldowns.
- CurrentGame hanterar all hämtning av data från Riot Games API. Med dess serialiserade underklasser returnerar den ett komplett objekt av hela aktiva spelet.

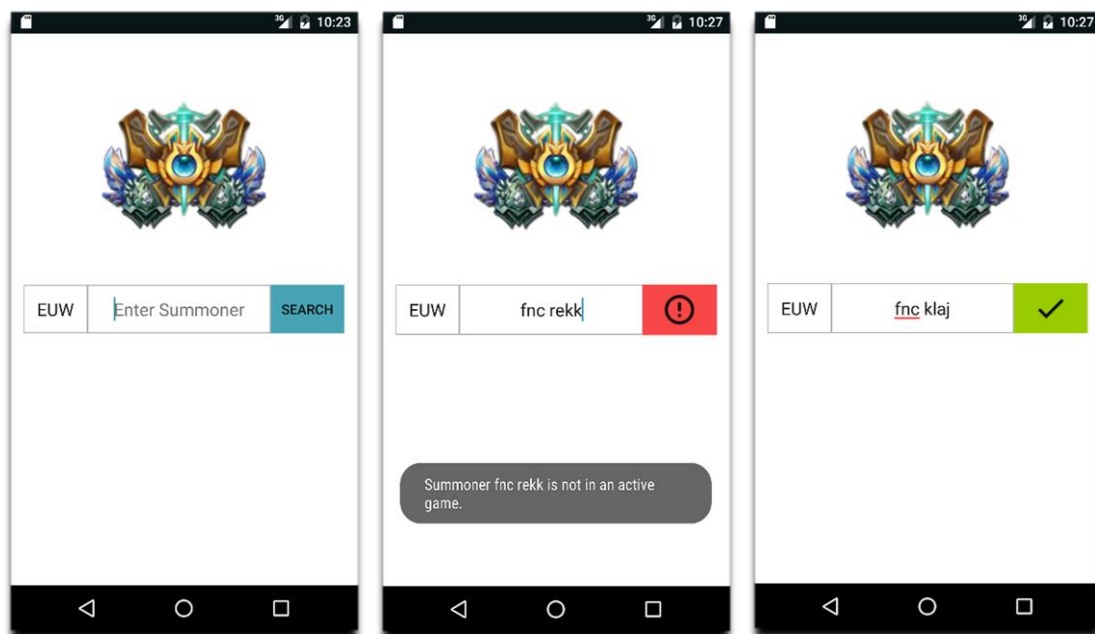


5. UI-skisser

Nedanstående bilder är UI-skisser på LoginActivity respektive MainActivity och dess TabFragment, för att ge överblickande insyn om hur färdiga produkten ser ut.

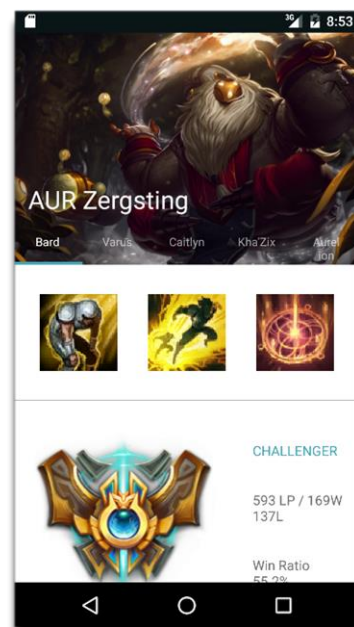
5.1 LoginActivity

LoginActivity är en enkel och stilren landningsskärm för användaren. När appen startas möts man av Freeelos logga och en simpel textruta, en drop-down-spinner för val av region och slutligen en sök-knapp. Målet var att göra appen så minimalistisk som möjligt, men fortfarande vara relevant och snygg nog för att fånga intresse. I textfältet skriver man in ett summoner name, och i drop-down-spinnern väljer man region. Efter de båda är inskrivna och valda kan man göra en sökning genom att trycka på "Search". Visuell representation följer nedan.



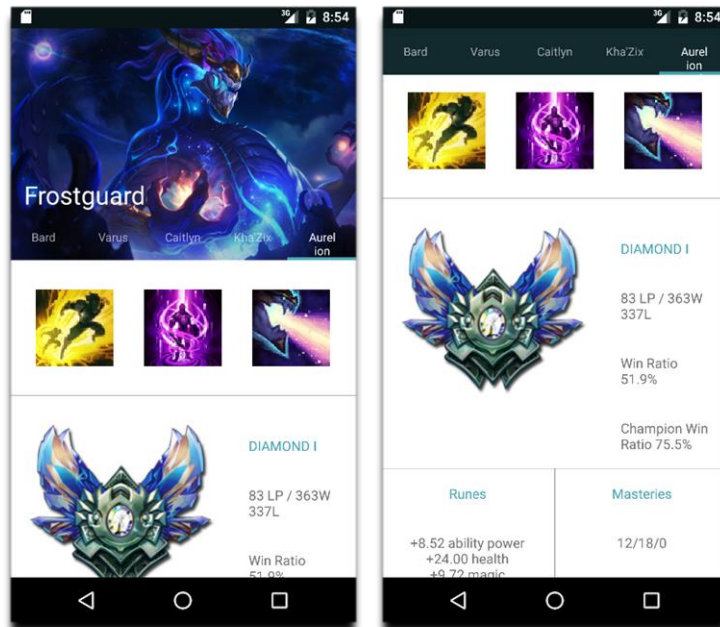
5.2 MainActivity

MainActivity och dess fragment TabFragment visar en simpel representation av alla motståndarspelare som finns med i nuvarande aktivt spel. Det finns ju endast fem intressanta spelare i varje match, nämligen motståndarna. De representeras i appen som TabFragment. När man klickar på en av ikonerna på toppen av skärmen ska MainActivity byta till motsvarande TabFragment, som visar vilka spelarens *masteries*, *runes*, *summoner spells* och *rank* är i matchen. I bilden till höger är den bruna ikonen en summoner spell vid namn "Exhaust". Den gula ikonen är en summoner spell vid namn "Flash". Den sista ikonen är en den karaktärens (Bard) ultimate, vid namn



”Tempered Fate”. Trofén i nästa stycke representerar spelarens rank och i exemplet nedan är ranken väldigt hög.

Nedan följer ännu ett exempel. Nu har användaren bytt till sista fliken. Spelaren i fråga har nu istället summoner spells vid namn ”Flash” och ”Teleport”. Karaktärens ultimate är ”Voice of Light”. Spelarens rank är lite lägre än den förra spelaren. Den andra bilden från vänster visar hur skärmen ser ut om användaren har skrollat en bit neråt.



Nästa exempel visar spelaren vars karaktär är ”Caitlyn”. Den första bilden är hur skärmen normalt sett visas. På den andra bilden är cooldowns för alla förmågorna aktiverade. Bilderna är nu alltså i gråskala med en timer som räknar ner till 0. När tiden är slut, så är karaktärens förmåga tillgänglig att använda igen.

