

Freeelo

Grupp 32

Projektplan

V. 4.0

2016-05-25



Författare:

Alexander Johansson

Christoffer Nilsson

Sigvard Nilsson

Dokumenthistorik

Version	Datum	Ändring	Av
1.0	9/3	Grov layout, överblickande innehåll.	Samtliga
1.1	10/3	Layout färdigställs, innehåll förfinas.	Samtliga
1.2	22/3	Layout och innehåll förfinas ytterligare för att passa mallen.	Samtliga
2.0	18/4	Innehåll uppdaterat	Samtliga
2.1	10/5	Liten revidering av diverse text	Alexander
3.0	18/5	Stora ändringar i text, undvikit "Vi"-form, reviderat det mesta, uppdaterat ordlista	Alexander
4.0	25/5	Förbättringar gällande layout och ett par ändringar gjordes från feedback. Korrekturläsning och ordlista uppdaterat.	Christoffer, Alexander

Innehåll

Dokumenthistorik	2
Ordlista	4
Referenslista	5
1. Dokumentets syfte	6
2. Projektöversikt	6
2.1 Projektets syfte	6
2.2 Projektets omfattning	6
2.3 Projektets mål	6
3. Produktbeskrivning	6
3.1 Målgrupp	7
4. Utvecklingsprocessen	7
4.1 Bemanning	7
4.2 Ansvarsområden	7
4.3 Planering	8
4.3.1 Aktivitetsbeskrivning	9
4.3.2 Milstolpar	9
5. Riskanalys	10

Ordlista

API	i detta sammanhang en server som tillåter tredje part att skicka specifika requests om olika typer av information angående spelet.
Champion	en spelares karaktär
Gamer	en person som spelar relativt mycket spel på fritiden.
JSON	ett format som används för att beskriva data i text
Mastery	en av totalt trettio egenskaper i ett egenskapsträd som påverkar en spelares förmågor och egenskaper i en match.
MOBA	står för <u>M</u> ultiplayer <u>O</u> nline <u>B</u> attle <u>A</u> rena – en spelgenre med två motsatta lag där var lags mål är att förstöra det andra lagets bas.
Rank	en spelares rang
Rune	en av totalt trettio runor som påverkar en spelares egenskaper i en match.
REST	ett begrepp som beskriver hur kommunikation mellan maskiner kan tillhandahållas
Summoner spell	en extra offensiv eller defensiv attack som alla karaktärer har.

Referenslista

[1] F. Tsui. O. Karam. B. Bernal., *Essentials of Software Engineering, 3rd Edition*. Jones and Bartlett Learning, 2014. Chapter 13 - figure 13.5.

1. Dokumentets syfte

Detta dokument beskriver projektets syfte, omfattning och mål. Det beskriver dessutom produkten och dess målgrupp. Utvecklingsprocessen förklaras och projektets bemanning samt varje projektmedlems övergripande ansvarsområde beskrivs. Dokumentet innehåller också en grovplanering som innehåller varje veckas aktiviteter och diverse milstolpar som bör uppnås. En riskanalys följer i slutet av dokumentet.

2. Projektöversikt

2.1 Projektets syfte

Syftet med projektet är att genom utveckling av en Androidapplikation förbättra spelupplevelsen för League of Legends-spelare. Applikationen kommer att hålla reda på fienders *summoner spells* och på så sätt kan en användare bättre anpassa användandet av sina egna förmågor för att beseгра fiendelaget.

2.2 Projektets omfattning

Projektet har begränsats till att endast hålla reda på fienders *summoner spells* och inte användarens eller användarens lagmedlemmars. Anledningen är att detta skulle innebära en fördubbling av information som ska visas i applikationen och då vi redan har väldigt mycket information skulle det anses överväldigande. Dessutom finns redan denna information i spelet och således skulle det vara onödigt att inkludera i applikationen.

Om tid och intresse finns när våra krav är tillfredsställda är det möjligt att implementera ytterligare funktioner, dock *endast* när kraven är tillfredsställda och tid finns över. Exempel på möjliga funktioner är ytterligare användning av API:et är statistik i form av grafer över procentsatsen för vinster, förluster och karaktärers *ban rate* och *pick rate*.

2.3 Projektets mål

Vårt mål är en färdig, fungerande Androidapplikation som hanterar fienders cooldowns med hjälp av Riot Games API.

3. Produktbeskrivning

Produkten är en applikation till spelet League of Legends. League of Legends är ett *MOBA*-spel med tio spelare. Varje lag har fem spelare och målet med spelet är att man ska förstöra det andra lagets bas före motståndarlaget förstör sin egen bas.

Appen ska hämta information från Riot Games API. Man skriver in sitt namn i appen så hämtas vilka spelare man möter och deras karaktärer. Den ska även hämta varje motståndares *masteries*, *runes* och deras *summoner spells* samt *rank*. Motståndarens karaktärer och attacker ska då komma upp i applikationen. Under varje karaktär kommer det finnas tre knappar, en för varje *summoner spell*. När man trycker på en knapp ska ett tidtagarur startas och när den

närmar sig bli klar (vid skiftet så knappen visas som klar) säger applikationen “[Champion] [Summoner spell] ready”, exempelvis ”Ezreal Flash ready”.

3.1 Målgrupp

Målgruppen för applikationen är inriktad på personer mellan 16-35 år som redan spelar eller ska börja spela League of Legends. De ser sig som *gamers* och tar sitt spelande på allvar eller vill börja spela seriöst. Applikationen ska ge dem en bättre spelupplevelse och fungera som ett verktyg för att få ut så mycket som möjligt av deras spelandeupplevelse.

4. Utvecklingsprocessen

Utvecklingsprocessen följer en variant av Scrum. Vid varje möte så tillbringas en kvart till att diskutera vad som har blivit gjort och vad som skall göras till nästa möte.

Varje sprint varar tre veckor och i slutet av varje sprint hålls ett retrospektmöte där alla involverade reflekterar över vad som gjorts samt ger och får feedback. Involverade parter bör även reflektera över om förhållning till processen kan förbättras. Processen är inkrementell och en “fungerande” produkt kommer alltid finnas. Tidsmässigt finns det 18 timmar per person per vecka till förfogande och totalt fyra sprintar (12 veckor totalt). I slutet av sista sprinten hålls ett sista, lite större retrospektmöte.

4.1 Bemanning

Projektet bemannas av 3 personer:

- Christoffer Nilsson
- Sigvard Nilsson
- Alexander Johansson

Kunskap om Java, det relevanta API:et, REST, JSON samt Android Studio innehas av samtliga närvarande. Övrig information eller kunskap förvärvas under projektets gång.

4.2 Ansvarsområden

Eftersom gruppen har få medlemmar har samtliga i gruppen ett mer översiktligt ansvar för utvecklingen av projektet och produkten. Enligt det sociala kontraktet som samtliga har skrivit under ska alla ska vara närvarande och deltagande i val av design, implementation och krav. Samtliga jobbar med de olika delar av produkten, men våra grova (lite mer specifika) ansvarsområden är:

- Christoffer och Alexander skriver kod för kommunikation med API-servern.
- Sigvard och Alexander skriver kod och designar GUI.
- Samtliga bidrar till att skriva alla dokument.

4.3 Planering

					DL	RS1		DL	RS2		DL	RS3		DL	RSP
		Vecka:	10	11	12	13	14	15	16	17	18	19	20	21	22
Aktivitet	Person														
Planering															
Projektplan	CN, SN, AJ														
Kravspecifikation	CN, SN, AJ														
Designspecifikation	CN, SN, AJ														
Socialt kontrakt	CN, SN, AJ														
Testdokument	CN, SN, AJ														
Implementation															
Kod + Research	CN, SN, AJ														
UI	CN, SN, AJ														
Testning															
Verifiering/validering	CN, SN, AJ														
Integrationstester	CN, SN, AJ														
Risikanalys	CN, SN, AJ														
Reflektionsrapport															
Individuell fördjupning															

Tabell 4.1. Grovplanering. Personer refereras per initialer – ex. Christoffer Nilsson, CN

Tidsplaneringen utgår från Gantt-schemat [1] ovan. Tidsplanering är utformad efter en 12-veckors period där processen är uppdelad i fyra olika sprinter. Sprinterna är tre veckor vardera, och arbetet under sprinterna redovisas vid respektive retrospektmöte.

4.3.1 Aktivitetsbeskrivning

Planeringsmomentet innefattar följande delmoment (uppskattad tid inom parentes):

Projektplan (32h) – översikt av projektet som beskriver syfte, planering, omfattning och mål med projektet. Projektplanen innehåller även övergripande beskrivningar av produkten, målgrupp och dess utvecklingsprocess.

Kravspecifikation (38h) – kravlista med funktionella och kvalitativa krav samt mer utförlig produktbeskrivning och analys av målgrupp.

Designspecifikation (32h) – dokument där produktens över- och ingripande arkitektur beskrivs, inklusive beskrivningar, diagram och skisser över gränssnittet.

Socialt kontrakt (1h) – upprättandet av ett socialt kontrakt vars underskrift behövs av samtliga gruppmedlemmar för godkännande.

Testdokument (40h) – dokument som beskriver testprocessen samt innehåller de testfall som ligger till grund för testerna.

Implementationsmomentet innefattar följande delmoment (uppskattad tid inom parentes):

Kod + Research (295h) – implementeringen i Android Studio, all relaterad kod till applikationen gällande datahantering, API-requests, m.fl, inklusive förvärvande av kunskap

UI (90h) – hur det grafiska gränssnittet implementeras.

Testningsmomentet innefattar följande delmoment (uppskattad tid inom parentes):

Verifiering/validering (53h) – tester som kontrollerar att allting fungerar som det ska, beskrivning av testprocess för projektet. Systemtestning utförs med utgångspunkt från kraven.

Integrationstester (53h) – testfall för designelementens integration i applikationen.

Riskanalysmomentet (1h) är då risker och dess sannolikhet samt konsekvens granskas.

Reflektionsrapportsmomentet (5h) är en individuell reflektion skriven i slutet av projektet. Rapporten ger möjlighet att reflektera över betydelsen av samarbete i projektgrupper.

Individuell fördjupning (20h) är det moment som varje projektmedlem väljer att fördjupa sig själv inom ett specifikt område av utvecklingen.

4.3.2 Milstolpar

RS1 – vid deadline för retrospektmöte 1 ska det finnas till förfogande en *projektplan V1*, ett *kravdokument V1* och en *applikationsdemo*.

RS2 – vid deadline för retrospektmöte 2 ska det finnas till förfogande ett *designdokument V1*, *testdokument V1*, en *applikation* med delvis uppfyllda krav samt *testrapporter*.

RS3 – vid deadline för retrospektmöte 3 ska det finnas till förfogande en *applikation V1* och prioriterade krav ska vara implementerade och testade i version 1.

RSP – vid deadline för retrospektmöte för projektet ska det finnas till förfogande en slutversion av applikationen och vissa lågprioriterade krav bör ha implementerats i slutversionen. Förutom detta ska dessutom en slutversion av projektplan, designdokument, testdokument och kravdokument finnas.

5. Riskanalys

Definitioner:

S Sannolikhet att händelsen inträffar. Skala 1 - 5

K Konsekvens om händelsen inträffar. Skala 1 - 5

RV Riskvärde (sannolikhet * konsekvens)

#	Identifierad risk	S	K	RV
R1	Sjukdom	4	2	8
R2	Ändrad planering/hantering av risker	3	2	6
R3	Verktyg (Eclipse/Android Studio) fungerar inte	2	3	6
R4	Kan uppstå problem med diverse ramverk (t.ex. Riot Games API)	1	5	5
R5	Kostnad/tidsbudget	2	3	6
R6	Integration mellan server och app	3	3	9
R7	Fixande av buggar inom rimlig tidsram	2	2	4

Tabell 5.1. Identifierade risker med sannolikhet- och konsekvensskala.

R1 - Sjukdom

Beskrivning: Vid eventuell sjukdom kan arbetet förskjutas och/eller bli okoordinerat.

Handlingsplan: Upprätta bra kontakt mellan parter, ha en öppen kommunikation ifall sjukdom skulle inträffa. Möjligtvis jobba hemifrån vid behov.

R2 - Ändrad planering/hantering av risker

Beskrivning: Vid ändring av planering/hantering av risker kan arbetet förskjutas eller förkortas.

Handlingsplan: Ha god kommunikation så att samtliga är med på planeringen/hanteringen av risker.

R3 - Verktyg (Eclipse/Android Studio) fungerar inte

Beskrivning: Vid eventuellt fel rörande mjukvara kan programmeringsförloppet förskjutas.

Handlingsplan: Använd sökmotorer för att lösa problemen.

R4 - Kan uppstå problem med diverse ramverk

Beskrivning: Problematik uppstår ifall ramverk är nere, till exempel datahämtningen från Riot Games API. Vid fel kan inte data hämtas.

Handlingsplan: Ramverket är sällan nere. Ifall ramverket skulle vara nere, fungerar inte applikationen som den ska.

R5 - Kostnad/budget

Beskrivning: Kan vara svårt att förutspå hur lång tid varje del av projektet tar för att hålla sig inom tidsramen.

Handlingsplan: Vid underskott av timmar i förhållande till budgeten så gäller det att planera in fler möten och arbetstillfällen för att ta igen förlorad tid.

R6 - Integration mellan server och app

Beskrivning: Integrationen mellan server och app kan förkomma att bli problematisk om vi inte finner smidiga lösningar på nuvarande problem.

Handlingsplan: Finna sätt att lösa problemen på, genom antingen sökmotorer eller annat inhämtande av kunskap.

R7 - Fixande av buggar inom rimlig tidsram

Beskrivning: Förskjutning på grund av bristande kommunikation eller sjukdom kan fördröja eller förhindra fixande av buggar inom sagd tidsperiod/sprint.

Handlingsplan: Försöka fixa dessa på rimligt vis, kommunicera problemen och hjälp varandra.