

# Digital akutjournal

## Testplan

Atoui, Josef      Bäckman, Viktor      Homssi, Rachel

Johansson, Tommy      Jonsson, Jesper      Lindgren, Felix

Runestam, Johan      Wijk Stranius, Simon

8 maj 2020

Version 1.0



Status

Granskad	JR	2020-05-08
Godkänd	JR	2020-05-08

## Versionshistorik

Version	Datum	Anmärkning
0.1	2020-03-06	Inlämning 2
0.2	2020-04-20	Inlämning 3, kompletteringar från inlämning 2
1.0	2020-05-08	Inlämning 4, kompletteringar från inlämning 3

## Projektidentitet

Namn	Ansvar	Telefon	E-post
Josef Atoui	Teamledare	070-776 91 16	josat799@student.liu.se
Viktor Bäckman	Systemarkitekt	072-740 02 22	vikba308@student.liu.se
Rachel Homssi	Kvalitetssamordnare	070-487 53 23	racho401@student.liu.se
Tommy Johansson	Utvecklingsledare	072-224 86 94	tomjo891@student.liu.se
Jesper Jonsson	Konfigurationsansvarig	076-131 04 43	jesjo430@student.liu.se
Felix Lindgren	Testledare	070-875 14 23	felli675@student.liu.se
Johan Runestam	Dokumentansvarig	070-252 12 85	johru036@student.liu.se
Simon Wijk Stranius	Analysansvarig	073-909 59 14	simst932@student.liu.se

## Innehåll

<b>1</b>	<b>Introduktion</b>	<b>2</b>
<b>2</b>	<b>Omfattning</b>	<b>2</b>
<b>3</b>	<b>Avgränsningar</b>	<b>2</b>
<b>4</b>	<b>Mjukvarurisker</b>	<b>2</b>
<b>5</b>	<b>Tillvägagångssätt</b>	<b>3</b>
5.1	Testnivåer . . . . .	3
5.1.1	Enhetstestning . . . . .	3
5.1.2	Integrationstestning . . . . .	3
5.1.3	End-to-end testning . . . . .	3
5.1.4	Användartestning . . . . .	3
5.1.5	Acceptanstestning . . . . .	4
5.1.6	Regressionstestning . . . . .	4
5.2	Testverktyg och testomgivning . . . . .	4
5.3	Testprocess . . . . .	4
5.4	Tidsplan . . . . .	5
5.5	Tester att utföra . . . . .	5
5.6	Möten . . . . .	5
<b>6</b>	<b>Testleveranser</b>	<b>5</b>
<b>7</b>	<b>Ansvarsområden</b>	<b>5</b>
7.1	Generella ansvar . . . . .	5
7.2	Personansvar . . . . .	6

## 1 Introduktion

Det här dokumentet innehåller testplanen för projektet VERA i kandidatkursen TDDD96. Testplanen är tänkt att användas av teammedlemmar eller andra intressenter och beskriver vilka delar av den digitala akutjournalen VERA som kommer testas och hur det ska ske. Dess mål är att se till att produkten presterar som förväntat.

Projektet kommer att bestå av sju sorters testning: enhets-, integrations/system-, end-to-end-, användar-, acceptans- och regressionstestning. Varje sorts test beskrivs i denna rapport.

Eftersom projektet, under sex månader påverkar det att (sex (6) månader) gör det att små förseningar kan förekomma och skapa problem för testplanen. Acceptanstester förväntas göras i början av maj.

## 2 Omfattning

Följande punkter beskriver de delar av systemet där fokuset på testerna kommer ligga.

- (A) Översiktsvy
- (B) Patientvy
- (C) Notissystem
- (D) Teamöversikt

## 3 Avgränsningar

Eventuella tester av dessa delar kommer inte ske avsiktligt, utan som en påföljd av andra tester.

- (A) Nätverkssäkerhet
- (B) Integration av produkten i kundens nuvarande system
- (C) Systemet kommer endast testas på Google Chrome

## 4 Mjukvarurisker

Här beskrivs de delar av systemet som kommer behöva extra testning, de är beroende på utomstående system som kan skapa oförutsedda problem.

- (A) Användning av EHRscape API:er
- (B) Synkron skrivning till olika inmatningsfält
- (C) Koppling till molntjänster

## 5 Tillvägagångssätt

I detta kapitel beskrivs metoderna för testningen mer på djupet.

### 5.1 Testnivåer

Testningen för projektet kommer bestå av enhets-, integrations-/system-, end-to-end- och acceptanstestning. På grund av projektets korta begränsad tid och personal kommer det inte finnas någon testperson utan erfarenhet med koden. Tester kommer skapas och utföras av alla medlemmar i teamet.

#### 5.1.1 Enhetstestning

Ett enhetstest kommer definieras som en funktion som inte är beroende av andra funktioner. Den som skriver koden kommer också vara ansvarig för att skriva enhetstester. Testerna skapas i Angular med modulen Jasmine som sedan exekveras genom Karma. Enhetstesterna bör endast testa funktionaliteten för komponenten den tillhör, antingen genom att testa komponentens metoder eller genom att testa rendering av komponentens HTML.

#### 5.1.2 Integrationstestning

Ett integrationstest kommer definieras som en funktion som är beroende av en eller flera funktioner. För att göra enklare integration mellan komponenter går det att använda Jasmine och Karma. På samma sätt som för enhetstester, men istället för att bara använda den egna komponenten importera andra komponenter och tjänster. För större integrationstester, där flera delar ska sammankopplas, bör det utföras tillsammans med utvecklarna av de givna delarna.

#### 5.1.3 End-to-end testning

End-to-end testning (från och med nu E2E) kommer göras genom att testa programmet i webbläsare med hjälp av verktyget Protractor. Det gör det möjligt att testa systemet på samma sätt som det kommer användas i verkligheten. Det kommer göras rullande när det finns flera delar som redan har börjat kopplats samman.

#### 5.1.4 Användartestning

Användartester kommer utföras tillsammans med kunden och tänkta målgruppen av systemet. Med antingen dator- eller pappersprototyper kommer en testanvändare att utföra ett antal förbestämda användaruppgifter. Under dessa tillfällen kommer SUS (System-Usability-Scale) mätningar göras.

### 5.1.5 Acceptanstestning

Om när acceptanstestning kommer det göras tillsammans med slutanvändare och teamet. Eftersom målet med projektet [2] är att ta fram en prototyp kommer testningen ses som färdig då kunden är nöjd med acceptanstestningen.

### 5.1.6 Regressionstestning

Under projektets gång kommer det ske regressionstestning löpande genom att skapa automatiska tester när problem uppstår. Dessa kommer då kunna köras senare för att bekräfta att tidigare problem inte återkommit.

## 5.2 Testverktyg och testomgivning

Angular tillhandahåller en testsvit för att enkelt skapa och sedan automatiskt köra enhetstester samt E2E-tester. Tester för varje komponent kommer att sparas tillsammans med källkoden, testerna blir enkla att hålla reda på och kan utföras automatiskt med ett verktyg.

- (A) All kompilering och debugging av Angular kod kommer göras med Angular CLI.
- (B) Jasmine, ett ramverk för testning i JavaScript/TypeScript. Det är efter deras standard tester utformas.
- (C) Karma, ett verktyg som används för att automatiskt köra tester skriva i Jasmine.
- (D) Protractor, ett verktyg som automatiskt startar en webbläsare och testar systemet på samma sätt en användare skulle kunna. Det bygger på också på Jasmine som ramverk för hur tester utformas.

## 5.3 Testprocess

Teammedlemmarna kommer att skriva och exekvera tester för komponenter som de har implementerat. Fokuset för testerna bör ligga på komplexa algoritmiska och numeriska funktioner, komplicerade funktioner som inte är beroende av andra komponenter/metoder. Komponenter som binder samman och använder flera olika komponenter/tjänster bör undvikas då det kan bli komplicerat och ett slöseri med tid att skapa mock-ups. Om ett testfall inte går igenom ska det åtgärdas direkt av testaren, om det inte kan fixas får ligga kvar på sin branch tills buggen har tagits bort.

Under en sprint, eller beroende på hur mycket tid teamet kan tillhandahålla, kommer utveckling göras med testdriven utveckling, istället för den mycket vanligare metoden att testa sist. Detta görs för att ge underlag till en av medlemmarnas individuella del i kandidatrapporten.

## 5.4 Tidsplan

Under de första sprintarna är det tänkt att fokus bör ligga på att framställa fungerade prototyper. Därmed har testning ett mindre fokus. När det närmar sig projektets milstolpar för de olika delarna bör testningen göras mer ingående för att därmed säkerställa en kvalitetsnivå, för systemets delar. Se Tabell 1 för preliminära datum när tester av systemets delar ska vara godkända av teamet.

Tabell 1: Testmilstolpar

Nr	Beskrivning	Datum
1	Testning av patientvy godkänd	2020-04-06
2	Testning av enhetsöversikt godkänd	2020-04-20
3	Testning av teamvy godkänd	2020-04-27

## 5.5 Tester att utföra

De tester som är bestämda i förväg finns beskrivna i Testrapporten [3]. De är kopplade direkt till de testbara kraven i Kravspecifikationen. [1]

## 5.6 Möten

Testmöten kommer ske i samband med sprintplanering. Då diskuteras hur testning har gått och vad som behöver testas inför nästa sprint.

# 6 Testleveranser

Kunden har inte specificerat några krav på tester utom att de verktyg och automatiska testsviter som skapas till produkten ska följa med leveransen. Som en del av kursen ska en testplan samt en testrapport lämnas in.

# 7 Ansvarsområden

Här beskrivs vilket ansvar för tester som finns för teamet.

## 7.1 Generella ansvar

Samtliga teammedlemmar kommer vara ansvariga för att skriva tester för sina egna delar av systemet, för att bestämma vad som ska testas får eget omdöme användas.

E2E-tester kommer delas upp efter de olika modulerna, teamet kommer bestämma vilka som skriver tester för olika moduler.

## 7.2 Personansvar

Testledaren kommer vara ansvarig för att hålla i utbildning på hur man skriver tester i Angular samt hur testplanen ska implementeras i projektet.

## Referenser

- [1] *Kravspecifikation*. Grupp 6. 2020.
- [2] *Projektplan*. Grupp 6. 2020.
- [3] *Testrapport*. Grupp 6. 2020.