

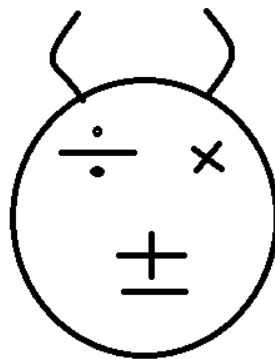


# Testplan

PUM09

2024-03-08

Version 2.0



Status

Granskad		
Godkänd		



## Projektidentitet

Grupp E-post: [emiho191@student.liu.se](mailto:emiho191@student.liu.se)

Hemsida: <http://www.liu.se/grouppage>

Beställare: Kristian Sandahl, Linköpings universitet  
Tfn: 013-28 19 57  
E-post: [kristian.sandahl@liu.se](mailto:kristian.sandahl@liu.se)

Kund: Jörgen Blomvall, Linköpings universitet  
Tfn: 013-28 14 06  
E-post: [jorgen.blomvall@liu.se](mailto:jorgen.blomvall@liu.se)

Handledare: Lena Buffoni  
Tfn: 013-28 40 46  
E-post: [lena.buffoni@liu.se](mailto:lena.buffoni@liu.se)

Kursansvarig: Kristian Sandahl, Linköpings universitet  
Tfn: 013-28 19 57  
E-post: [kristian.sandahl@liu.se](mailto:kristian.sandahl@liu.se)

## Projektdeltagare

Namn	Roll	E-post
Mabest Amin (MA)	Arkitekt	<a href="mailto:mabam091@student.liu.se">mabam091@student.liu.se</a>
Odin Dahlström (OD)	Utvecklingsledare	<a href="mailto:odida723@student.liu.se">odida723@student.liu.se</a>
Emil Holmstedt (EH)	Teamledare	<a href="mailto:emiho191@student.liu.se">emiho191@student.liu.se</a>
Martin Hultgren (MH)	Kvalitetssamordnare, Databasansvarig	<a href="mailto:marhu242@student.liu.se">marhu242@student.liu.se</a>
Casper Erik Nerf Kanefall (CK)	Testledare	<a href="mailto:casne582@student.liu.se">casne582@student.liu.se</a>
Eric Van Nunen (EVN)	Analysansvarig	<a href="mailto:eriva185@student.liu.se">eriva185@student.liu.se</a>
Wiliam Puranen (WP)	Konfigurationsansvarig	<a href="mailto:wilpu732@student.liu.se">wilpu732@student.liu.se</a>
Yadgar Suleiman (YS)	Dokumentansvarig	<a href="mailto:yadsu309@student.liu.se">yadsu309@student.liu.se</a>



## INNEHÅLL

1	Introduktion	1
1.1	Syfte	1
1.2	Mål	1
1.3	Begrepp och förkortningar	1
1.4	Verktyg	1
2	Enhetstest	3
2.1	Syfte	3
2.2	Ansvar	3
2.3	Tidsplan	3
2.4	Genomförande	3
3	Integrationstest	4
3.1	Syfte	4
3.2	Ansvar	4
3.3	Tidsplan	4
3.4	Genomförande	4
4	Systemtest	5
4.1	Syfte	5
4.2	Ansvar	5
4.3	Tidsplan	5
4.4	Genomförande	5
5	Användartest	9
5.1	Syfte	9
5.2	Ansvar	9
5.3	Tidsplan	9
5.4	Genomförande	9
5.5	Frågor till användartest	9
6	Acceptanstest	10
6.1	Syfte	10
6.2	Ansvar	10
6.3	Tidsplan	10
6.4	Genomförande	10
6.5	Godkänt	10



## DOKUMENTHISTORIK

Version	Datum	Utförda förändringar	Utförda av	Granskad
1.0	2024-02-19	Första utkast	CK	OD
2.0	2024-03-08	Lade till specifika systemtester för varje krav, åtgärdat grammatik- och stavfel. Ändrat en del formuleringar.	CK, WP	EH



# 1 INTRODUKTION

Detta dokument ämnar att förklara och definiera vilka olika tester som kommer att tillämpas samt genomföras under projektets gång. Under projektets gång är detta ett levande dokument och kommer att uppdateras löpande. Alltefter projektet växer och mer kod och funktioner tillämpas ställs allt högre Krav på programmerare att bibehålla en god kvalitet. Efter en viss punkt kommer det dock att bli omöjligt för en ensam individ att ha insyn i hela projektet. Därför behövs testning som säkerställer att koden kommer att interagera med resten av projektet på ett säkert sätt. En workshop har genomförts med samtliga projektdeltagare för att utbilda samtliga gruppmedlemmar om testning samt gemensamt komma fram till hur testning ska gå till.

## 1.1 Syfte

Syftet med detta dokument är att specificera vilka tester som kommer att genomföras samt hur dessa kommer att genomföras under projektets gång. Det ska också finnas en definition för hur varje Krav från Kravspecifikationen ska testas. Utan dessa tester finns det inget tydligt sätt att veta om ett Krav är uppfyllt. Per definition innebär detta också att om alla tester definierade i detta dokument har skapats och lyckats är alla Krav uppfyllda.

## 1.2 Mål

Genomförda tester och utvärderingar leder till ett stabilt system som uppfyller alla Krav.

## 1.3 Begrepp och förkortningar

- Black-box testning - Black-box testning innebär att testaren inte nödvändigtvis behöver ha någon tillgång till koden. Allt som testas är att skicka in värden och jämföra resultatet med ett förväntat resultat.
- White-box testning - I White-box testning känner testaren till koden och kan därför anpassa testerna för att testa olika delar.
- Botten-upp testning - En test-filosofi där man börjar med att testa den minst komplexa komponenten och därefter arbetar sig uppåt mot mer komplexa och interberoende komponenter.
- Test-coverage - Ett procentuellt mått på hur stor andel av koden i ett projekt som testas. Mäts automatiskt av vissa testverktyg

## 1.4 Verktyg

- Gitlab issue board - En typ av virtuell tavla som ger en överblick av alla pågående och planerade uppgifter, så kallade issues.
- Git-branch - Gitlab låter utvecklare arbeta parallellt på olika delar av kodbasen. När ny kod skrivits kan den sedan distribueras till alla i projektet genom att infoga (göra en commit) till huvudgrenen.
- Merge request - Ett sätt att låta andra utvecklare granska ändringar innan de infogas till huvudgrenen.
- Gitlab CI/CD-pipeline - Ett sätt att automatisera testning varje gång ny kod läggs till i en Git-branch.



- Container - En mjukvaru-container är ett portabelt paket som innehåller den miljö som krävs för att köra specifik mjukvara.
- Docker - Docker är ett verktyg för att hantera och packetera mjukvaru-containerar.



## 2 ENHETSTEST

Test av de separata, minsta funktionerna i projektet. För att säkerställa att de fungerar på korrekt vis behöver de testas enskilt. Om ett fel integreras i huvudprojektet kan det leda till att större delar av koden drabbas och mer resurser behöver ägnas åt att spåra och åtgärda felet. Denna typ av test behöver genomföras innan ny kod kan sammanfogas med projektets huvudgren i versionshanteringsverktyget.

### 2.1 Syfte

Syftet med enhetstester är att verifiera att den funktionalitet som eftersträvas hos en modul är uppfylld.

### 2.2 Ansvar

Alla som skrivit ny eller ändrat kod ansvarar för att den testas innan det infogas in i huvudgrenen.

### 2.3 Tidsplan

Testen skapas löpande varje gång en ny funktionalitet läggs till i koden. Det är sedan ett Krav att dessa tester lyckas innan en *merge-request* accepteras. Om testerna inte går igenom måste detta åtgärdas innan merge-requesten blir godkänd.

### 2.4 Genomförande

1. Ny funktionalitet skapas.
2. En *issue* för att skapa ett test för funktionaliteten läggs till i gruppens *issue-board*.
3. Testet skapas och läggs till i en dedikerad testmapp.
4. Testerna körs automatiskt i gruppens *CI/CD-Pipeline*.
5. Testverktygen visar även vilken *coverage* testerna har.
6. Ingen merge-request får accepteras innan alla tester i CI/CD-pipelinan har gått igenom.



### 3 INTEGRATIONSTEST

Vid integrationstest testas applikationen mer som helhet med flera olika funktioner. Målet med detta är att se till att applikationens delar fungerar tillsammans. Testen genomförs utefter botten-upp strategier med hjälp av *white-box* testning.

#### 3.1 Syfte

Syftet med integrationstesterna är att verifiera att alla funktionella krav är uppfyllda.

#### 3.2 Ansvar

Utvecklaren ansvarar själv för att integrationstester genomförs på moduler den arbetat med.

#### 3.3 Tidsplan

Testen sker då alla funktioner på en viss sida skapats.

#### 3.4 Genomförande

1. Skapa testkod för sidan med hjälp av test-funktionen i Flutter.
2. Utvecklare kör testkoden.
3. Utvärdera om eftersträvad funktionalitet är uppfylld.





## 4 SYSTEMTEST

Systemtest ämnar att testa applikationen som helhet och identifierar problem som uppstår när alla delar av koden interagerar med varandra. Testen genomförs med manuell Blackbox-testning.

### 4.1 Syfte

Syftet med systemtester är att verifiera att alla systemkrav samt funktionella krav är uppfyllda.

### 4.2 Ansvar

Testledare ansvarar för att systemtest genomförs.

### 4.3 Tidsplan

Systemtester ska vara genomförda och godkända senast 2024-05-01.

### 4.4 Genomförande

Följande tester utförs manuellt av utvecklare på både *Android*- och *iOS*-enheter

Test nr	1
Krav nr	20, 21
Syfte	Testa att skapa konto med flera separata användare.
Metod	Blackbox-test
Beskrivning	Skapa ett nytt konto på en enhet via email. Lägg till tre olika användare. Byt mellan alla tre användare.
Förväntat Resultat	Kontrollera att autentiseringsmail vid skapandet av kontot anländer inom 5 minuter. Vid byte av användare ska kontot förbli inloggat.
Godkänt	

Test nr	2
Krav nr	22, 23, 24
Syfte	Testa inloggning och synkronisering för existerande konton.
Metod	Blackbox-test
Beskrivning	Skapa två nya användare på en enhet som har anslutning till internet med en email. Lös matematiska uppgifter som den ena användaren så att nivå tre låses upp. På en annan enhet med anslutning till internet logga in med samma email.
Förväntat Resultat	Kontrollera att autentiseringsmail anländer inom 5 minuter. Kontrollera att användare 1 på den nya enheten har upp till nivå tre upplåst i addition samt att användare 2 inte har någonting upplåst.
Godkänt	



Test nr	3
Krav nr	2, 6, 7
Syfte	Testa att se statistiken i databasen för hur länge användare använder visualiseringsverktyget samt vilka knapptryckningar som görs.
Metod	Blackbox-test
Beskrivning	Starta en ny docker instans utan några användare. Skapa en ny profil med två användare. Gå in på visualiseringssidan i 1 minut för ena användaren. Stäng av internetkopplingen och gå in i visualiseringssidan 2 minuter för andra användaren. Lös ett tal inom addition korrekt. Återanslut internetkopplingen.
Förväntat Resultat	Kontrollera att tiden i visualiseringen för användare 1 och 2 är 1 minut respektive 2 minuter. Kontrollera att användare 2 har ett knapptryck på korrekt tal registrerat i databasen.
Godkänt	

Test nr	4
Krav nr	3, 4, 8
Syfte	Testa att designen av applikationen stämmer överens med kundens önsningar.
Metod	Blackbox-test
Beskrivning	Starta applikationen och öppna prototypen som godkänts av kunden. Gå igenom sidorna för inloggning, användarhantering, meny, nivåer räknesätt, statistik och inställningar.
Förväntat Resultat	Kontrollera att varje skärm är visuellt designad som Figma-prototypen. Kontrollera att appens menysida inte innehåller någon text. Kontrollera att statistik över addition, addition med okänd variabel, subtraktion, multiplikation, multiplikation med okänd variabel och division visas på statistiksidan.
Godkänt	



Test nr	5
Krav nr	1, 13, 14, 17
Syfte	Testa att applikationen får matematiska uppgifter från optimeringsalgoritmen, att inga animationer sker vid räkning och att användaren kan låsa upp alla räknesätten.
Metod	Blackbox-test
Beskrivning	Skapa en ny användare. Lös matematiska uppgifter korrekt tills nästa räknesätt är upplåst. Fortsätt lösa tal tills räknesätten addition, addition med okänd variabel, subtraktion, multiplikation, multiplikation med okänd variabel och division är upplåsta.
Förväntat Resultat	Kontrollera att alla räknesätt är tillgängliga för användaren och att nya matematiska uppgifter genereras.
Godkänt	

Test nr	6
Krav nr	5, 15, 16
Syfte	Testa visualisering av räknesätt.
Metod	Blackbox-test
Beskrivning	Skapa en ny användare och gå in via huvudmenyn till sidan för visualisering. Gå in på en nivå för addition, öppna därifrån visualisering för talet.
Förväntat Resultat	Kontrollera visualiseringen för addition, addition med okänd variabel, subtraktion, multiplikation, multiplikation med okänd variabel är tillgängligt. Kontrollera att visualiseringen öppnad från övningsidan visar talet som efterfrågades i övningen. Verifiera att visualiseringen är så som kunden har specificerat.
Godkänt	

Test nr	7
Krav nr	9, 10, 11, 12
Syfte	Testa användarinteraktion vid räkning.
Metod	Blackbox-test
Beskrivning	Skapa en ny användare och öppna räknesidan för addition. Svara först fel på ett tal. Svara sedan rätt på 10 tal.
Förväntat Resultat	Kontrollera att användaren får en indikation när fel svar anges. Kontrollera att det inte sker några animationer vid räkning av resterande 10 tal. Kontrollera att det senast räknade talet visas för användaren. Kontrollera att förloppsindikatorn uppdateras när de 10 talen räknats.
Godkänt	



Test nr	8
Krav nr	18, 19
Syfte	Testa applikationens inställningar.
Metod	Blackbox-test
Beskrivning	Skapa en ny användare och gå in i inställningar. Sätt på inställningen för ljudlöst. Skriv in en email i fältet för lärare. Gå in i addition och lös tre tal.
Förväntat Resultat	Kontrollera att inget ljud hörs vid lösning av tal. Kontrollera i databasen att användaren är associerad med lärarens email.
Godkänt	

Test nr	9
Krav nr	25, 26, 30
Syfte	Testa att applikationen fungerar på äldre telefoner.
Metod	Blackbox-test
Beskrivning	Testa applikationen på en iPhone 5S. Gå in på addition och lös 3 tal. Testa applikationen på Android 5. Gå in på addition och lös 3 tal.
Förväntat Resultat	Kontrollera att tiden mellan talen max är 150 ms.
Godkänt	

Test nr	10
Krav nr	27, 28
Syfte	Testa offline-funktionalitet.
Metod	Blackbox-test
Beskrivning	Skapa en ny användare på en enhet utan tillgång till internet. Gå in i addition och lös 3 tal.
Förväntat Resultat	Kontrollera att all funktionalitet som behövs för att kunna räkna matematik finns tillgänglig.
Godkänt	



## 5 ANVÄNDARTEST

Användartester bygger på att låta användare som stämmer överens med applikationens riktiga målgrupp använda och ge återkoppling på applikation.

### 5.1 Syfte

Syftet med användartest är användare som stämmer överens med applikationens målgrupp får testa produkten och ge återkoppling. Återkopplingen kan sedan användas för att förbättra applikation samt att validera att vidareutvecklingen av applikation faktiskt har lett till en positiv förändring.

### 5.2 Ansvar

Testledare och analysansvarig i samråd med kunden ansvarar för att användartest utförs.

### 5.3 Tidsplan

Ett användartest planeras att utföras vecka 17.

### 5.4 Genomförande

1. Kunden bokar tid med en skolklass i en timme och 20 minuter.
2. Installera vidareutvecklad versionen av applikationen på en enhet och versionen från projektstart på en annan enhet. Mobiltelefoner tillhandahålls av kunden.
3. Kalla in två barn åt gången.
4. Barnen genomför additionstal i 5 minuter.
5. Barnen svarar på frågor om applikationen de precis använde.
6. Barnen byter telefon och löser additionstal i 5 minuter.
7. Barnen svarar återigen på frågor om den applikationen de precis använde samt frågor om de båda applikationerna jämfört med varandra.
8. Upprepa punkt 3-7 tills totalt 8 barn genomfört testet.

### 5.5 Frågor till användartest

1. Var applikationen lätt att använda
2. Kände du någon gång att du inte visste hur du skulle göra någonting? Vad i så fall?
3. Skulle du kunna tänka dig att använda applikationen igen för att räkna matematik?
4. Tyckte du att det var kul att räkna matematik i applikationen?



## 6 ACCEPTANSTEST

Efter att utvecklingen är klar skickas produkten till kunden som kontrollerar att alla krav har uppfyllts.

### 6.1 Syfte

Syftet med acceptanstester är att kunden kan verifiera att all önskad funktionalitet uppfylls och att denne är nöjd. Därefter beslutas om produkten är godkänt för publicering, beslutat baseras främst på om samtliga krav från kravspecifikationen har uppfyllts [1].

### 6.2 Ansvar

Acceptanstest är kundens ansvar.

### 6.3 Tidsplan

Förslag på slutgiltig produkt levereras till kunden 2024-05-08, i samband med denna leverans är det upp till kunden att genomföra acceptanstester inom en vecka.

### 6.4 Genomförande

Acceptanstest genomförs självständigt av kund.

### 6.5 Godkänt

Testet anses vara godkänt om kunden godkänner produkten.



## REFERENSER

[1] PUM09, “Kravspecifikation optimalt lärande,” 2024.