

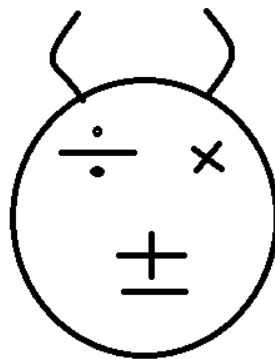


# Testplan

PUM09

2024-02-19

Version 1.0



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: [lana.buffoni@liu.se](mailto:lana.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	2
2.1	Syfte . . . . .	2
2.2	Ansvar . . . . .	2
2.3	Tidsplan . . . . .	2
2.4	Genomförande . . . . .	2
3	Integrationstest	3
3.1	Syfte . . . . .	3
3.2	Ansvar . . . . .	3
3.3	Tidsplan . . . . .	3
3.4	Genomförande . . . . .	3
4	Systemtest	4
4.1	Syfte . . . . .	4
4.2	Ansvar . . . . .	4
4.3	Tidsplan . . . . .	4
4.4	Genomförande . . . . .	4
5	Användartest	5
5.1	Syfte . . . . .	5
5.2	Ansvar . . . . .	5
5.3	Tidsplan . . . . .	5
5.4	Genomförande . . . . .	5
6	Acceptanstest	6
6.1	Syfte . . . . .	6
6.2	Ansvar . . . . .	6
6.3	Tidsplan . . . . .	6
6.4	Genomförande . . . . .	6
6.5	Godkänt . . . . .	6



## DOKUMENTHISTORIK

Version	Datum	Utförda förändringar	Utförda av	Granskad
1.0	2024-02-19	Första utkast	CK	OD



# 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.

## 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.

## 1.2 Mål

Genomförda tester och utvärderingar leder till ett stabilt system utan buggar.

## 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 inter beroende komponenter.

## 1.4 Verktyg

- Gitlab issue board:  
En typ av virtuell tavla som ger en överblick av alla pågående och planerade uppgifter.
- 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.



## 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 main branchen.

### 2.3 Tidsplan

Testen sker löpande alltefter nya funktioner införs.

### 2.4 Genomförande

1. Kör kodanalys verktyget i Flutter.
2. Lägg in en issue för testning av koden i gitlab boarden.
3. Annan utvecklare än den som skrev koden gör en code review.
4. Om godkänt infoga koden i projektets main branch.



### 3 INTEGRATIONSTEST

Vid integrationstest testas applikationen mer som helhet med flera olika funktioner. Detta testar att enheterna fungerar tillsammans. Testen genomförs utefter botten-upp strategin med hjälp av whitebox testning.

#### 3.1 Syfte

Verifierar att alla funktionella krav är uppfyllda.

#### 3.2 Ansvar

De som skrivit koden ansvarar för att testet genomförs.

#### 3.3 Tidsplan

Testen sker då alla funktioner på en viss sida skapats.

#### 3.4 Genomförande

1. Skapa testkod för sidan med hjälp av testnings funktionen i Flutter.
2. Utvecklare kör testkoden.
3. Utvärdera om eftersträvad funktionallitet är uppfyllt.



## 4 SYSTEMTEST

Systemtestet testar hela appen och identifierar problem som uppstår när alla delar av koden interagerar med varandra. Testen genomförs med manuell blackbox testning.

### 4.1 Syfte

Syftet är att verifiera alla funktionella och systemkrav är uppfyllda.

### 4.2 Ansvar

Testansvarig har ansvaret för systemtest genomförs.

### 4.3 Tidsplan

Testet sker så snart som möjligt efter alla integrationstester genomförts.

### 4.4 Genomförande

1. Skapa testfall för kontohantering och de olika räknesätten.
2. Utvecklare genomför manuellt i applikationen de olika testerna.
3. Upprepa testerna på både android och IOS enheter.





## 5 ANVÄNDARTEST

I samråd med kunden kommer att användartest på en skolklass utföras.

### 5.1 Syfte

Syftet med användartestet är att både kunden och riktiga användare kan testa produkten och ge feedback.

### 5.2 Ansvar

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

### 5.3 Tidsplan

Testet utförs så snart som möjligt efter systemtest.

### 5.4 Genomförande

1. Rekrytera en lågstadielklass och boka en och en halv timme med dem i deras klassrum.
2. Installera appen på en varierad mängd äldre mobiltelefoner som tillhandahålls av kund.
3. Vid den bokade tiden dela ut mobiltelefonerna till alla individer i klassen.
4. Utför testet och låt användarna spendera 15 minuter per räknesätt.
5. Håll en 20 minuter lång gruppdiskussion om användarupplevelsen där en person antecknar.
6. Utvärdera resultatet i grupp och i samråd med kunden.



## 6 ACCEPTANSTEST

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

### 6.1 Syfte

Syftet är att kunden kan verifiera att alla krav uppfyllts och att denne är nöjd. Därefter beslutas om produkten är redo för leverans.

### 6.2 Ansvar

Acceptanstest är kundens ansvar.

### 6.3 Tidsplan

TBD

### 6.4 Genomförande

Acceptanstest genomförs självständigt av kund.

### 6.5 Godkänt

Testet är accepterat om de krav specificerade i kravspecen är uppfyllda.



## REFERENSER