

PH Bicycle Storage 8000

Testplan

Projektdeltagare:

Mattias Nordahl (dt07mn0@student.lth.se)
Hannes Nevalainen (dt07hn2@student.lth.se)
Daniel Olofsson (dt07do1@student.lth.se)
Fredrik Andersson (dt07fa5@student.lth.se)
Emil Einarsson (dt07ee3@student.lth.se)
Sandra Nilsson (dt07sn8@student.lth.se)

Gruppenr: 18

Kursansvarig: Martin Höst

Dokumentansvarig: Daniel Olofsson
Version: 1.1

2008-04-28

Innehåll

1	Ändringshistorik	2
2	Referenser	2
3	Introduktion	2
3.1	Testat system	2
4	Testprocessen	2
4.1	Testprocessens översikt	2
4.2	Enhetstest	2
4.3	Integrationstest	3
4.4	Systemtest	3
4.5	Acceptanstest	3
5	Testade punkter	4
6	Dokumentationsmetoden för testning	4
6.1	Enhetstest	4
6.2	Integrationstest	4
6.3	Systemtest	4
6.4	Acceptanstest	4
7	Testfall för systemtestning	4
7.1	Testfall	4
7.2	Kravtäckning och spårbarhet	5
A	Appendix: Testfall	6
A.1	Definitioner	6
A.2	Testfall för funktionella krav	6
A.3	Testfall för icke-funktionella krav	19
B	Appendix: Mall för testprotokollet	21

1 Ändringshistorik

Version	Datum	Ansvarig	Beskrivning
0.1	2008-04-15	D.O.	Start av dokument
0.2	2008-04-15	D.O.	Rättning
0.3	2008-04-23	D.O.	Testfall tillagda
0.4	2008-04-27	D.O.	Rättning efter granskning
1.0	2008-04-28	D.O.	Baseline
1.1	2008-04-28	D.O.	Ändringar i Appendix B Ändrat testfall

2 Referenser

[1] *Project and Exercises in the Software Engineering Process*
Software Engineering Research Group
Department of Computer Science
Lund University, 2008-03-25

[2] Kravspecifikation version 1.0

3 Introduktion

3.1 Testat system

Systemet som ska testas är ett cykelgarage, där en användare och användarens cykel kan registreras av en operatör. Sedan kan användaren ställa in sin cykel genom att läsa av streckkoden på cykeln. När användaren har en inställd cykel kan han även hämta cykeln genom att först slå in sin PIN-kod (bestående av de sex första siffror i personnummret plus en fyrsiffrig pin) och sedan hämta sin cykel och sedan läsa av streckkoden på sin cykel. Anledning för det här dokumentet är att ha en specifikation för testningen av cykelgaraget mjukvara. Testprocessen kommer bestå av följande delar: Enhetstest, integrationstest, systemtest och acceptanstest.

4 Testprocessen

4.1 Testprocessens översikt

Testen kommer utföras enligt projektmodellen som presenterats i [1]. Testprocessen kommer bestå av följande delar: Enhetstest, integrationstest, systemtest och acceptanstest. Regressionstestning kommer också utföras när fel hittas.

4.2 Enhetstest

Enhetstestningen kommer ske med strukturell testning (white-box). Varje utvecklare ska ha testat varje rad av sin kod innan han lämnar in någon slutgiltig version av koden.

- Utförs av: Utvecklarna

- Typ av test: Strukturell
- Kriterier: Fullständig täckning av alla satser
- Slutfört: När alla testen går igenom utan fel

Enhetstestningen utvecklas inte mer i detta dokument.

4.3 Integrationstest

Integrationstestningen utförs av utvecklarna, men det behöver inte vara samma utvecklare som utvecklat enheterna som integreras i programmet. Integrationstestningen ska testa alla anrop mellan enheternas API (Application Programming Interface).

- Utförs av: Utvecklarna
- Typ av test: Strukturell
- Kriterier: Fungerar korrekt med användargränssnittet
- Slutfört: När alla testen går igenom utan fel

Integrationstestningen utvecklas inte mer i detta dokument.

4.4 Systemtest

Systemtestningen är baserad på kraven som har blivit definierade för systemet.

- Utförs av: Utvecklarna
- Typ av test: Funktionell
- Kriterier: Alla krav uppfylls
- Slutfört: Inga kritiska fel upptäcks när alla definierade testfall körts

Testfallen som kommer att användas under systemtestet finns under Appendix A.

Alla krav enligt [2] ska bli verifierade av minst ett testfall och alla testfall ska verifiera minst ett krav.

4.5 Acceptanstest

Acceptanstestning utvecklas inte mer i detta dokument, eftersom det görs av kunden.

5 Testade punkter

Enhetstestet utförs huvudsakligen med hjälp av källkoden. Kraven används huvudsakligen här för att kontrollera om testfallet är lyckat om det inte kan bestämmas med hjälp av koden. Integrationstestet använder källkoden och designen för att kontrollera om testfallen uppfyller kriterierna framställda ovan. Systemtestet använder huvudsakligen kraven för att kontrollera testfallen och

för att validera systemet. Acceptanstestet använder endast den slutgiltiga produkten.

Enhets- och integrationstestet utförs i utvecklingsmiljö. Om det behövs så kommer tillfälliga implementationer göras för att enhets- och integrationstestet ska kunna genomföras. Systemtestet bör göras med ett riktigt cykelgarage, men själva garaget hör inte till detta projektet därför kommer Systemtestet utföras med virtuell hårdvara.

6 Dokumentationsmetoden för testning

6.1 Enhetstest

Varje utvecklare testar sin kod. Resultaten behöver inte dokumenteras någonsans.

6.2 Integrationstest

Vid varje testtillfälle ska ett dokument enligt mallen i Appendix B skapas. Detta dokument ska innehålla alla testfallens resultat (ifall de gått igenom eller misslyckats). Dokumentet kommer att sparas på vår ftp-server tills slutet av projektet.

6.3 Systemtest

Vid varje testtillfälle ska ett dokument enligt mallen i Appendix B skapas. Detta dokument ska innehålla alla testfallens resultat (ifall de gått igenom eller misslyckat). Dokumentet kommer att sparas på vår ftp-server tills slutet av projektet.

6.4 Acceptanstest

Detta test utförs av kunden och beskrivs inte här.

7 Testfall för systemtestning

7.1 Testfall

Det är endast systemtestfallen som presenteras här. Enhets- och integrationstestfallen görs med hjälp av koden och designen och acceptanstestning görs av kunden. Testfallen går att hitta under Appendix A.

7.2 Kravtäckning och spårbarhet

Kravtäckningen betyder att alla funktionella krav ska bli testade av minst ett testfall så det är säkert att all funktionellitet blir testad. En tabell över alla krav och testfall finns bifogat i bilaga 1.

A Appendix: Testfall

A.1 Definitioner

Pre	Villkoret före testfallet körs
Post	Villkoret efter testfallet har körts
Gui	Användargränssnittet

A.2 Testfall för funktionella krav

Testfall 1. Grön lampa ska lysa för korrekt PIN-kod

- Pre: Terminalen är i startläge, 8812291234 är en giltig PIN-kod
 - Post: Terminalen är i startläge, händelsen har loggats
1. Grön lampa blinkar en gång var femte sekund
 2. Användaren anger en korrekt PIN-kod (881229 1234)
 3. Låset öppnas
 4. Händelsen loggas
 5. Grön lampa tänds
 6. Efter tio sekunder läses låset igen
 7. Grön lampa slocknar
 8. Grön lampa blinkar en gång var femte sekund
 9. Kontrollera att loggposten är korrekt formaterad

Testfall 2. Grön lampa ska lysa för korrekt streckkod

- Pre: Terminalen är i startläge
 - Post: Terminalen är i startläge, händelsen har loggats
1. Användaren skannar en streckkod som är giltig
 2. Låset öppnas
 3. Händelsen loggas
 4. Grön lampa tänds
 5. Efter tio sekunder läses låset igen
 6. Grön lampa slocknar
 7. Kontrollera att loggposten är korrekt formaterad

Testfall 3. Röd lampa ska lysa om fel PIN-kod matas in

- Pre: Terminalen är i startläge, 1234567890 är en ogiltig PIN-kod
- Post: Terminalen är i startläge, dörren öppnas inte, felet har loggats

1. Användare matar in felaktig PIN-kod (1234567890)
2. Röd lampa tänds
3. Felet loggas
4. Det går ej att öppna dörren
5. Efter tre sekunder slocknar röd lampa
6. Kontrollera att loggposten är korrekt formaterad

Testfall 4. Röd lampa ska lysa för felaktig streckkod

- Pre: Terminalen är i startläge
 - Post: Terminalen är i startläge, dörren öppnas inte, felet har loggats
1. Användare skannar en streckkod som inte är giltig
 2. Röd lampa tänds
 3. Felet loggas
 4. Det går ej att öppna dörren
 5. Efter tre sekunder slocknar röd lampa
 6. Kontrollera att loggposten är korrekt formaterad

Testfall 5. Ägaren slår rätt PIN-kod men har ingen cykel i garaget

- Pre: Terminalen är i startläge, 8812291234 är en giltig PIN-kod, det finns inga cyklar i garaget för användaren
 - Post: Terminalen är i startläge. Dörren öppnas inte
1. Användaren anger en korrekt PIN-kod (8812291234)
 2. Grön och röd lamp blinkar tre gånger
 3. Det går ej att öppna dörren

Testfall 6. Om det är tio sekunder mellan två inmatningar återställs PIN-kodsterminalen

- Pre: Terminalen är i startläge, 8812291234 är en giltig PIN-kod
 - Post: Terminalen är i startläge. Dörren öppnas inte
1. Användaren anger 881229
 2. Användaren väntar i elva sekunder
 3. Röd lampa blinkar tre gånger
 4. Terminalen är i startläge
 5. Användaren anger 1234

6. Dörren går inte att öppna
7. Användaren väntar i elva sekunder
8. Terminalen är i startläge

Testfall 7. * (stjärna) återställer PIN-kodsterminalen till startläge

- Pre: Terminalen är i startläge, 8812291234 är en giltig PIN-kod
 - Post: Terminalen är i startläge
1. Användaren anger 881229
 2. Användaren trycker på *
 3. Terminalen är i startläge
 4. Användaren anger 1234
 5. Dörren går inte att öppna
 6. Användaren trycker på *
 7. Terminalen är i startläge

Testfall 8. Låset är upplåst i tio sekunder

- Pre: Terminalen är i startläge, 8812291234 är en giltig PIN-kod
 - Post: Terminalen är i startläge, händelsen har loggats
1. Användaren anger 8812291234
 2. Låset öppnas
 3. Händelsen loggas
 4. Användaren väntar i nio sekunder
 5. Användaren försöker öppna dörren
 6. Dörren går att öppna
 7. Användaren stänger dörren
 8. Användaren väntar i en sekund
 9. Användaren försöker öppna dörren
 10. Dörren går inte att öppna
 11. Kontrollera att loggposten är korrekt formaterad

Testfall 9. Sortera cykeltabellen genom att klicka på kolumnrubriker

- Pre: GUI:t befinner sig i läget som det ser ut när man startade programmet. Det finns minst 10 cyklar i systemet.

- Post: Återställ GUI:t till grundtillståndet

1. Gå in på “Visa” i menyraden
2. Klicka på “Alla cyklar”
3. Klicka på rubrikerna en i taget
4. Observera hur tabellen sorteras om
5. Töm tabellen

Testfall 10. Sortera användartabellen genom att klicka på kolumnrubriker

- Pre: GUI:t befinner sig i läget som det ser ut när man startade programmet. Det finns minst 10 st användare och 10 st cyklar i systemet, ojämnt fördelat över 5 av användarna.

- Post: Återställ GUI:t till grundtillståndet programmet.

1. Gå in på “Visa” i menyraden
2. Klicka på “Alla användare”
3. Klicka på rubrikerna en i taget
4. Observera hur tabellen sorteras om
5. Töm tabellen så att den blir tom

Testfall 11. Statistikrutan

- Pre: GUI:t befinner sig i läget som det ser ut när man startade programmet. Systemet är tomt.

- Post: GUI:t befinner sig i läget som det ser ut när man startade programmet. Systemet innehåller 5 st användare och 4 st cyklar. Händelserna har loggats.

1. Lägg till en ny användare genom att klicka på knappen “Ny användare” och fylla i.
2. Lägg till två nya cyklar till föregående användare genom att klicka på knappen “Ny cykel” och fylla i.
3. Lägg till två nya användare genom att klicka på knappen “Ny användare” och fylla i.
4. Lägg till vardera en cykel till dessa användare genom att klicka på knappen “Ny cykel” och fylla i.
5. Lägg till två nya användare genom att klicka på knappen “Ny användare” och fylla i.
6. Öppna den första och tredje cykeln som skapades och välj att dessa ska ha status “I garaget”.

7. Kontrollera att statistikrutan visar:
Totalt antal cyklar: 4
Totalt antal användare: 5
Cyklar i garaget: 2
Cyklar/användar ratio: 80%

8. Kontrollera att loggposten är korrekt formaterad

Testfall 12. Information om användare

- Pre: GUI:t befinner sig i läget som det ser ut när man startade programmet. Systemet innehåller minst 10 st användare.
 - Post: GUI:t befinner sig i startläge. Händelserna har loggats.
1. Gå in på “Visa” i menyraden
 2. Klicka på “Alla användare”
 3. Öppna den sista användaren i lista
 4. Avläs följande informationen:
 - Personnummer
 - PIN-kod
 - Namn
 - Hemtelefon
 - Mobil
 - Adress
 - Övriga anteckningar
 - Antal registrerade cyklar
 - När användaren registrerades
 - Senaste ingång
 - Antal inloggningar
 5. Stäng användaren
 6. Starta om programmet
 7. Öppna samma användare igen och kontrollera att informationen är identisk med hur den var tidigare
 8. Kontrollera att loggposten är korrekt formaterad

Testfall 13. Information om cykel

- Pre: GUI:t befinner sig i läget som det ser ut när man startade programmet. Systemet innehåller minst 10 st cyklar.
- Post: GUI:t befinner sig i startläge. Händelserna har loggats.

1. Gå in på “Visa” i menyraden
2. Klicka på “Alla cyklar”
3. Öppna den sista cykeln i lista
4. Avläs följande information:
 - Id-nummer (streckkoden)
 - Beskrivning
 - Ev. ramnummer
 - Koppling till ägare
 - Övriga anteckningar
 - Om cykeln finns i garaget
 - Tid för senast in och utgång
 - När cykeln registrerades i systemet
5. Stäng cykeln
6. Starta om programmet
7. Öppna samma cykeln igen och kontrollera att informationen är identisk med hur den var tidigare
8. Kontrollera att loggposten är korrekt formaterad

Testfall 14. Testar om mer än en användare kan få samma personnummer/kod

- Pre: Systemet är tomt
 - Post: Systemet innehåller 10 000 användare och 10 000 cyklar. Händelserna är loggade.
1. Skapa 10 000 användare, där de 6 första siffrorna i personnumret är identiskt
 2. Ge alla användare varsin cykel
 3. Sätt alla cyklarnas status till att stå i garaget
 4. Ange de sex första siffrorna följt av alla PIN-kodskombinationer, 0000-9999, en efter en. Alla ska utgöra en korrekt kombination.

Testfall 15. Söka efter en cykel

- Pre: Det finns minst en cykel/användare i systemet.
 - Post: Den eftersökta cykeln visas i fönstret
1. Markera “Cykel”-fliken
 2. Klicka på sökrutan i hörnet
 3. Skriv in cykeln du söker efter som du vet finns i systemet.

4. Cykeln visas i fönstret

Testfall 16. Söka efter en användare

- Pre: Det finns minst en cykel/användare i systemet.
 - Post: Den eftersökta personen visas i fönstret
1. Markera "Användar"-fliken
 2. Klicka på sökrutan i hörnet
 3. Skriv in namnet på en personen som du vet finns i systemet.
 4. Personen visas i fönstret.

Testfall 17. Borttagning av cykel

- Pre: Det finns minst en cykel/användare i systemet.
 - Post: Fönstret ska vara tomt. Händelsen är loggad
1. Dubbelklicka på cykeln som du vill ta bort.
 2. Klicka på "Ta bort"
 3. En dialogruta kommer upp
 4. Du klickar "Nej"
 5. Kontrollera så att cykeln är kvar.
 6. Klicka "Ta bort" igen
 7. Klicka "Ja" i dialog rutan.
 8. Händelsen loggas
 9. Kontrollera så att cykeln är borta.
 10. Kontrollera att loggposten är korrekt formaterad

Testfall 18. Operatrören kan ändra statusen för en cykel

- Pre: Det finns minst en cykel i systemet.
 - Post: Cykelns status ska ha ändrats, händelsen är loggad
1. Markera cykeln
 2. Ändra status
 3. Händelsen loggas
 4. Kolla så att statusen ändrats

Testfall 19. Alla registrerade användare kan ses i en tabell

- Pre: Det finns minst en användare i systemet.

- Post: Alla användare visas
1. Klicka på “Användare”-fliken
 2. Alla användare skall nu visas

Testfall 20. Alla registrerade cyklar kan ses i en tabell

- Pre: Det finns minst en cykel i systemet.
 - Post: Alla cyklar visas
1. Klicka på “Cyklar”-fliken
 2. Alla cyklar skall nu visas

Testfall 21. En cykel ska inte kunna tas bort om den är inställd i garaget

- Pre: Det finns minst en cykel/användare i systemet.
 - Post: Programmet i startläget, cykeln finns kvar i registret, händelsen är loggad
1. Växla till tabellen med cyklar som finns i garaget.
 2. Dubbelklicka på valfri cykel
 3. Tyck på knappen “Ta bort”.
 4. Systemet varnar och säger att cykeln inte kan tas bort för att den finns i garaget.
 5. Ändra cykeln så att den är markerad som att den inte finns i garaget.
 6. Tryck på “Ta bort” knappen igen.
 7. Systemet frågar om du vill ta bort cykeln.

Testfall 22. All data om en cykel ska kunna ändras (ej streckkoden)

- Pre: Det finns minst en cykel i systemet.
 - Post: Markerad cykel är ändrad, händelsen är loggad
1. Välj en cykel.
 2. Öppna upp cykeln.
 3. Ändra fälten till följande:
 - Beskrivning: Jag är en glad banan.
 - Ramnummer: 1231-AD-32
 - Övrig anteckning: cykeln är rostig
 - Cykeln finns i garaget: ändra till motsatt tillstånd.
 4. Tryck på “Spara”

5. Händelsen loggas
6. Stäng fönstret
7. Öppna upp samma cykel igen och se om datan stämmer med ovan.
8. Kontrollera att loggposten är korrekt formaterad

Testfall 23. Ändra användar information

- Pre: Det finns minst en cykel/användare i systemet.
- Post: Användarinformationen ändras, händelsen är loggad

1. Välj en användare.
2. Öppna upp användaren.
3. Ändra fälten till följande:
 - Namn: Kalle Karlsson
 - Hemtelefon: 1111-11 11 11
 - Mobil: 2222-22 22 22
 - Adress: Gula gatan 23, 222 33 Gulberga
 - Övriga anteckningar: Jag är en gul banan.
4. Tryck på "Spara"
5. Händelsen loggas
6. Stäng fönstret
7. Öppna upp samma användare och verifiera att datan stämmer med ovan.
8. Kontrollera att loggposten är korrekt formaterad

Testfall 24. Utskrift av befintlig streckkod

- Pre: GUI:t är i startläge
- Post: En befintlig streckkod har skrivits ut och händelsen loggats

1. Välj en cykel.
2. Öppna upp cykeln.
3. Tryck på knappen "Skriv ut streckkod"
4. Streckkoden skrivs ut
5. Händelsen loggas
6. Kontrollera att loggposten är korrekt formaterad

Testfall 25. En cykel måste kopplas till en ägare

- Pre: Systemet är i startläge

- Post: Systemet är i startläge
1. Tryck på knappen "Ny Cykel" utan att markera en användare
 2. Ingen cykel kommer läggas till
 3. Systemet är i startläge

Testfall 26. En användare kan ha noll, en eller flera cyklar

- Pre: Systemet är i startläge
 - Post: Systemet är i startläge och händelserna har loggats
1. En ny användare skapas utan någon cykel
 2. Händelse loggas
 3. Kontrollera att användaren har noll cyklar
 4. Markera den nya användaren och tryck på knappen "Ny Cykel"
 5. Lägg till informationen om den nya cykeln
 6. Händelsen loggas
 7. Kontrollera att användaren har en cykel
 8. Markera användaren igen och tryck på knappen "Ny Cykel"
 9. Lägg till informationen om den nya cykeln
 10. Händelsen loggas
 11. Kontrollera att användaren har två cyklar

Testfall 27. En användare kan bara komma in i garaget med sin PIN-kod om användaren har en cykel där

- Pre: Det finns minst en cykel i garaget kopplad till en användare
 - Post: Cykelns ägare kommer in i garagen med sin PIN-kod, händelsen är loggad
1. Användaren kommer till garaget utan cykel
 2. Användaren trycker in sin PIN-kod
 3. Låset öppnas
 4. Händelsen loggas
 5. Ägaren kan komma in i garaget
 6. Kontrollera att loggposten är korrekt formaterad

Testfall 28. En användare slår sin PIN-kod för att komma in i garaget

- Pre: En ägare har en/flera cyklar i garaget och en utanför

- Post: Dörrens lås är öppet, händelsen är loggad

1. Ägaren trycker in sin PIN-kod
2. Händelsen loggas
3. Låset öppnas
4. Händelsen loggas
5. Ägaren kan komma in i garaget
6. Kontrollera att loggposten är korrekt formaterad

Testfall 29. En användare läser av en sträckkod för att komma in i garaget

- Pre: En ägare har en/flera cyklar i garaget och en utanför
- Post: Portlåset låses upp, händelsen är loggad

1. Ägaren läser in streckkoden på sin cykel
2. Låset öppnas
3. Händelsen loggas
4. Ägaren kan komma in med sin cykel i garaget
5. Kontrollera att loggposten är korrekt formaterad

Testfall 30. En person identifieras genom sitt personnummer

- Pre: Användaren trycker in sitt personnummer
 - Post: Användaren identifieras
1. Användaren får trycka in sitt personnummer, ÅÅMMDD, på operatörens dator
 2. Personnummret letas upp i databasen
 3. Användaren identifieras

Testfall 31. En cykel identifieras av sin sträckkod som är unik för varje cykel

- Pre: Det finns flera olika cyklar i systemet
- Post: Cykeln kan nu identifieras med hjälp av streckkoden

1. Öppna upp olika cyklar
2. Jämför cyklarnas streckkoder
3. Ingen cykel har samma streckkod
4. En cykel kan bli identifierad med streckkoden

Testfall 32. Cykelns ID-nummer är 5 siffror lång

- Pre: En cykel har ett ID nummer
 - Post: ID-nummret är 5 siffror långt
1. Öppna tio cyklar
 2. Kontrollera ID-nummrets längd
 3. ID-nummret är 5 siffror långt för varje cykel

Testfall 33. Cykel som redan är i garaget skannas vid ingång

- Pre: Cykel finns i garaget
 - Post: Cykelns inställningstid uppdateras och händelsen loggas
1. Cykel tas ut ur garaget utan att skannas av
 2. Cykeln tas tillbaka till garaget
 3. Cykeln skannas av
 4. Inställningstiden ändras till den nya tiden cykeln ställdes in
 5. Händelsen loggas i loggen
 6. Kontrollera att loggposten är korrekt formaterad

Testfall 34. Cykel som redan är ute ur garaget skannas vid utgång

- Pre: Cykeln finns i garaget men har statusen som ute
 - Post: Cykelns utgångstid uppdateras och händelsen loggas
1. Ägaren kommer till garaget och slår sin PIN-kod
 2. Ägaren tar ut sin cykel och skannar av den
 3. Utgångstiden uppdateras
 4. Händelsen loggas i loggen
 5. Kontrollera att loggposten är korrekt formaterad

Testfall 35. Hårdvaran ska kunna användas parallellt

- Pre: Ingen hårdvara används
 - Post: Ingen hårdvara används
1. En användare skannar av en cykel samtidigt som operatören lägger till ny användare i systemet
 2. Användarens cykel skannas av och den nya användaren läggs till i systemet
 3. Ingen hårdvara används

Testfall 36. PIN-kods terminalen återställs till startläge när streckkod skannas under inmatning

- Pre: PIN-kodensterminalen är i startläge
 - Post: PIN-kodensterminalen är i startläge
1. Ägaren börjar slå sin PIN-kod
 2. Ägaren skannar av sin cykel
 3. PIN-kodsterminalen går tillbaka till startläge

Testfall 37. PIN-kodsterminalen ska befinna sig i start läge så länge dörren är upplåst

- Pre: Ägaren slår in sin PIN-kod eller skannar sin cykel
 - Post: PIN-kodsterminalen är kvar i startläget, händelsen är loggad
1. En ägaren slår sin PIN-kod eller skannar sin cykel
 2. Dörrens lås öppnas
 3. Händelsen loggas i loggen
 4. PIN-kodsterminalen ställs i startläge
 5. En annan ägare försöker slå in sin PIN-kod
 6. PIN-kodsterminalen stannar i startläge
 7. Kontrollera att loggposten är korrekt formaterad

Testfall 38. En ny streckkod avläses när dörren är upplåst

- Pre: Dörren är upplåst
 - Post: Dörren är låst
1. Dörren är upplåst
 2. En cykel läses av vid ingången inom tio sekunder
 3. Dörren är upplåst i tio sekunder till
 4. Dörren är låst

Testfall 39. Fel och händelser ska loggas separat

- Pre: Ett antal fel och händelser har inträffat.
 - Post: -
1. Öppna upp mappen med loggfilerna
 2. Kontrollera att alla fel som inträffat finns i en fil och alla händelser som inträffat finns i en annan fil

A.3 Testfall för icke-funktionella krav

Testfall 40. Systemkrasch

- Pre: Systemet är igång
 - Post: Systemet har inte förlorat någon data
1. Systemt körs
 2. Strömsladden dras ut ur systemet
 3. Strömsladden kopplas in
 4. Systemt startas
 5. Systemet har inte förlorat någon data

Testfall 41. Systemet utför åtgärd

- Pre: Systemet är igång
 - Post: Systemet är igång
1. Operatören trycker på en knapp i systemet
 2. Det går mindre än en sekund
 3. Knappens funktions utförs
 4. Systemet är igång

Testfall 42. Systemklockan ställs under körning

- Pre: Systemet är igång
 - Post: Systemklockan är omställd och systemet fungerar
1. Systemet är fullt funktionellt
 2. Systemklockan ändras
 3. Systemet fungerar fortfarande korrekt

Testfall 43. Systemet kan hantera 100 000 cyklar

- Pre: Systemet har inga cyklar inlagda i databasen
 - Post: Systemet har 100 000 cyklar inlagda i databasen
1. Systemet har noll cyklar inlagda i databasen
 2. 100 000 cyklar läggs in i systemet
 3. Systemet har 100 000 cyklar inlagda i databasen

Testfall 44. Systemet kan hantera 100 000 användare

- Pre: Systemet har 0 användare inlagda i databasen

- Post: Systemet har 100 000 användare inlagda i databasen

1. Systemet har noll användare inlagda i databasen

2. 100 000 användare läggs in i systemet

3. Systemet har 100 000 användare inlagda i databasen

Testfall 45. Användargränssnittet har stöd för flera språk.

Detta testfall är möjligt att testa genom att ändra språket i programmet till något annat tillgängligt språk och se om språket i hela systemet ändras.

B Appendix: Mall för testprotokollet

Här under följer vad varje testprotokoll som skapas ska innehålla. Det är endast när ett integrationstestet eller systemtestet som ett testprotokoll ska skapas. Dokumentansvarig kommer Emil Einarsson att vara.

Namn på testare: (Vem har utfört testet)

Datum för testet: (När utfördes testet)

Version av kravspecifikationen: (Vilken version av kravspecifikationen användes)

Kod som har testats: (Vilken kod har testats)

Hur utfördes testet: (Beskrivning av utförandet av testet)

Resultatet av testet: (Resultatet)