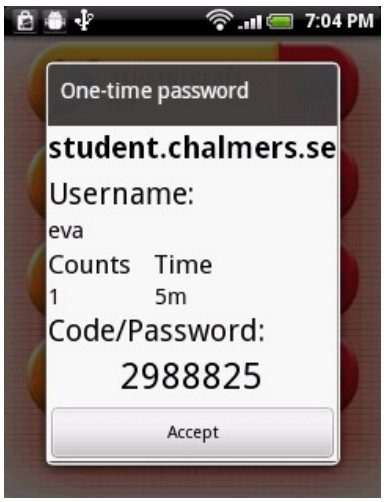
Rapport – Mobile Keyring



Mobile Keyring är en system som syftar till att tillhandahålla en standard för ett säkrare sätt att logga in på en hemsida, där det finns risk att lösenord kommer i orätta händer när inloggning sker, t.ex. då man använder en dator på internetkafé eller osäkert lan.

Av Alesandro Sanchez, Jesper Sjövall, Martin Sonesson, Rickard Persson och Simon Ivarsson

1 Index

2 Introduktion

Mobile Keyring är en system som syftar till att tillhandahålla en standard för ett säkrare sätt att logga in på en hemsida, där det finns risk att lösenord kommer i orätta händer när inloggning sker, t.ex. då man använder en dator på internetkafé eller likande osäkra platser.

2.1 Bakgrund

I dagens läge är lösenord mycket attraktiva mål för hackare. Många människor har relativt enkla lösenord och återanvänder dessa på flera olika ställen. Dessutom finns det många hemsidor där det finns möjlighet att för enkelhetens skull spara sin kreditkortsinformation för att förenkla framtida köp och på så sett reducera säkerheten på kreditkortet till endast sitt användarnamn och lösenord. Om en hackare kan få tag på någons lösenord till en sådan hemsida kan han eller hon beställa varor och tjänster utan problem och utan användarens samtycke. Även om hemsidan inte har något monetärt värde så kan den istället innehålla känslig information, så som mail-korrespondens, personlig fakta om bostad, telefonnummer, personnummer och dylikt. Trots att den personliga förlusten för en individ vars lösenord läcker ut är stor så är det nästan bara banker som har försökt hitta nya lösningar utöver lösenord för att skydda sina användares tillgångar. Därför behövs det ett enkelt sett för webb-administratörer att implementera extra säkerhet för sina användare, vilket är varför Mobile Keyring har skapats.

2.2 Syfte och Mål

Syftet med projektet är att göra en applikation samt ett protokoll vars ändamål är att tillhandahålla engångslösenord för inloggning på hemsidor.

Mål och delmål:

- Applikationerna - Funktionalitet
  - Skall vara lätt att använda för folk utan någon större datorvana.
  - Skall, om inte användaren specificerat annat, inte lagra riktiga lösenord på telefonen.
  - Skall vara så säker som möjligt och använda kryptering för att säkerställa informationens integritet.
  - Skall finnas någon möjlighet att stänga av applikationens funktionalitet ifall telefonen blir stulen.
- Applikationerna – Programmering
  - Skall ha stöd för enkel implementering av framtida MKP-kommandon.
  - Skall vara modulär för enkel modifiering och övrig utbyggnad.
  - Skall använda sig av HTTPS och XML.
- Mobile Keyring Protocol
  - Skall vara så litet som möjligt.
  - Skall vara enkelt att implementera i en webbserver även för en webbadministratör som inte har tidigare erfarenhet av MKP.
  - Skall enkelt kunna utökas med nya funktioner.

- Projektet
  - Skall sätta en tänkt standard för denna typ av säkerhetslösningar.
  - Skall ha fungerande prototyp tills mitten av projekttiden.

Se även Appendix Kravspecifikation – Mobile Keyring för mer detaljer om vilka funktioner som var tänkt att stödjas från början.

3 Delprojekt

Som en del av Mobile Keyring projektet så har fyra underprojekt skapats som ansvarar för de olika centrala delarna av Mobile Keyring projektet.

De olika underprojekten som har skapats är följande:

- Mobile Keyring Protocol – Hur kommunikationen skall fungera.
- Hamster Keyring – Android appen som fungerar som klient.
- Keyring Manager – PC-programmet som används som komplement till Hamster Keyring.
- Demohemsida – Som vi använder som demo för projektet.

3.1 Mobile Keyring Protocol

En central del av projektet var att ta fram det protokoll som skall användas för kommunikationen mellan webb-servern och klienten (mobilen).
Det första som vi gjorde var att fastlägga vilka krav som fanns på protokollet för att kunna göra det vi önskade.

- Enkelhet – att införa på en webbserver.
- Använder HTTP – protokollet skall vara ett underprotokoll till HyperText Transfer Protocol.
- Skalbarhet – möjlighet att införa nya funktioner.
- Säkerhet – flera olika smarta säkerhetsfunktioner.

Vi bestämde oss för att kalla protokollet för Mobile Keyring Protocol vilket vidare kommer att förkortas som MKP.

3.1.1 Varför dessa krav

I förra stycket så nämns fyra olika krav, enkelhet, skalbarhet, över HTTP och säkerhet.

Orsaken till kravet ”Enkelhet” är att om vi vill att vårt system skall användas i verkligheten så måste implementation på webb-server sidan vara tillräckligt enkel att en vanlig webbadministratör har möjlighet att införa MKP på sin sida.

Ett annat viktigt krav var att MKP kan köras över HTTP eller HTTPS vilket innebär att ingen extern programvara eller behörighet behövs, vilket kan vara ett problem om man har sin hemsida på ett webbhotell och inte har möjligheten att installera andra program.

Ett annat mål med MKP var att göra det skalbar, vilket innebär att det är möjligt att införa mer funktionalitet om så önskas.

Säkerheten är en av grundpelarna eftersom huvudidéen med hela projektet är att erbjuda en bättre säkerhetslösning än vad som ofta användas i dag.

3.1.2 Hur fungerar MKP

Av praktiska skäl så kommer vi ej går in i detalj på hur MKP fungerar utan hänvisar istället till dokumentationen för *Mobile Keyring Protocol 1.1 (Data Protocol 1.0),* vilket är den senaste upplagan av MKP. Istället så kommer vi titta lite på MKP i allmänhet och de kraven vi har.

Vi valde att införa MKP över HTTP eller HTTPS vilket innebär att innehåll kan skickas som en vanlig hemsida. För att skicka data från klienten (mobilen) till webbserven så sker det med hjälp av HTTP POST anrop samtidigt som serven svarar genom och skicka tillbaka XML dokument. Hela protokollet är request-baserat och sessionslöst vilket passar bra då HTTP protokollet också har dessa egenskaper.

Vårt andra mål är skalbarhet, vilket vi har löst genom att låta MKP består av olika ”tjänster” som stöds, där varje tjänst kan göra en specifik sak. Varje tjänst har ett namn och kan blir identifierad med hjälp av en URL. För att kunna lista alla tjänster som en specifik server stödjer så kräver även att MKP har en ”Index” tjänst som kan lista information om vilka tjänster som stöds av serven. Några av tjänsterna som MKP stödjer är:

- Access – skapa en engångsnyckel för att logga in med.
- Deaccess – gör en engångsnyckel ogiltigt att använda.
- Deactive – inaktivera autentisering av mobilen, vilket även kan göras från ett datorprogram.
- Forcelogout – påtvinga utloggning av alla inloggade enheter.

3.1.3 Resultat

Resultatet är enkel och skalbart protokoll som en webbadmiratör kan implementera inom en dag. Om Mobile Keyring Protocol införs på korrekt sätt så kan MKP ge ökad säkerhet vid inloggning och auktorisering än vad som är normalt kan erhållas.

3.2 Android Programmet - Hamster Keyring

Android applikationen, kallad ”Hamster Keyring”, är andra delen av vårt projekt. Medan MKP protokollet är produkten som webbadministratörer kommer att använda så är ”Hamster Keyring” produkten för deras användare.
Målet med appen är att ge dess användare möjligheten att utnyttja all funktionalitet som MKP protokollet har att erbjuda. Användaren kommer att kunna logga in säkert på sina olika konton på nätet enkelt genom att lägga till dem i appen. För detta krävs det att hemsidan stödjer MKP.

3.2.1 Planering

Planeringen påbörjades tillsammans med själva protokollet. Även om vi inte hade helt klart för oss allt som MKP skulle ta hand om, så visste vi att ”Access” (Läs 3.1.2) var huvuduppgiften för appen. Därför beslutade vi att appen skulle ge användaren en överblick över alla sina inlagda konton och lätt kunna få ett engångslösenord. Då appen innehåller känslig data så beslöts det också att det krävs en pinkod varje gång man startar appen.

Målet var att appen skulle vara lätt att hantera för en vanlig användare, därför delade vi vår app i 3 ”skärmar”:

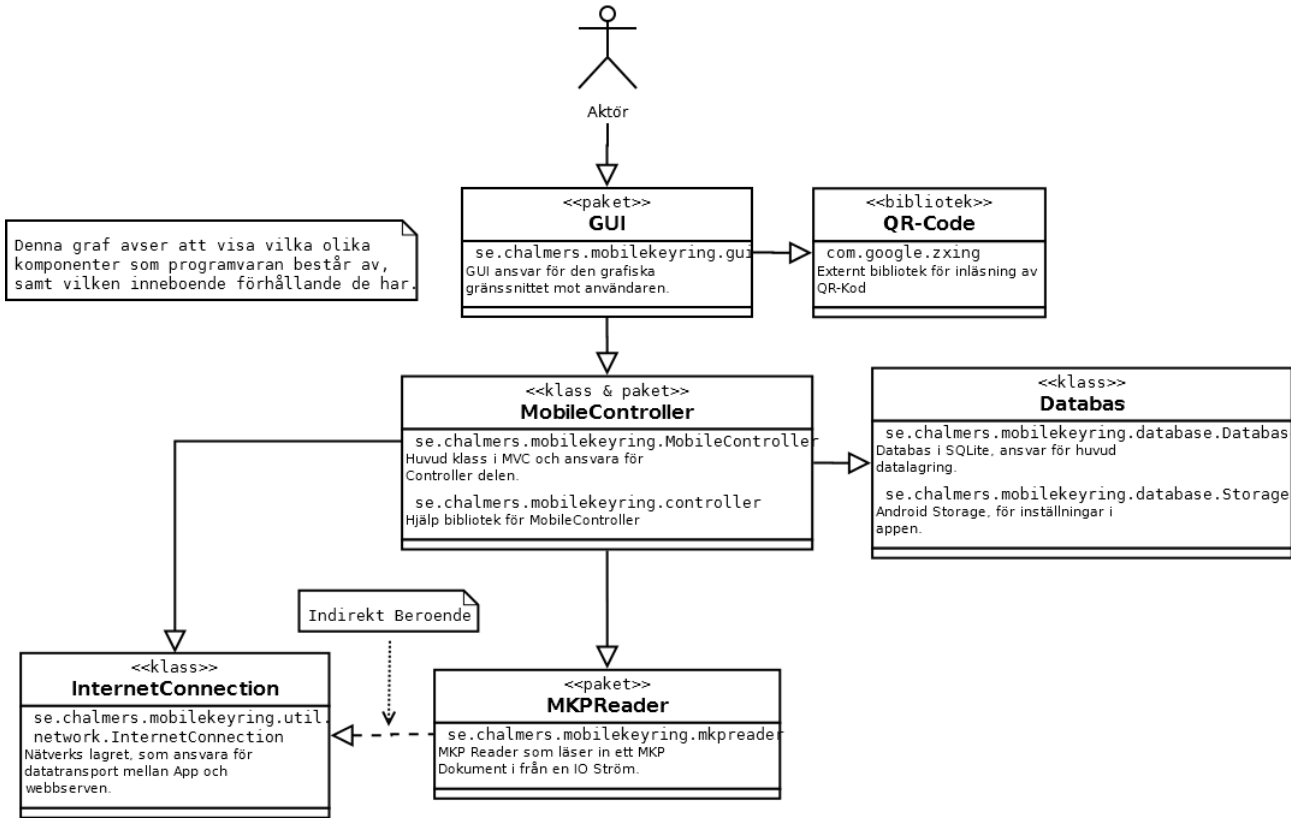
- Pinkodsskärm, Det första användaren ser varje gång han startar appen. Användaren måste mata in sin pin kod för att kunna komma åt appens funktioner. Detta av säkerhetsskäl.
- Överblicksskärm, Efter lyckad inloggning så kan nu användaren se en lista över alla sina inlagda konton. Här kan användaren lägga till fler konton samt skaffa engångslösenord.
- Extrafunktionalitetsskärm, Då det inte var klart i början allt som MKP skulle stödja så bestämdes det att från Överblicksskärmen så skulle man kunna, för varje konto, ta sig till en speciell skärm som innehåller allt annat som stöds av MKP.

Se vidare i användarmanualen för Hamster Keyring om exakt hur Hamster Keyring fungerar, och vilken funktionalitet som finns.

3.2.2 Modelldiagram

Vi har valt och utvecklar Android appen enligt Model-View-Controller modellen efter som MVC har de fördelarna som vi önskade, skalbarhet, överskådlighet, men framför allt möjligheten att dela upp arbetet mellan olika gruppmedlemmar utan onödig komplexitet.
Android appen består av följande moduler:

- MobileController, ansvara för att samordna de olika modulerna i Android appen.
- GUI, är den grafiska gränssnittet mot användaren i telefonen.
- QR-Code, är det biblioteket som användas för att läsa in en QR-kod i programmet.
- Databas, ansvarar för att lagra de tillagda kontona och inställningar som skall sparas i telefonen.
- InternetConnection, hälper appen att hämta innehåll i från webbserven både över HTTP och HTTPS.
- MKPReader, tolkar innehållet från webbserven som en del av Mobile Keyring Protocol standarden.



3.2.3 Skapandet av applikationen

Vi delade upp arbetet mellan oss och var och en var ansvarig för sin del av arbetet. När vi började bli klara med våra delar satte vi ihop dem. Att arbeta på det här sättet hjälpte mycket, till exempel vid kodandet av PC programmet (Läs 3.3) då koden är väldigt modulär. När appen var i princip färdig så skapades olika testfall för att se till att allt stämde överens. En del fel och buggar hittades och fixades.

3.2.4 Resultat

Slutprodukten av detta arbete blev mycket lyckad, då den täcker alla mål som var tänkta från början. Applikationens interface är mycket enkelt för en ny användare att börja använda. Den stödjer allt som MKP har att erbjuda och med hög säkerhet.

Vid starten av applikationen krävs det alltid pinkodinmatning (Se fig 1). Om användaren skulle ha råkat lämna mobilen någonstans eller helt enkelt lånat den till en kompis så kommer det alltid att krävas en pinkod vilket säkrar att ingen får tillgång till den.
När användaren väl loggat in så visas en lista som innehåller alla inlagda konton (Se fig 2). Därifrån så krävs det enbart en knapptryckning för att kunna få ett engångslösenord (Se fig 3) till den valda hemsidan. Detta är en del av appens användarvänlighet då denna är huvudtjänsten som kommer

troligen att vara den mest använda av alla.
Användaren kan också välja att använda resterande funktioner av MKP genom att gå in i ett visst konto och därifrån få en lista över alla MKPs funktioner (Se fig 5).
Det finns också en mängd olika andra alternativ för användaren att anpassa appen, från att lägga till konton till en favoritlista som alltid visas längst upp till att byta ikon eller byta namn på hemsidan för enklare åskådlighet.

Att lägga till konton i appen görs enkelt genom att mata in hemsidans och användarens namn. Det krävs även en kod från hemsidan för att verifiera användaren. Av säkerhetsskäl rekommenderas att koden är en speciell kod från hemsidan som fås på användarens begäran. Den borde också gälla enbart en gång och vara tidsbegränsad. Detta då andra personer skulle kunna skapa en app som använder sig utav MKP och spara användarens riktiga lösenord vilket borde undvikas.
Då koderna borde vara säkra så är det bäst att göra dem riktigt långa. Detta skulle inte gillas av många användare då ingen vill skriva en så lång kod. Därför ger vi användaren möjligheten att använda sig utav QR-kods inläsning (Se fig 4). På detta sätt kan användaren enkelt läsa in en QR-kod (hämtad från den önskade hemsidan) och få allt inmatad enkelt och snabbt.

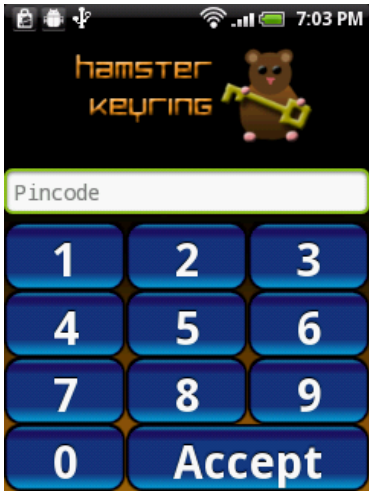


Fig 1

Fig 2

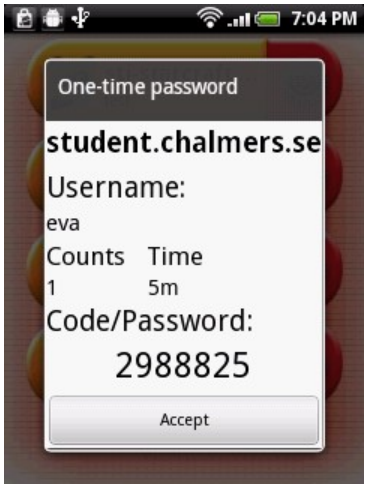


Fig 3

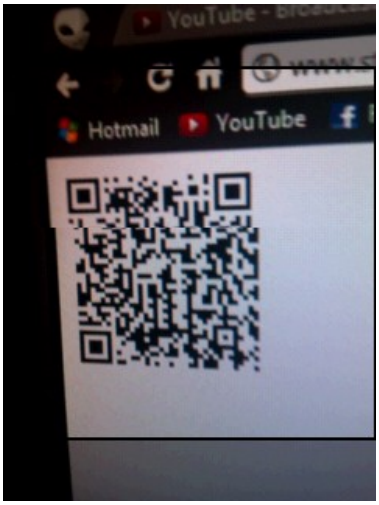


Fig 4

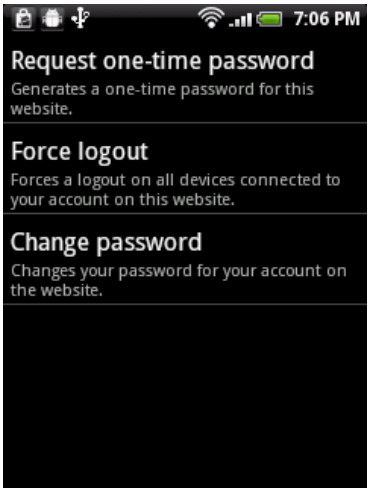


Fig 5

3.3 Datorprogrammet - Keyring Manager

Keyring Manager är ett datorprogram som erbjuder extra funktionalitet till Hamster Keyring. Tanken med programmet är att kunna utföra operationer med hjälp av MKP som inte kräver åtkomst till sin mobiltelefon, vilket kan vara bra ifall användaren för tillfället enbart har tillgång till en dator men inte sin smartphone.

I början hade vi lite svårt att komma överens om vad exakt Keyring Manager skulle kunna göra. Det stod emellan att antingen bara låta Keyring Manager kunna deaktivera aktiva nycklar eller att låta Keyring Manager kunna göra allt som Android appen kan, förutom möjligtvis att generera nya tillfälliga nycklar. Ett annat problem som behövde lösas var hur Keyring Manger skulle få information om olika accounts kopplade till appen.

Vi bestämde oss för att göra så att Keyring Manager enbart erbjuder möjligheten att deaktivera nycklar och kan inte göra andra saker som enbart är tänkta för appen, såsom Force logout, byta lösenord etc. Därmed låter vi appen vara den huvudsakliga programvaran för vårat projekt och Keyring Manager är således inte tänkt att vara någon form av portning till PC eller alternativ till Android appen. Det är istället ett tillägg vars syfte är att kunna deaktivera nycklar utan åtkomst till appen, vilket kan vara användbart i vissa nödlägen.
Keyring Manager kräver att man i förväg har läst in sina olika konton som man har kopplat till appen först. Vi bestämde oss för att göra detta genom att man i Keyring Manager läser in en fil som exporteras från appen, som innehåller all information om alla konton. Dessutom behöver man ett lösenord för att låsa upp filen då den är krypterad när man importerar den för att kunna få åtkomst, så att inte någon annan kan komma över och läsa av filen lika enkelt.

Keyring Manager innehåller vissa delar appen, såsom en liknande SQL-lite databas och klasser för att ansluta mot internet. Keyring Manager är helt och hållet skrivet i Java och har ett grafiskt användargränssnitt som använder sig av både AWT och Swing bibliotek.

3.3.1 Hur programmet fungerar

För att använda programmet använder du dig bara av följande anvisningar.

Det första du behöver är såklart att ha Android appen installerad på din smartphone så att du kan exportera. Du behöver förstås också ha lagt till minst ett konto för annars finns det inget att exportera. I Android appens huvudmeny, där man ser sina konton listade, trycker man på telefonens Meny-knapp och väljer sen ”Exportera”. En fil skapas nu på mobilens minneskort.

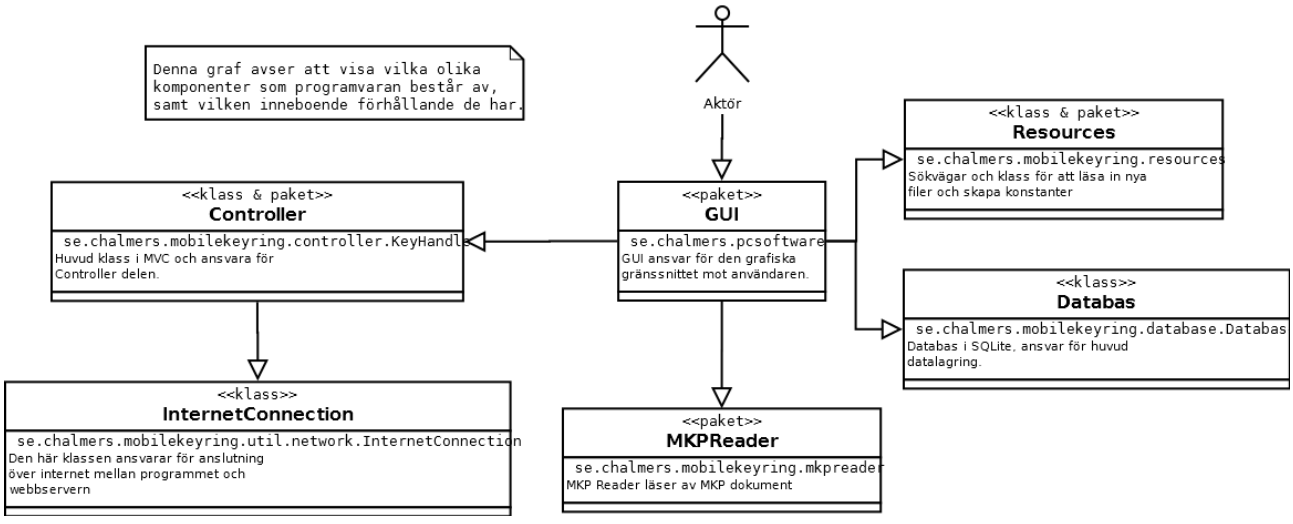
Koppla in din mobil till datorn och välj att läsa av som diskenhet. På minneskortet bör det finnas en mapp som heter HamsterKeyring och där hittar du databas-filen. Överför filen till din dator och starta sedan Keyring Manger.

I Keyring Manager väljer du sen ”Import” och öppnar databasfilen. Alla konton kopplade till filen läses nu in och bör synas på programmets huvudpanel.

Härifrån kan du nu deaktivera konton. Det är bara att markera de konton du vill dekativera och sedan trycka på ”Deactivate”. Ha tålamod, ifall det är många konton du deaktiverar på en gångkan det ta lite tid eftersom programmet måste skapa en internetanslutning till varje server och skicka begäran om deaktivate.

3.3.2 Modelldiagram

Vi har valt att utveckla Keyring Manager enligt Model-View-Controller modellen efter som MVC har de fördelarna som vi önskade, skalbarhet, överskådligt, men framför allt möjligheten att dela upp arbetet mellan olika gruppmedlemmar utan onödig komplexitet.
Keyring Manager består av följande moduler:



3.4 Demohemsida

Som en del av projektet så ingick det att skapa en demohemsida som kan användas i demonstrationssyfte för Mobile Keyring projektet.
Vi har valt att utveckla demosidan i PHP efter som PHP är ett av de webbprogrammeringsspråken som två medlemmar i gruppen kan och för dess enkelhet att utveckla webbsidor med, till skillnad mot t.ex. java som vi hade i kursen webbprogrammering. En annan fördel med PHP är att det är enkelt att sätta upp och köra då många servrar använder PHP.

Vår hemsida (hemsidor) är uppbyggt av tre olika moduler, Databasen, MKP Implementationen och front-end delen som visas för användaren som besöker sidan.

3.4.1 Front-End

Vi har valt att utveckla två olika Front-end som vi kommer att köra på två olika servrar.
Syfte med detta är att kunna visa Mobile Keyring fungerar på flera olika sidor parallellt och det är möjligt att har samma app kopplad till flera olika hemsidor samtidigt.

3.4.2 PHP Databasen

För enkelhetens skull och med tanke på detta enbart är en demosida så valde vi att utveckla en enkel databas i PHP som sparar data till en fil på serven i stället för att använda en riktigt databashanterare så som tex MySQL vilket skulle innebära betydligt mycket mer jobb för oss förhållande till vad som är tillrådligt för en demonstrationssida.

Även om vår databas inte är en riktig relationsdatabas så simulerar den en enkel sådan för att visa hur det skulle kunna implementeras i verkligheten.

Vår databastabell se ut enligt följande:

User	Passwd	Laccess	Mkpen	Authkey	Keyval	Keytime	Keycount	Message
Kalle	a2d4c86ae5d21177ee1cbfb11cf3c4c1	0	False	″″	″″	0	0	My password is password 123
Sara	5347d239db3cf008fbd6af1d744ff9c1	1329502625	True	7f77f0438bfef9ab86a706a0e953a234	a72df0742c9d67bb611910bc62b06f39	1361016322	1	I use MKP

Som ni ser i ovan databastabell så är Passwd, Authkey och Keyval hashade för att säkerställa lösenorden även om databasen blir läst av någon obehörig. Vi har valt och hasha alla lösenorden enligt följande logik username+@+passwd. Orsaken till varför vi använder användarnamnet som en del i hashen är förhindra uppslagsattacker mot lösenorden, vilket är en vanlig metod när man önskar knäcka krypterade lösenord.

3.4.3 Implementationen av MKP

Den sista delen av hemsidan är implementationen av MKP och den delen som pratar med den mobila enheten som kan användas för att autentisera sig.
Här har vi valt att konfigurera den ena serven med den minimala upplagan av MKP där man i praktiken enbart kan skapa en engångskod för att auktorisera sig med, medan den andra serven har vi konfigurerat med den fullständiga MKP där man både kan skapa engångskoder som är giltigt längre tid samt flera gånger, påtvinga utloggning, byta lösenord mm.

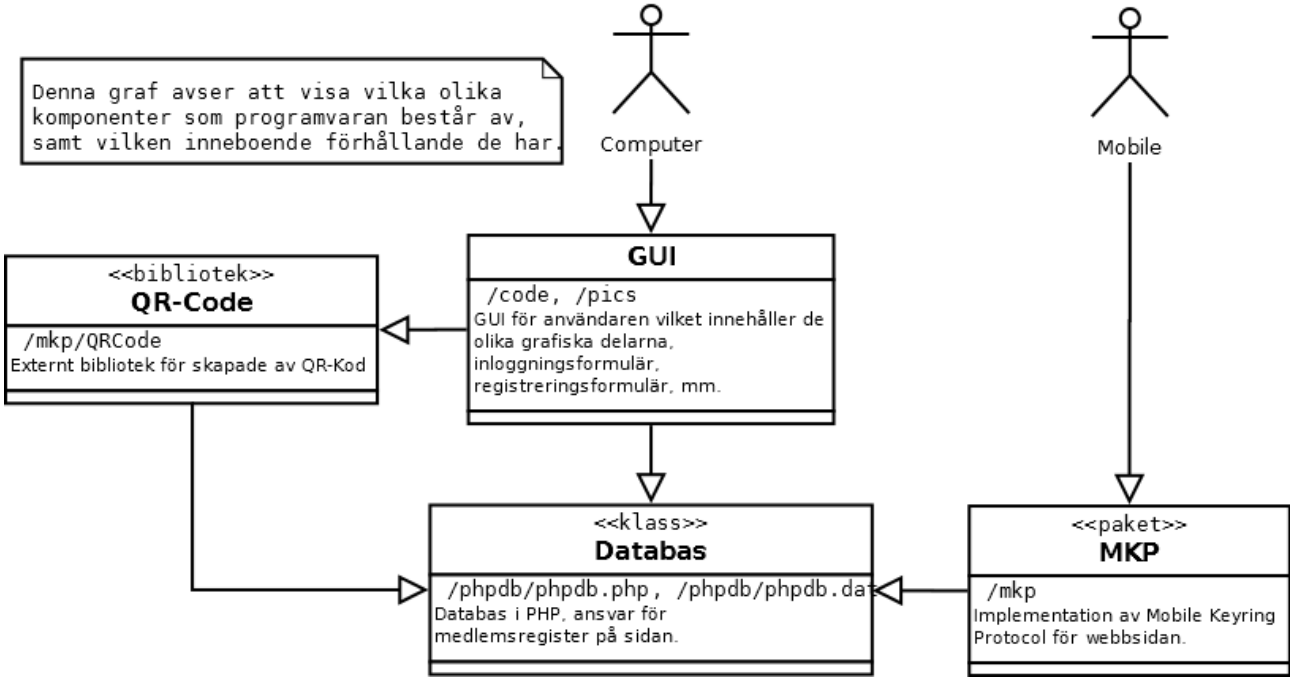
En giltigt QR-Kod för MKP

Den andra serven stödjer även QR-kods auktorisering av den mobila enheten som går ut på att man i telefonen läser in en QR-kod, vilket underlätta för användare som då inte behöver skriva in sidans URL, användarnamn och lösenord på telefonen.

3.4.4 Modelldiagram

Vi har valt att utveckla Keyring Manager enligt Model-View-Controller modellen eftersom MVC har de fördelarna som vi önskade, skalbarhet, överskådligt, men framför allt möjligheten att dela upp arbetet mellan olika gruppmedlemmar utan onödig komplexitet.
Hemsidan består av följande moduler:

- GUI, ansvarar för den grafiska delen som en användare ser när han besöker sidan via webbläsaren.
- QR-Code, är till för att generera QR-koden som appen använder för att lägga till ett konto.
- MKP, är interfacet mot vår telefon som använder sig av Mobile Keyring Protocol för att utbyta information mellan webbserven och telefonen.



4 Genomförande

Planeringen för projektet började redan innan den egentliga terminsstarten då projektidén tog form och gruppmedlemmar valdes, genom detta var det möjligt att påbörja arbetet direkt. Den första delen av projektet ägnades åt att definiera protokollet som skulle användas vid kommunikationen mellan mobilen och webbplatsen, hela den första veckan ägnades åt denna designfas.

Den första veckan ägnades också åt arbetsfördelning vilken baserade sig på tidigare erfarenheter samt intressen vilket medförde följande; Alesandro ägnade sig åt appens gui, Jesper med php, Martin webb-design för demosidan, Rickard http/https kommunikation för appen och Simon XML inläsning. Jesper ägnade sig även åt mötessamordning.

När protokollet och de olika modulerna för applikationen var specificerade påbörjades programmeringen, under de första veckorna skedde detta utan ett versionsystem. Trots detta fortgick projektet tack vare modulariseringen. I slutet av arbetsvecka tre fick vi tillgång till en svn-server på skolan vilket underlättade distributionen av kod.

Genom projektet fortskridande började de olika delarna bli klara, även om mycket debuggning återstod. Då mycket tid återstod för kursen beslutades det att applikationen även skulle kunna läsa in QR-koder, vilket medförde att protokollet behövde kompletteras med detaljerad information för detta, samt att ytterligare en modul behövde skapas. De som framförallt blev involverade i detta tillskott var Alesdandro och Jesper.

Även andra gruppmedlemmar började arbeta med andra moduler än de initiala detta medförde att Rickard började arbeta med databaser för Android appen samt pc programmet, Simon med kontrollenheten som hantera kommunikation mellan de olika modulerna och Martin med gui:et för PC applikationen.

När februari närmade sig sitt slut var Android appen till stora delar klar, även om mindre buggfixande återstod. Genom fokuset på den mobila programvaran hade pc delen blivit lidande vilket medförde att den resterande tiden av kursen ägnades åt detta, som tidigare blev Rickard ansvarig för http/https implementeringen. Även Alesandro blev involverad i pc-appen genom att han arbetade med sammankopplingen av de olika modulerna i programmet.

Slutligen i början av mars vad mycket av koden färdigskriven och många buggarna var lösta, vilket gjorde att framförallt Alesandro skrev ett flertal testfall för android appen, detta framförallt för att han hade skapat gui'et för vilket testfallen gällde. Vid ungefär samma tidpunkt färdigställde gruppen som helhet denna rapport.

Kontinuerligt under projektet gång anordnades det möten för att lösa fel och problem som uppkommit, överse arbetsflödet för de olika modulerna samt andra reflektioner angående projektet. Under designfasen skedde det tre till fyra möten per vecka, när arbete övergick detta till ungefär en eller två och i den slutliga fasen av projektet då mycket var avklarat var antal möten per vecka nere i ett.

5 Slutsats

Den första delen av projektet efter val av inriktning var att designa protokollet som utgör grunden för kommunikationen mellan webbservern och mobiltelefonen. Vi utgick från http för design, men strukturerade enligt xml för underlätta förståelsen av meddelanden. Eftersom ingen av oss tidigare haft erfarenhet av detta medförde detta en ny lärdom, problematiken innefattade att få information entydig och strukturerad samtidigt som alla egenskaperna innefattades. Vår lösning var dela upp problemet i mindre bitar som var specifika för en viss del vilket underlättade och gjorde protokollet mer tydligt. Givet att många andra protokoll har behövt revideras med tiden är det troligt att detta även är fallet för vårt protokoll.

Under utvecklingsfasen var ett utav de mest komplicerade problemen QR-koden, hur anpassas en generell modul för sträckkoder till att endast läsa aktuell kod. Svårigheten ligger delvis i att det inte finns en exakt lösning för vårat problem, utan att det finns en generell lösning för QR-koder som sedan måste anpassas för vårt aktuella problem. Detta förenklas ej genom att det har saknats detaljkunskap om det specifika problemet, vilket har testat gruppens förmåga att under kort tid sätta sig in i ett problem.

Genom den tidigare kursen i databaser har vi delgivits kunskaper i lämpliga tillvägagångsätt för att minimera återupprepning av data i tabeller. Denna kunskap till trots har vi valt att utforma tabellerna utifrån logisk struktur. Detta har medför att det finns överflöd i tabellerna men eftersom mängden data som ska sparas är begränsas ansåg vi det olämpligt att designa en komplex databas för aktuell data. Erfarenheten som gavs av denna del av projektet är att anpassa databasdesignen efter behovet.

6 Appendix

Kravspecifikation - Mobile Keyring v1.0

Kravspecifikation för Mobile Keyring projektet och en del av förarbetet till Mobile Keyring.

Mobile Keyring Protocol 1.0

Innehåller fullständigt dokumentation för Mobile Keyring Protocol 1.0.

Mobile Keyring Protocol 1.1 (Data Protocol 1.0)

Innehåller fullständigt dokumentation för Mobile Keyring Protocol 1.1 DP 1.0, vilket ersätter tidigare versioner av Mobile Keyring Protocol.

Tiny Shuffle Algorithm

Beskriver algoritmen Tiny Shuffle Algorithm vilket är en del av Mobile Keyring Protocol.

Användarmanual för Hamster Keyring

Användarmanual för Android appen Hamster Keyring.

Testcase för Hamster Keyring

Testcase för Android appen Hamster Keyring.