

Merja Paaso

Child Care Help and Friend Family Web Service Design

Helsinki Metropolia University of Applied Sciences

Master of Engineering

Degree Programme

Thesis

15 February 2015

| | |
|--|---|
| Author(s) Title Number of Pages Date | Merja Paaso Child Care Help and Friend Family Web Service Design 71 pages 15 February 2015 |
| Degree | Master of Engineering |
| Degree Programme | Information Technology |
| Specialisation option | Multimedia Communications |
| Instructor(s) | Kari Järvi, Principal Lecturer |
| <p>The purpose of this thesis was to design a service where families, which need occasional child care help and families which can give occasional child care help, could meet. A further aim of the service is to help families meet other families as friends. The service was named FamilyLibrary and that name is used throughout this thesis.</p> <p>This study is based on a feature and database design of the FamilyLibrary service and on testing how Microsoft SQL Server 2014 works on top of Windows Server 2012 R2. Testing the environment was carried out by creating a database project and the tables needed with Microsoft Visual Studio 2013. The designs of the features are explained in words and also graphically. The database is designed to be compatible with the features.</p> <p>This thesis was successful in creating the design for the FamilyLibrary service. As the scope of this study did not include the implementation of the service, it has been left for a separate project. Almost all software will need modifications during its lifetime. This FamilyLibrary service is not an exception. The FamilyLibrary service may need some new features or corrections for the version to be implemented. It is worth noting that the implementation of the FamilyLibrary service will not be a small case, even if only the features designed in this thesis are implemented.</p> | |
| Keywords | FamilyLibrary, database, SQL Server 2014, Windows Server 2012, Visual Studio 2013 |

Contents

List of Abbreviations

| | | |
|-------|--|----|
| 1 | Introduction | 1 |
| 1.1 | Writer | 1 |
| 1.2 | Problem | 1 |
| 1.3 | FamilyLibrary Service | 2 |
| 1.4 | The Target of the Thesis | 3 |
| 2 | Basics and Used Tools | 4 |
| 2.1 | Basis of Relational Databases | 4 |
| 2.1.1 | On-premises Database | 5 |
| 2.1.2 | Database on Cloud | 5 |
| 2.2 | Basic Introduction of Used Tools | 6 |
| 2.2.1 | Windows Server 2012 R2 | 6 |
| 2.2.2 | SQL Server 2014 Enterprise 64b | 7 |
| 2.2.3 | Introduction of SQL | 7 |
| 2.2.4 | Microsoft Visual Studio 2013 | 7 |
| 2.2.5 | Microsoft SQL Server 2014 Express with Advanced Services | 8 |
| 2.2.6 | Microsoft Azure and SQL Database | 8 |
| 2.3 | FamilyLibrary Database | 9 |
| 2.4 | Personal Data Act | 10 |
| 3 | Designing and Implementation of FamilyLibrary | 11 |
| 3.1 | Installing Applications Needed | 11 |
| 3.2 | Designing FamilyLibrary Service | 11 |
| 3.2.1 | Environment | 11 |
| 3.3 | FamilyLibrary Service Features | 14 |
| 3.3.1 | Registration | 14 |

| | | |
|--------|--|----|
| 3.3.2 | Identification | 16 |
| 3.3.3 | Personal Data Updating | 17 |
| 3.3.4 | Login with an E-mail Address and Password | 18 |
| 3.3.5 | Query for a Forgotten Password | 20 |
| 3.3.6 | Adding to Calendar Possible Times to Give Child Care Help | 23 |
| 3.3.7 | Search Tool for Seeking Child Care Help | 24 |
| 3.3.8 | Search Tool for Seeking Friend Families | 26 |
| 3.3.9 | Recommendation Collection | 28 |
| 3.3.10 | Stopping the Service Use | 29 |
| 3.3.11 | Logging Out | 29 |
| 3.3.12 | Activity Timer | 31 |
| 3.4 | Designing of the FamilyLibrary Database | 31 |
| 3.5 | Creating Database Project and Tables with Visual Studio | 34 |
| 3.5.1 | User Table | 37 |
| 3.5.2 | Calendar Table | 38 |
| 3.5.3 | Children Table | 38 |
| 3.5.4 | Children_to_take_care Table | 39 |
| 3.6 | Creating Database and Tables with Microsoft SQL Server Management Studio | 40 |
| 3.7 | Implementation of the FamilyLibrary Service | 46 |
| | Summary | 47 |
| | References | 48 |
| | Appendices | |
| | Appendix 1. Basic SQL Commands | |
| | Appendix 2. Common SQL Datatypes in Microsoft SQL Server | |
| | Appendix 3. Contents of Tables | |

List of Abbreviations

| | |
|--------|---|
| ANSI | American National Standards Institute |
| DBA | Database Administrator |
| DBMS | Database management system. Software for maintaining, querying and updating data and metadata in a database. |
| DevOps | Combination on two words: development and operations |
| IaaS | Infrastructure-as-a-Service |
| NSN | Nokia Solutions and Networks. Network company of Nokia |
| ORM | Object-relational mapping. The set of rules for transferring the data stored in instance variables of an object-oriented programming language into tables of a relational database, and vice versa. |
| PaaS | Platform-as-a-Service |
| SaaS | Software-as-a-Service |
| SLA | Service Level Agreement |
| SQL | Structured Query Language. |

1 Introduction

The topic of this thesis is to design a so called FamilyLibrary service. The reason why this kind of service was interesting for the writer is that the writer was a housewife during the time when the thesis was written and it was sometimes hard to find child care help if that was temporarily needed. Thus, the topic was chosen out of a real need.

1.1 Writer

This thesis and the project it describes is the writer's own project. The writer is a 37-year old woman, who was on maternity leave and then on study leave during the writing of the thesis. The writer's employer was NSN (Nokia Solutions and Networks) when the maternity leave begun, but it has changed to Nokia because Nokia mobile phones was sold to Microsoft. NSN became the main business area of Nokia. The writer started to work for Nokia in 2004 and moved to NSN in 2008. The educational background of the writer is in Information Technology with a Bachelor's degree in engineering from Rovaniemi Polytechnic acquired in the year 2000.

1.2 Problem

There is no good and moderately priced service from where to get child care help when it is temporarily needed. Many families are nowadays quite lonely also and they need other families to be a friend family for them. Thus, this thesis aims to answer the following two questions:

1. What kind of service is needed?
2. How should the service be implemented?

1.3 FamilyLibrary Service

The main objective of the thesis is to design a so called FamilyLibrary service for families which will occasionally need some short term child care help. From the service, families can also find other families to act as a friend family.

The FamilyLibrary service needs a database where all the needed information of families, which are registered to the service, can be saved. Windows Server 2012 R2 and SQL Server 2014 Enterprise 64b are used for the database part of this thesis.

Many families find themselves faced with this problem at least sometimes. The mother and father have, for instance, some unexpected errands to run. They then have to find some child care help for their little children. Nowadays it is a very common situation that grandparents and other relatives live far away and maybe the family's close adult friends are busy with their own lives. If you have to buy child care service it might cost a lot and it might also be impossible to find a last-minute child care service. The FamilyLibrary service is a meeting place for families, which need some child care help and for families which are sometimes able to provide some child care help.

There are lots of tired mothers also, who sometimes need some time to rest a bit. It is very common these days that families have no support for everyday life nearby. Especially mothers can get tired and cast down. If there were families that could take care of the children of other families, then tired parents could have some time to themselves.

Sometimes families only need some other family that is in the same life situation to act as a friend family.

When we talk about kids we talk about the most important things for moms and dads. That's why it is very important that the place, which offers child care help, is a safe place. All registered persons have to be fully identified. It is important that all registered people can be trusted.

There might be four different kinds of users in the FamilyLibrary service:

- Those who give child care help

- Those who need child care help
- Those who give and need child care help
- Those who need friend families

The following information is stored in the Family table which is used by FamilyLibrary.

- Person1 name and personal identity number
- Person2 name
- Children
- Pets
- Allergies
- Children_to_take_care
- Address
- Phone_number
- E-mail_address
- Child_care_times
- Recommendations

1.4 The Target of the Thesis

The target of the thesis is to design a service where it is easy to find child care help or friend families. The service is called FamilyLibrary in this thesis. The idea for the name FamilyLibrary came from the thought that families can “borrow” other families when child care help and/or friendship is needed. When a FamilyLibrary service design is ready a

real software can be implemented. The outcome of the thesis is definitions and a specification of what that service should include.

A deeper look describes the database side of the FamilyLibrary service. The needed database tables will also be designed as the outcome of this thesis.

Another target of the thesis is to get familiar with a Windows Server 2012 R2 and SQL Server 2014 Enterprise 64b and Microsoft Visual Studio 2013 during the FamilyLibrary service designing. The Windows Server 2012 R2 and SQL Server 2014 Enterprise 64b will be installed and those will be tested by creating a test database project and tables. The database project and needed tables will be created by Microsoft Visual Studio 2013.

Implementing the service is not the target of this thesis. The target is to design the FamilyLibrary service and leave the implementation of the service for a future project. The design must be so clear that it is easy to start the next phase of the project.

2 Basics and Used Tools

In this section there are some basics of the concepts and tools which are used in this thesis.

2.1 Basis of Relational Databases

Almost all used databases nowadays are relational databases. The relational database is a set of tables. Each table (sometimes called a relation) contains one or more categories in columns. Each row on the table contains unique instance of data for the categories defined by the columns.

In this thesis there is for example a User table. The data of every family which uses the FamilyLibrary service is saved to the User table. Every individual family have their own row on the User Table where their data is saved. There is specific data which is needed to collect from every family and that data constructs the columns of the User table. The columns of the User table are for example, name, personal identity number and address.

The relational database is in many case the set of tables (relations). There are then relations between the tables.

In this thesis, for example, there are four tables used. When the family registers to the FamilyLibrary service it gets its own individual User_Id number which is used to identify the family on every table used. In this thesis there are tables for

- collecting data of the family
- collecting data of the children of the family
- collecting data of the children that need care taking
- creating a calendar which shows the times when child care help is offered.

2.1.1 On-premises Database

When a database is on-premises, it means that a company or some other constant have their own server where the used database is located. The test database, which is implemented in this thesis, is an on-premises database. Still, in many cases, it will be cleverer to use a cloud system when databases are created. In that case own servers are not needed.

2.1.2 Database on Cloud

This thesis is concentrated on a Microsoft environment, so also the cloud system introduced is served by Microsoft. This cloud system is Microsoft Azure.

What is cloud and cloud service? A public cloud is a group of data centers which are offered to all customers. Those data centers are used over the network with monthly payment. An application which is offered as a web service and is running on the cloud is called a cloud service. Usually these kinds of services are known as a Software-as-a-Service (SaaS). The cloud which is offered to be the platform to the services is called Platform-as-a-Service (PaaS). The resources of the cloud which are served as an IT infrastructure of the company are called Infrastructure-as-a-Service (IaaS). [6]

2.2 Basic Introduction of Used Tools

The following Microsoft applications are used in this thesis: Microsoft Windows Server 2012 R2, Microsoft SQL Server 2014 Enterprise 64b and Microsoft Visual Studio 2013. These applications are installed to an HP Compaq desktop computer. The desktop computer has the following features:

- Processor: Inter® Core™2 Duo CPU E8400 @ 3.00GHz 3.00 Ghz
- Installed memory (RAM): 16.0 GB (15.9 usable)
- System type: 64-bit Operating System, x64-based processor
- Pen and Touch: No Pen on Touch Input is available for this computer

The report itself is written with Acer laptop computer with Microsoft Word 365. The laptop has the following features:

- Processor: AMD A6- 1450 APU with Radeon(YM) HD Graphics 1.00 GHz
- Installed memory (RAM): 6,00 Gt (5,77 Gt usable)
- System type: 64-bit Operating System, x64-based processor
- Pen and Touch: Full Windows Touch support with 10 touch points

2.2.1 Windows Server 2012 R2

A server is meant to be always running, because it is a computer which is meant to accept requests from the client and give responses to it accordingly. The server can run in any computer but often those are referred to as the server. Servers can perform some tasks on behalf of the clients. For example the server can share data, information or any hardware and software resources. In this thesis the server will save user information which is used on the FamilyLibrary service. In this thesis the server acts as a database server. Also the FamilyLibrary itself is used from the server. In this case it can be said that the server is an application server.

According to the datasheet on Windows Server 2012 R2, it “offers businesses an enterprise-class, multi-tenant datacenter and cloud infrastructure that simplifies the deployment of IT services. With Windows Server 2012 R2, you can achieve affordable, multi-node business continuity scenarios with high service uptime and at-scale disaster recovery. As an open application and web platform, Windows Server 2012 R2 helps you build,

deploy, and scale modern applications and high-density websites for the datacenter and the cloud. Windows Server 2012 R2 also enables IT to empower users by providing them with flexible, policy-based resources while protecting corporate information. Whether your company is a large enterprise, a service provider, or a small or medium-sized business, Windows Server 2012 R2 can help you cloud optimize your business.” [2]

2.2.2 SQL Server 2014 Enterprise 64b

A database server is a computer program that provides database services to other computer programs or computers. In this thesis the computer program used is SQL Server 2014 Enterprise 64b. IT can be said that SQL Server 2014 Enterprise 64b is a database management system (DBMS).

Most of the database servers work with the base of query language. That is the case also with SQL Server 2014 Enterprise 64b. SQL (Structured Query Language) query language is almost the same on every relational database server. Some differences can exist and those can be studied from the manuals of the database server used.

2.2.3 Introduction of SQL

SQL (Structured Query Language) is used to communicate with the database. SQL is, according to ANSI (American National Standards Institute), the standard language for relational database management systems. SQL statements are used to perform tasks. With SQL statements, data saved to the database can be updated or data can be retrieved from the database.

In this thesis Microsoft SQL Server 2014 Enterprise 64b is used along with SQL language. Basic SQL commands can be found in Appendix 1. Common SQL Datatypes in Microsoft SQL Server can be seen in Appendix 2.

2.2.4 Microsoft Visual Studio 2013

Database tables of the FamilyLibrary itself are made with Microsoft Visual Studio 2013. Visual Studio 2013 offers tools for creating tables to SQL Server and testing the created database project / tables for example with Visual Basic.

The available updates for Visual Studio 2013 must be done in order for the SQL Server 2014 to work with the SQL Editor.

2.2.5 Microsoft SQL Server 2014 Express with Advanced Services

Microsoft SQL Server 2014 Express with Advances Services is tested in the thesis as well. With SQL Server 2014 Express it is possible to create an on-premises database system, a cloud database system or a database system which uses both of them.

The download page states that “Microsoft SQL Server 2014 Express is a free, feature-rich edition of SQL Server that is ideal for learning, developing, powering desktop, web & small server applications, and for redistribution by ISVs. The SQL Server 2014 Express release includes the full version of SQL Server 2014 Management Studio instead of SQL Server 2014 Management Studio Express.” [5]

Microsoft SQL Server 2014 Express can be downloaded from <http://msdn.microsoft.com/en-us/evalcenter/dn434042.aspx>.

2.2.6 Microsoft Azure and SQL Database

Microsoft Azure is not used in this thesis but it is introduced because cloud systems are a modern way to handle databases.

Microsoft’s cloud platform, called Azure, has a growing collection of integrated services including compute, storage, data, networking and app. Thanks to this, you can move faster, do more and save money.

Azure offers SQL database services. This is called database-as-a-service. These are the features which Azure offers for SQL Server users:

- Scalable to thousands of databases
- Predictable performance you can dial up or down
- Availability-backed by replicas & uptime SLA (Service Level Agreement)
- Data protection via auditing, restore & geo-replication

- Programmatic DBA-like (Database Administrator) functionality for efficient DevOps
- Self-managed for near-zero maintenance

[8]

The SQL database can be built by using Microsoft Azure instead of creating the on-premises database system.

2.3 FamilyLibrary Database

In this thesis the FamilyLibrary database is created for testing.

The idea of a test database is to create a database for a system from where child care help or family friendship contacts can be requested.

The test database collects information of families which use the FamilyLibrary service. This information will be collected:

- User ID
- Name(s) of adult member(s) of the family
- Personal identity number of the adult member of the family who registers the family to the service.
- Address
- Phone number
- E-mail address
- Password
- Pets in the family
- Allergies in the family
- Is there any children in the family and if yes, how many
- Times when child care help can be given or needed
- Age of every individual child of the family
- Gender of every individual child of the family

- Allergies of every individual child of the family
- Amount of the children needing care taking
- Age range of the children needing care taking
- Gender of the children needing care taking
- Allergies of the children needing care taking
- Recommendations

2.4 Personal Data Act

When the FamilyLibrary is in use it collects personal data from families and that is handled automatically. In Finland a personal data act is applied on that kind of cases.

Personal data act 22.4.1999/523 Chapter 1 2§ says:

“This Act applies to the automatic processing of personal data. It applies also to other processing of personal data where the data constitute or are intended to constitute a personal data file or a part thereof” [3]

When the FamilyLibrary service collects the personal data from families it has to be sure that the data collector has the right to save that data to the database. It has to make it clear to the families that when they register their family as a user of the FamilyLibrary service the service can use the collected personal data automatically. It has to make it clear to the families that when they register to the FamilyLibrary, any data they give, will be saved to the database for later use. When families register in the service, they will give a permission to collect and use any data they gave.

The processing of sensitive data is prohibited and any sensitive data is not needed in the FamilyLibrary service. One example of sensitive data is state of health. The FamilyLibrary service will be based on openness and honesty. For example if some child has some special needs, families can discuss with each other, whether some family could give that kind of child care help, which the child with special needs would need. And of course it is always nice to meet before leaving children to a foreign place. The FamilyLibrary service has no need to collect unnecessarily detailed data of families.

Personal data act 22.4.1999/523 Chapter 3 13§ says: “A personal identity number may be processed on the unambiguous consent of the data subject or where so provided in an Act. A personal identity number may also be processed if it is necessary to unambiguously identify the data subject.” [3]

The FamilyLibrary service collects the personal identity number because of openness and safety. The personal identity number specifies the adult members of the family exactly. There is no need to identify children with the same accuracy. Adults are people who have to take the responsibility for children. When the family registers to the FamilyLibrary service they have to understand that the personal identity number will be collected and the service will automatically use that data.

3 Designing and Implementation of FamilyLibrary

3.1 Installing Applications Needed

Microsoft Windows Server 2012 R2, Microsoft SQL Server 2014 Enterprise 64b, Microsoft Visual Studio 2013 and Microsoft Visio were loaded from mcp.metropolia.fi from where students of Metropolia University of Applied Sciences can load Microsoft programs for free.

The programs were installed by using installation wizards and Microsoft Windows Server 2012 R2 was installed with a graphical user interface.

3.2 Designing FamilyLibrary Service

This section introduces the parts, which are included in the FamilyLibrary service.

3.2.1 Environment

It has to be possible to use the FamilyLibrary Service with several tools. That is why it must be implemented for several environments. It will be easiest to implement a web application which works with a web browser as a first version. Still, multimedia phones

are so common nowadays that a phone application will also be needed for at least the following:

- Windows phone
- iOS
- Android

Features

The features which are needed in the FamilyLibrary service are introduced in this chapter.

Registration

A user has to register in to the FamilyLibrary service before they can use it. The user has to give the required information of her/himself and that data will be saved to a database. User identification will also be done during the registration. Identification is done only once during the registration and after the user is registered a password is used for logging in to the service.

Identification

When we talk about children, strong identification of the adults is very important. A bank account identification can be used.

Personal Data Updating

It must be possible for the user to update her/his data. Families could move, names could change, more children can be born, children get older, etc. When a personal data is updated it will need strong identification again. The service must have absolute certainty that the person updating the data is the correct person.

Login with E-mail Address and Password

After the registration is done and the user is identified then a user specified password and e-mail address can be used for logging in. An e-mail address is used for logging in

because every registered family has their own e-mail address. Naturally, the password needs to be strong enough.

Query for Forgotten Password

If the user has forgotten her/his password it can be sent to her/him via e-mail. The second option is that the FamilyLibrary service resets the password and sends that reset password to the users via e-mail.

Adding to Calendar Possible Times to Give Child Care Help

When child care help will be given then suitable times will be added to calendar. Those times will be saved to the Calendar table of the database.

Search Tool for Seeking Child Care Help

When child care help is needed the user can search for it by using a search tool of the FamilyLibrary Service. The search tool has to give choices by using given parameters: time, child amount, child age, child gender and location. The user has to choose when child care help is needed and how far from home the location can be. The search tool can find the rest of the data needed from the database.

Search Tool for Seeking Friend Families

When a family wishes to find friend families which are in the same life situation as themselves the friend family search tool will be used. This search tool will use some parameters for seeking: children's age range and location.

Recommendation Collection

When a family has used child care help from another family, the family can give recommendations. Recommendations are collected to the database.

Stopping the Service Use

When a family wants to stop the use of the FamilyLibrary service, all data concerning that family will be removed from the database.

Logging Out

There must be a possibility to log out from the FamilyLibrary service.

Recommendation Collection

Families can give positive or negative recommendations to other families.

3.3 FamilyLibrary Service Features

In this section the FamilyLibrary features are introduced and designed more deeply.

3.3.1 Registration

The registration procedure is designed in this chapter.

Figure 1 on the next page shows how a successful registration is done. At first a user has to be reliably identified. This identification can be done by using a bank account identification. When the user is identified she/he can start to fill in a registration form. The user has to write information of the family, children and if child care help is planned to be given, then information about visitor children must be added, as well. More detailed information about all the needed information fields can be found from the database design section.

When the registration form is complete the user sends it to the FamilyLibrary service. The FamilyLibrary service checks the data and if everything is right the data is saved to a database. The user can choose her/his password during the registration and the FamilyLibrary service will check if it's strong enough. When the data is successfully saved to the database the FamilyLibrary service informs the user that the registration has been successful. Now the user is ready for logging in to the service.

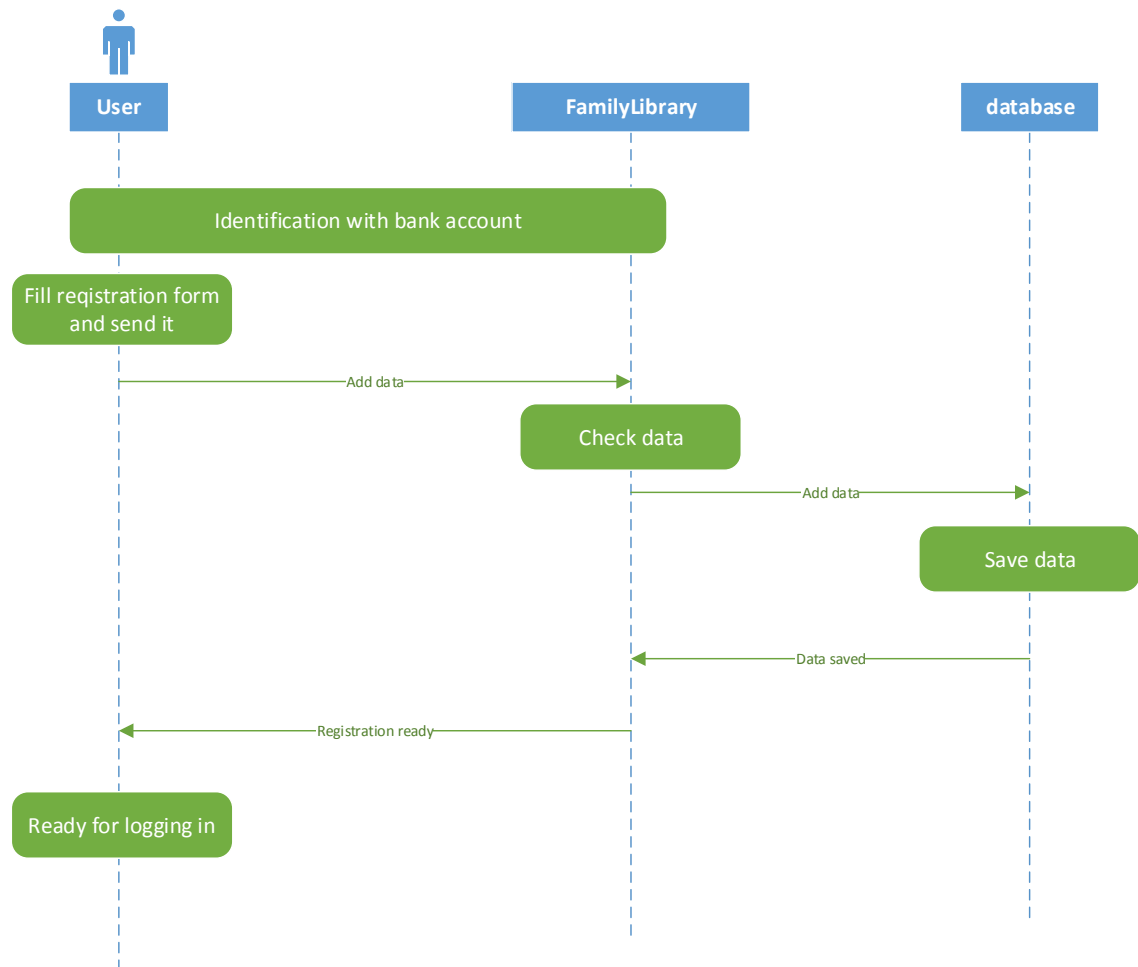


Figure 1. Successful registration

Figure 2 on the next page shows a failed registration. In this case there is something wrong with the data. There might be something missing or there might be something in a wrong format. The password, chosen by the user, can be too weak, for instance. Thus, the data is not saved to a database. The FamilyLibrary service informs the user that something has gone wrong. The user can check the registration form, make the needed changes and send the corrected form to the service again.

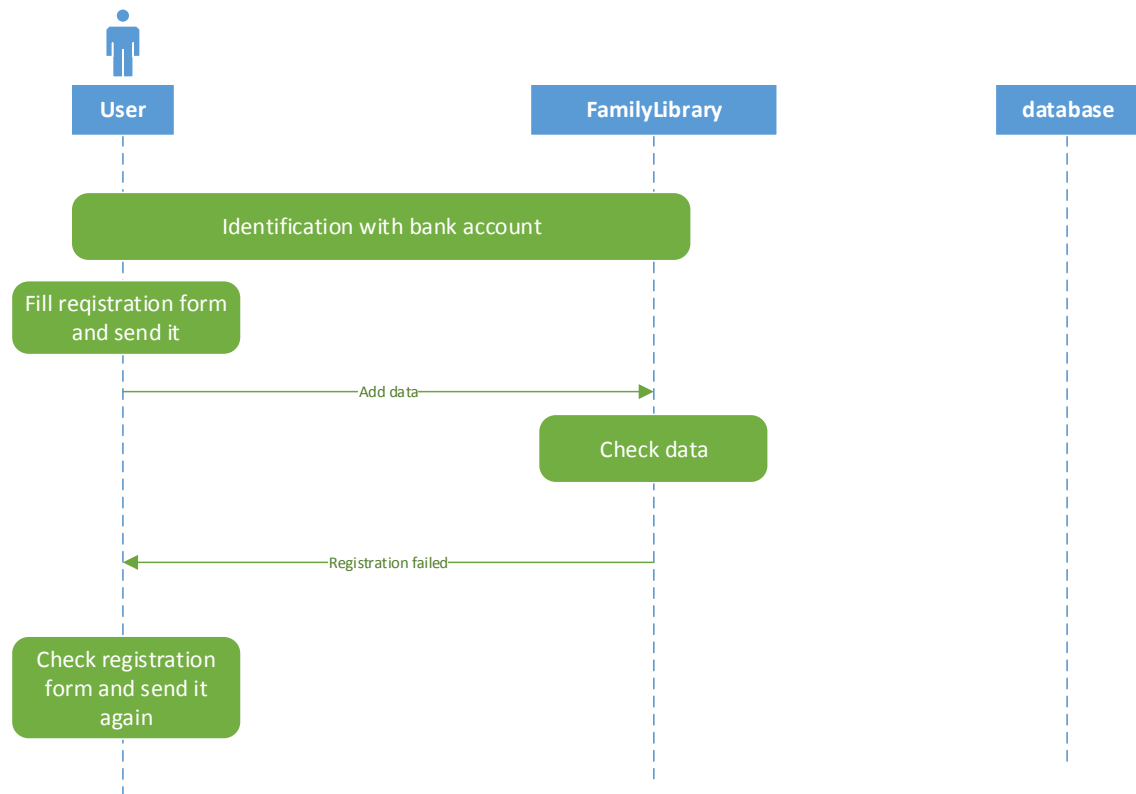


Figure 2. Failed Registration

3.3.2 Identification

A safe enough identification can be done by using a bank account identification. The bank account identification is called Tupas identification service. Tupas identification service is served by the banks in Finland. A service provider, in this case a party which serves the FamilyLibrary, can buy the Tupas identification service from some bank. A contract for using Tupas identification service is made between the bank and the FamilyLibrary service provider. The party who implements the FamilyLibrary service will have no contracts with the bank. It only implements the technical parts which are needed for using the Tupas identification service. The contract between the bank and service provider must be made with all banks whose Tupas identification service the service provider wants to use. [4]

3.3.3 Personal Data Updating

Updating any personal data, for example name, must be allowed. Personal data can be changed due to many different reason. For example the person moves, gets married, gets divorced, has a new child, children get older, password needs to be changed, etc. When personal data is updated, the service has to make sure that the person updating the data is the correct person. That is why strong identification has to be used. In that case the bank account identification is used once again. If only child care times are updated, then a normal password identification is enough and that is done by logging in to the FamilyLibrary service.

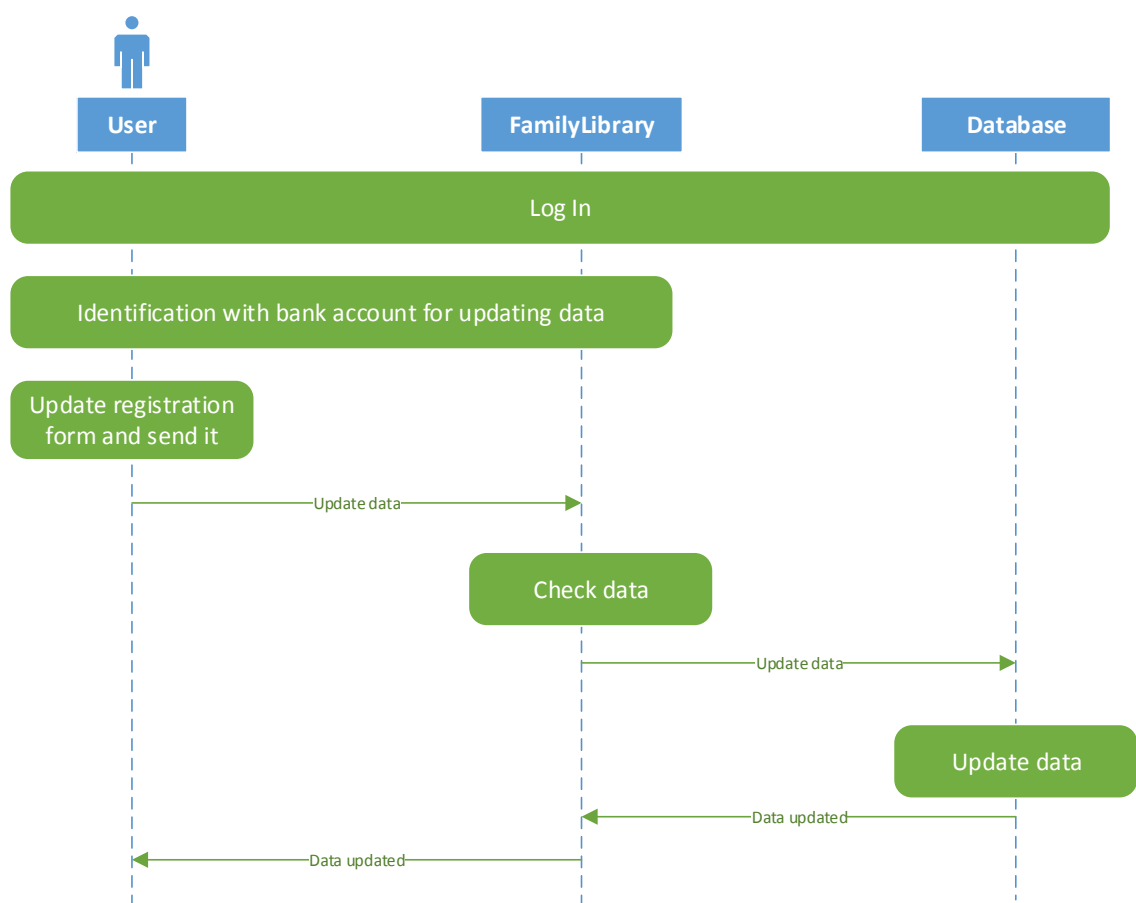


Figure 3. Successful Personal Data Update

Figure 4 shows a failed personal data updating procedure. In this case there is something wrong in the data. There might be something missing or there might be something in a wrong format. The new password might be too weak. Therefore, the data is not saved to the database. The FamilyLibrary service informs the user that something has gone

wrong. The user can check the registration form, make the needed changes and send the corrected form to the service again.

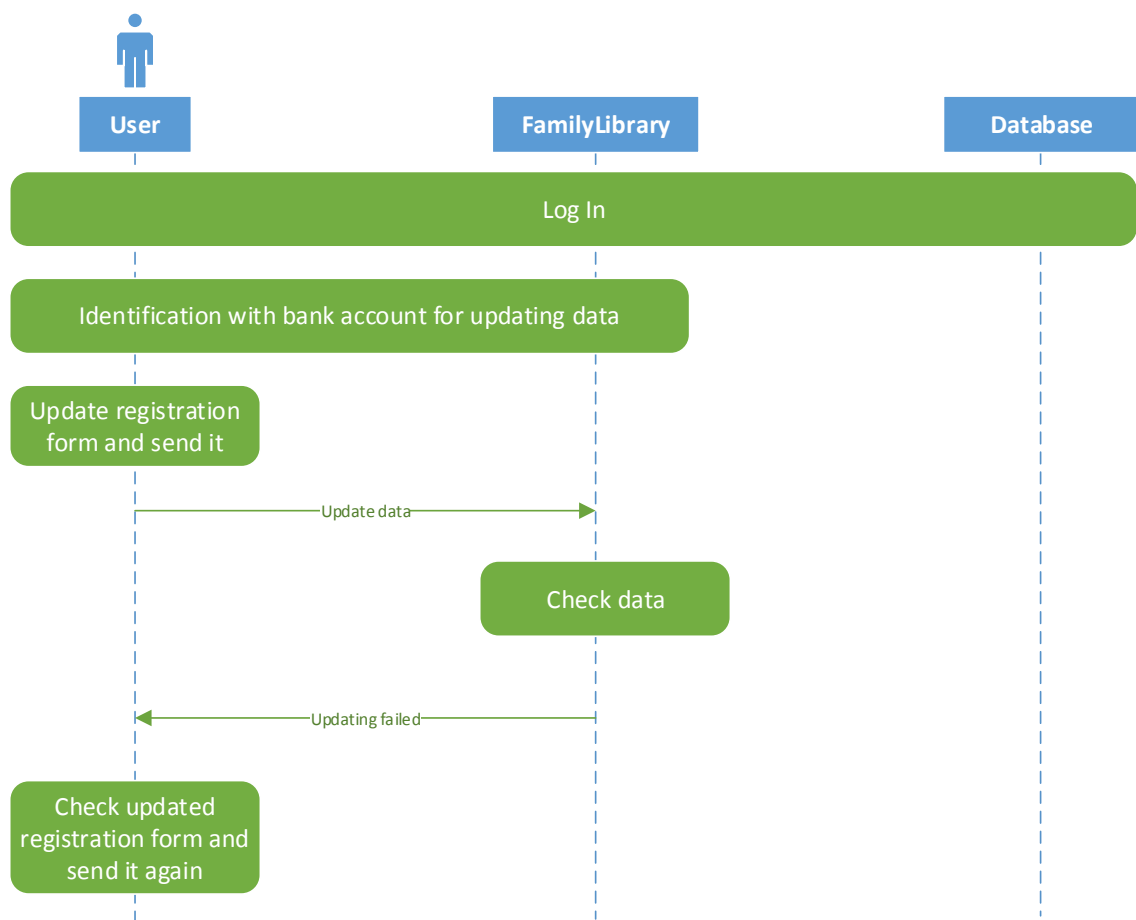


Figure 4. Failed Personal Data Update

3.3.4 Login with an E-mail Address and Password

Once the registration is successfully done the user can use her/his e-mail address and password for logging in to the FamilyLibrary service. The FamilyLibrary service checks from the database that a compatible e-mail address and password are found from the database. The log-in succeeds when a compatible e-mail address and password are found from the database.

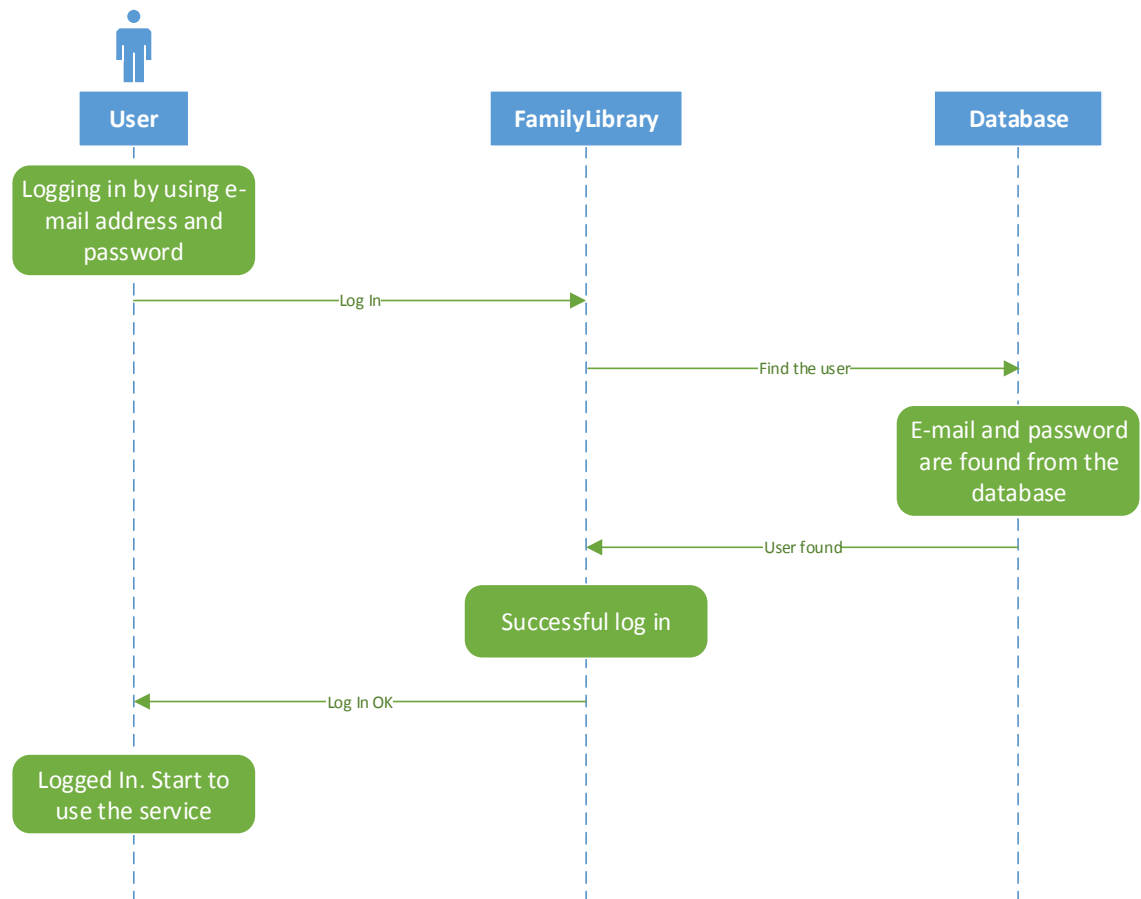


Figure 5. Successful Log In

If an e-mail address and/or password is not found from the database, log-in will fail. The user will be informed about that. The user can then check the e-mail address and password and try to log in again.

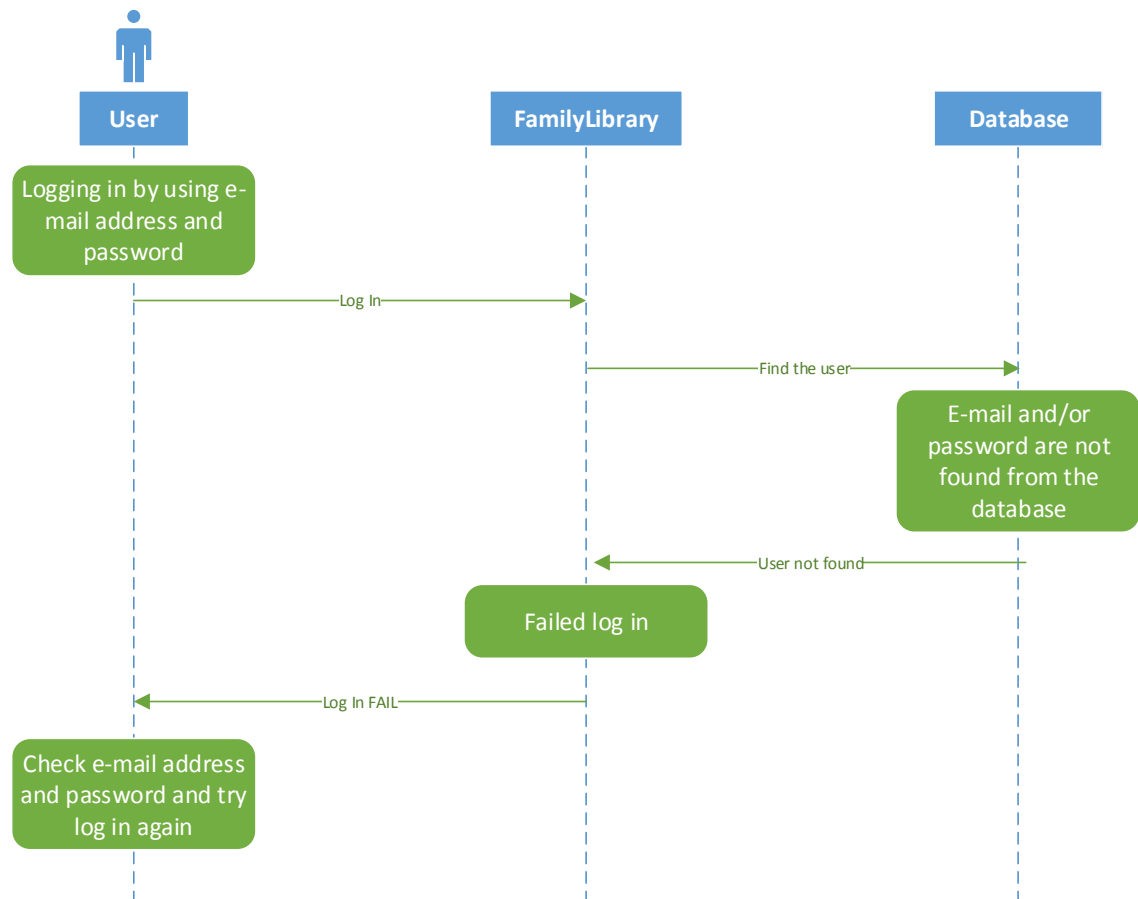


Figure 6. Failed Log In

3.3.5 Query for a Forgotten Password

It is quite a common situation that a user forgets her/his password. In this situation there must be a way in the FamilyLibrary service to make a query of the password or reset the old, forgotten password. When the log-in has failed due to a wrong password the service has to ask from the user: is the password forgotten. There must be an option to make a password query or resetting straight from the log-in page. The password query or resetting is done by using the user's e-mail address. The password or reset password will be sent to the user's e-mail.

Figure 7 shows the password query option.

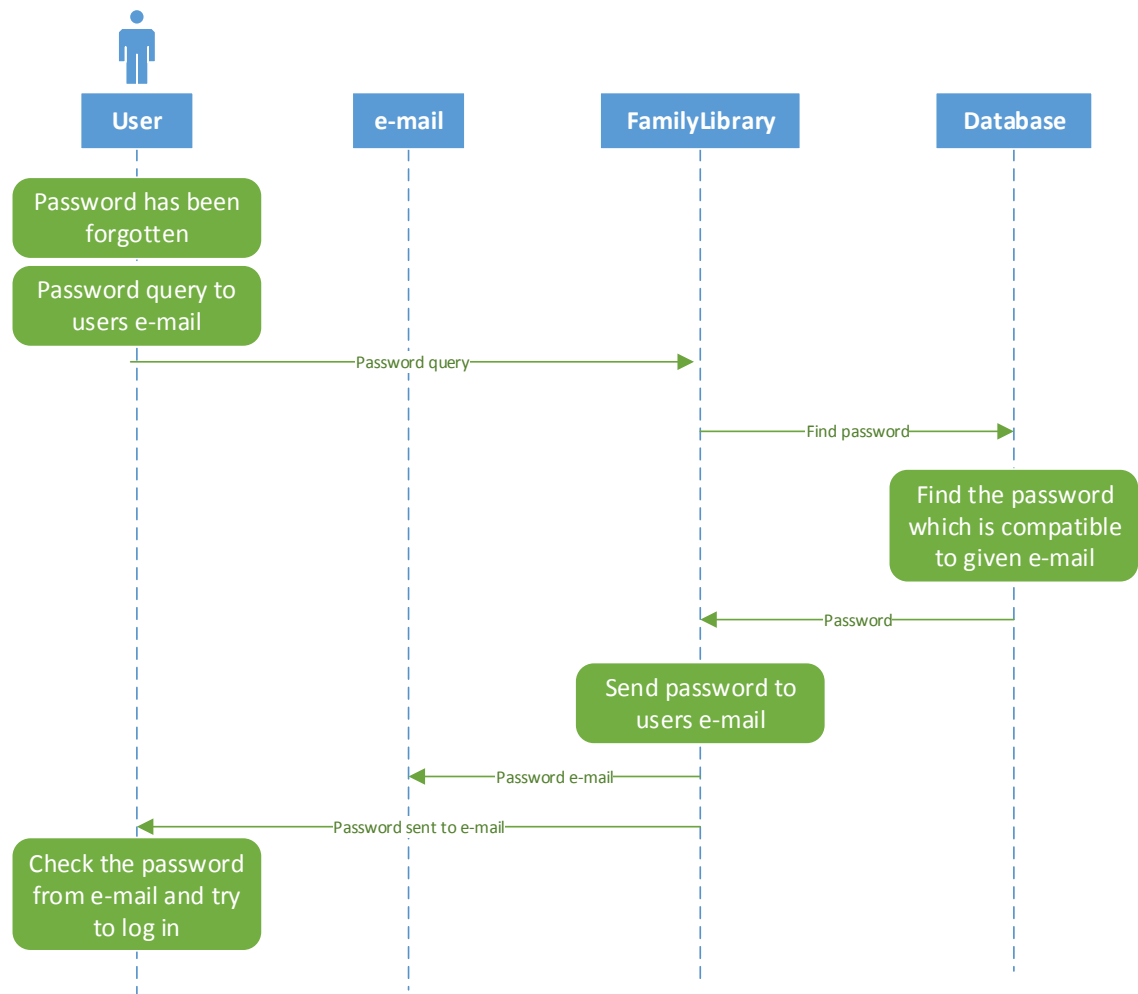


Figure 7. Successful Password Query

A successful password resetting option is shown in Figure 8.

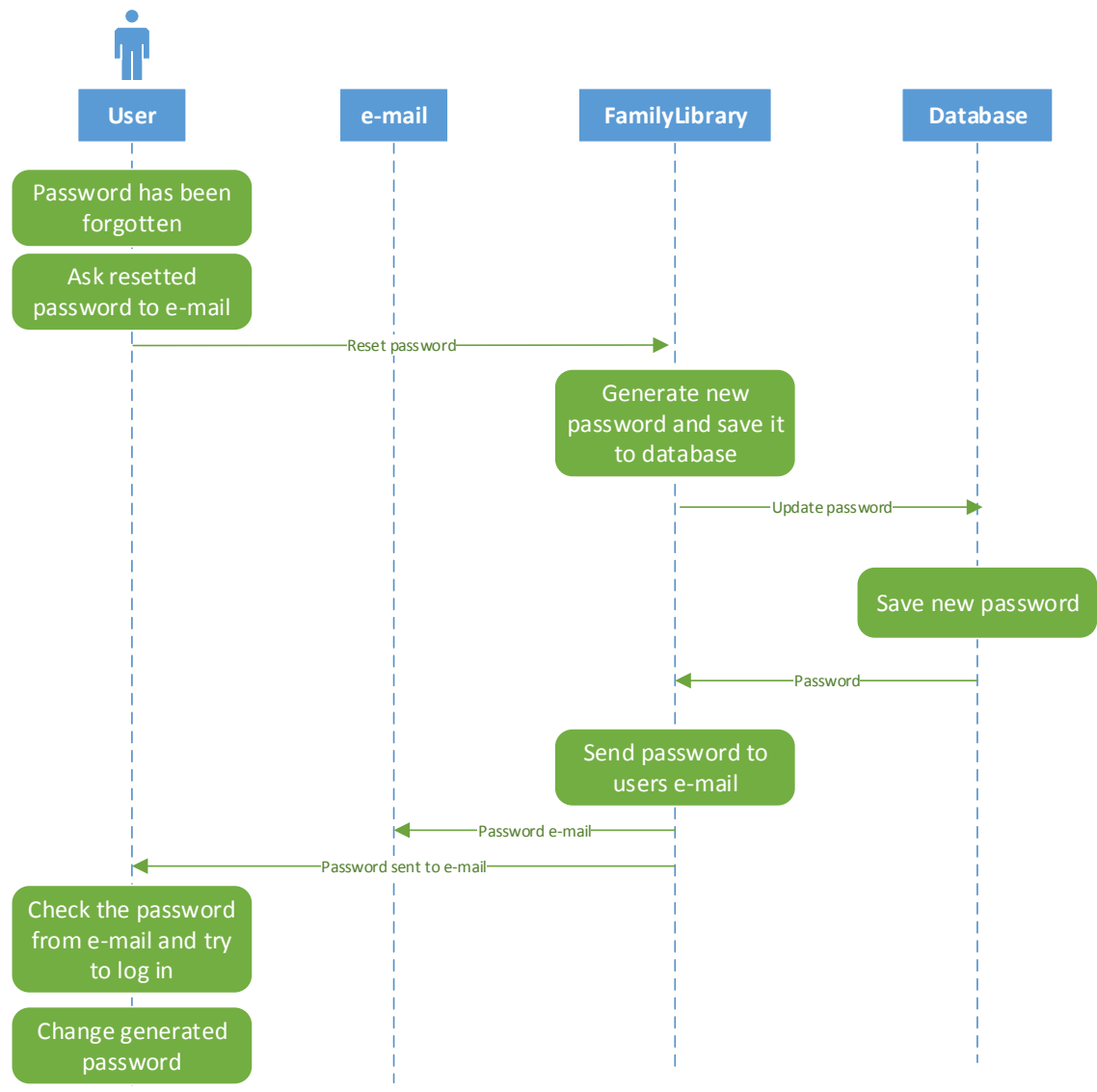


Figure 8. Successful Password Resetting

Failed password queries and resetting cases are not shown here but the FamilyLibrary service must be able to handle those cases too. It could be something as simple as the user giving an e-mail address that is not found in the database. The service must ask the user to check the e-mail address and to make a password query or resetting again. There might be also something else which can go wrong. For example a new password is not saved successfully in the database. Every failed case has to be informed to the user.

3.3.6 Adding to Calendar Possible Times to Give Child Care Help

After the registration is done and the user is logged in, the user can start to use the FamilyLibrary service. When child care help will be given, then all suitable times can be added to a calendar. There is a calendar in the FamilyLibrary service from where the user can click days and add suitable times for the child care help offered. All given times will be saved to the Calendar table of the database. It has to be also possible to update or remove the times entered and saved. Situations may change and it must be possible to make changes.

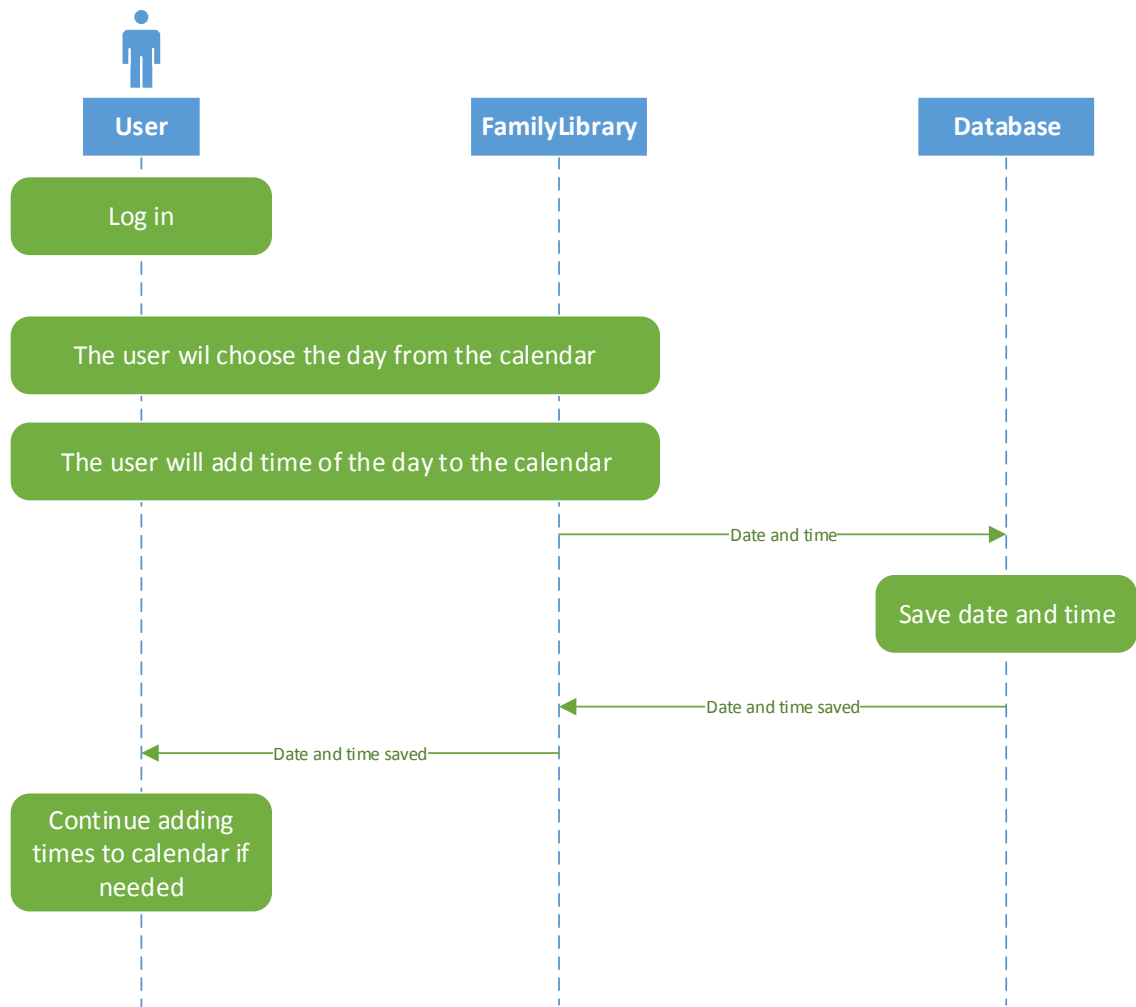


Figure 9. Adding Child Care Times Offered to the Calendar

3.3.7 Search Tool for Seeking Child Care Help

There will be a search tool, in the FamilyLibrary service, which is used for searching child care help. A user will give the parameters to the tool and all possible child care help choices are retrieved from the database. The user has to give the following parameters to the search tool:

- Date and time
- Location from the user's home address

The search tool uses also parameters from the database for searching possible child care help. The next parameters are straight from the database and are added to the database during registration:

- Child amount
- Child gender
- Home address

The search tool lists families which can give child care at the requested time, with the given parameters. The user can choose how far from the home address (2 km, 5 km, 10 km...) the search tool will list families. The user can choose the family and then see more information about them. The home address of the family is not shown due to security reasons. The address is only used by the search tool. The address is then clarified after families have contacted each other. The user can call or send e-mail asking for more details and agree with the family. When two families have made an agreement of child care help, the family which offers help will modify their calendar on the FamilyLibrary service and removes the agreed day from child care help days. The agreed day will be booked then.

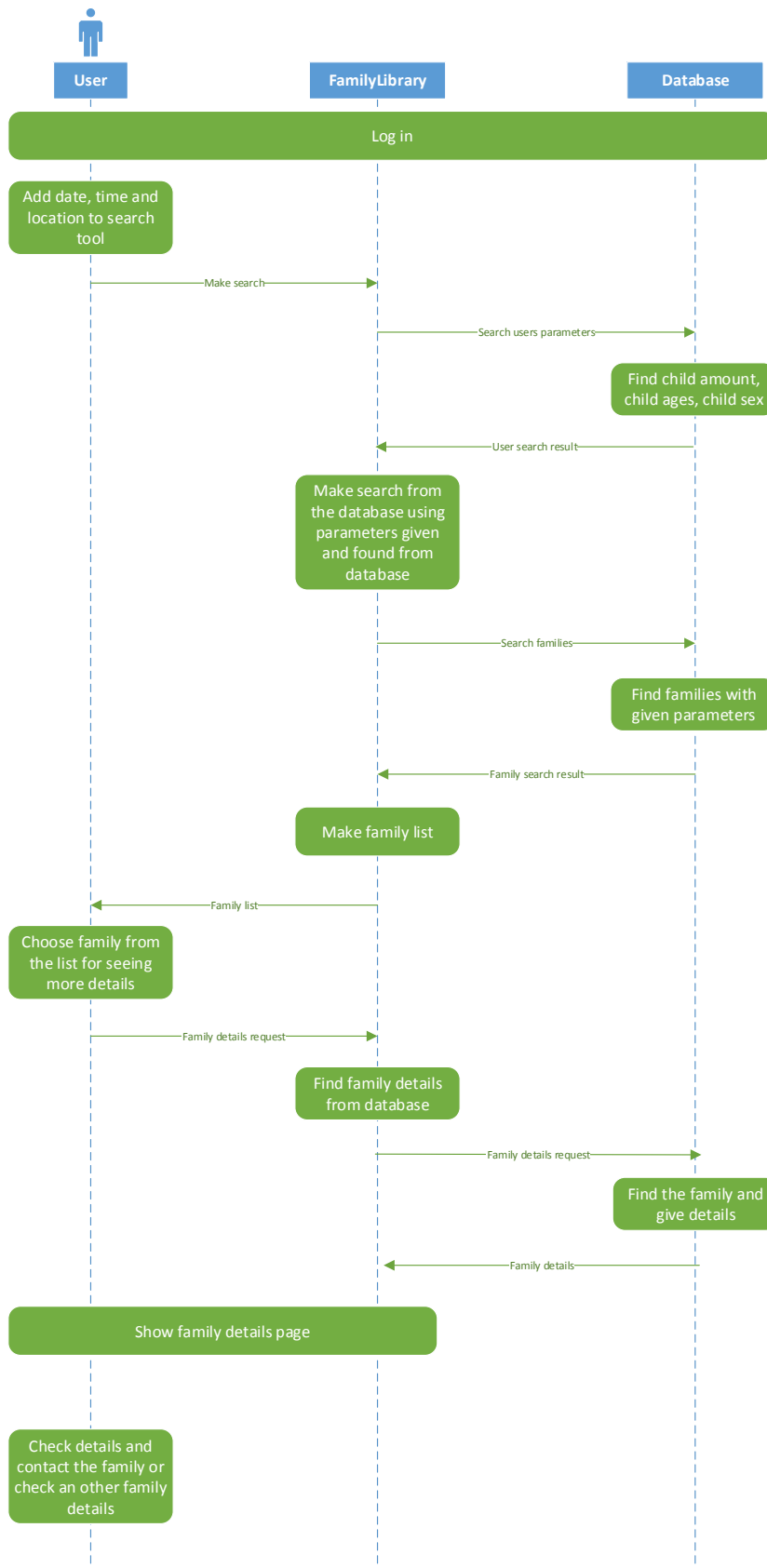


Figure 10. Search Tool for Child Care Help

3.3.8 Search Tool for Seeking Friend Families

There will be a search tool, in the FamilyLibrary service, which is used for seeking friend families. The user will give parameters to the tool and possible friend family choices are retrieved from the database. The user has to give the following parameters to the search tool:

- Children's age range (for example 0-5 years)
- Location from the user's home address

The Search tool lists families which are closest compared to the given parameters. The user can choose how far from the home address (2 km, 5 km, 10 km...) the search tool lists families. The user can choose the family and then see more information about them. The home address of the family is not shown due to security reasons. The address is only used by the search tool. The address is then clarified after the families have contacted each other. The user can call or send e-mail asking for more details or just talk with the other family which has the same kind of life situation. The families can organize for example a picnic to some park and see if the other family can be a friend family for their family.

3.3.9 Recommendation Collection

Families can collect recommendations. Families can give those recommendations after some child care help has been given or used. There will be buttons for recommendation in the FamilyLibrary service. If the family has liked the other family the user can click a "+" button on that family. If something went wrong the user can click a "-" button. The positive and negative clicks are collected in the database. The FamilyLibrary service calculates the amount of clicks. For example if the family has had three "+" clicks then three positive recommendations are found in the database.

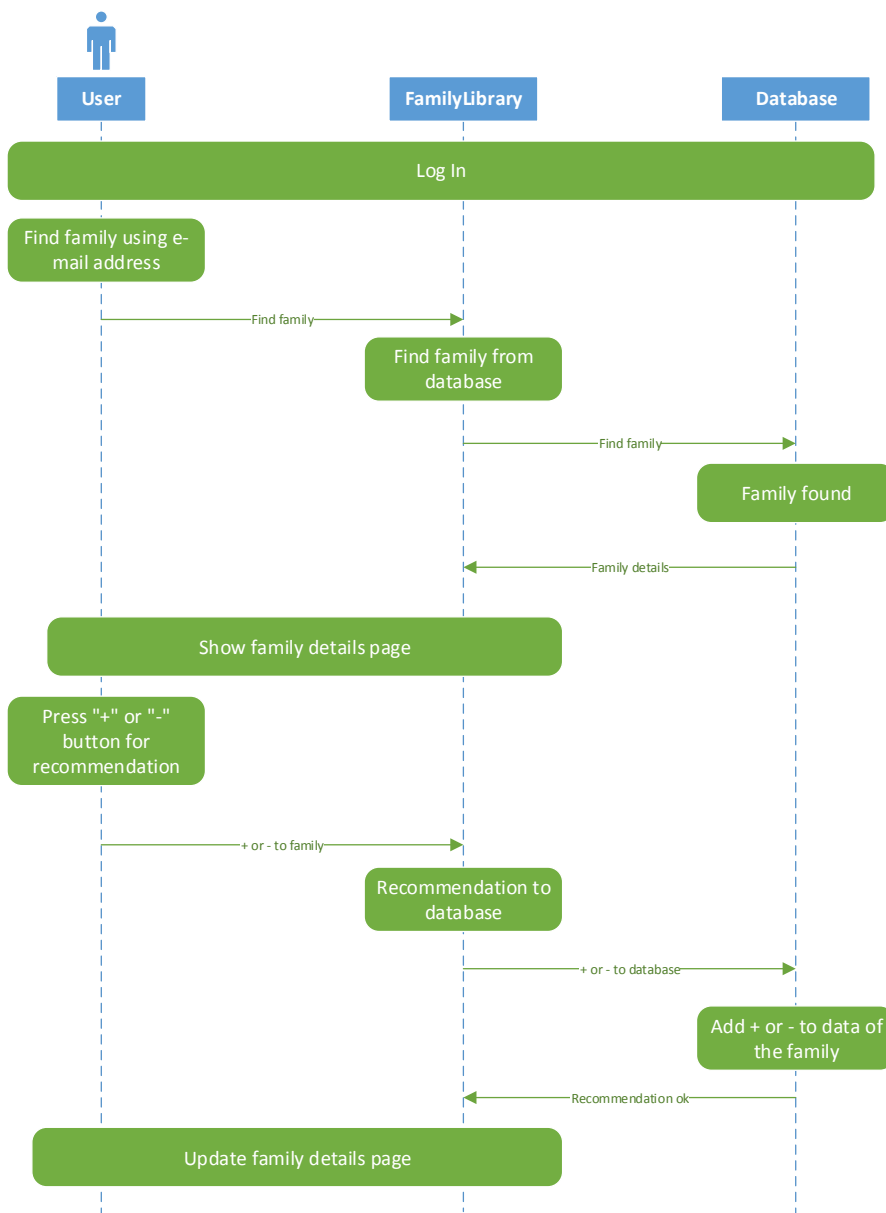


Figure 12. Recommendation Collection

3.3.10 Stopping the Service Use

The family might want to stop using the FamilyLibrary service. For example it might be so that the kids have been grown up and child care help is not needed anymore, or a family will not offer any child care help anymore. Maybe the family has found the needed amount of friend families and they don't need FamilyLibrary service anymore. When the family wants to stop using FamilyLibrary service, all data concerning that family will be removed from the database. Data removal is defined in Personal data act 22.4.1999/523 Chapter 7 34§ [3].

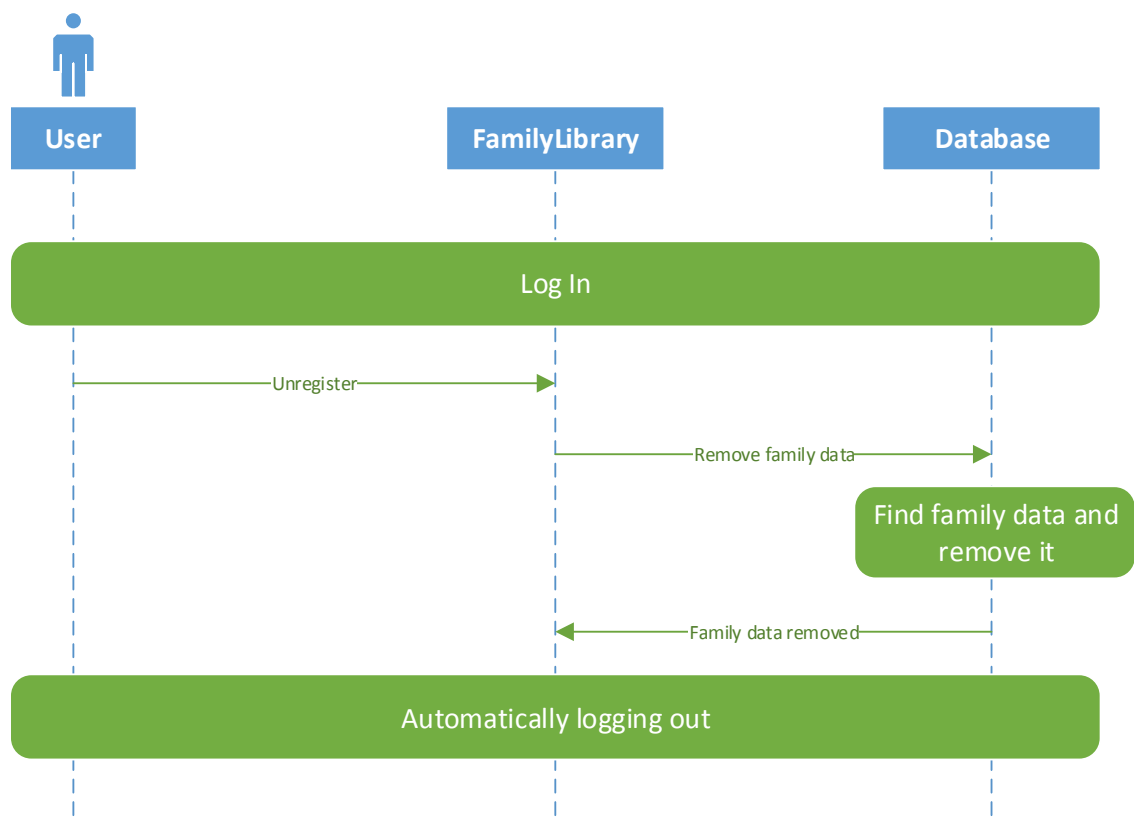


Figure 13. Stopping the service Use

3.3.11 Logging Out

When a user has done all she/he needed to do, it will be time to log out. There will be two ways for logging out. The first way is manual logging out. The user clicks a log out button and logging out will start. The second way is automatic logout. This happens when the user wants to stop using the service or when she/he forgets to log out. If the user

forgets to log out, the automatic logout starts after the user has been inactive for fifteen minutes. The two ways are depicted in Figures 14 and 15.

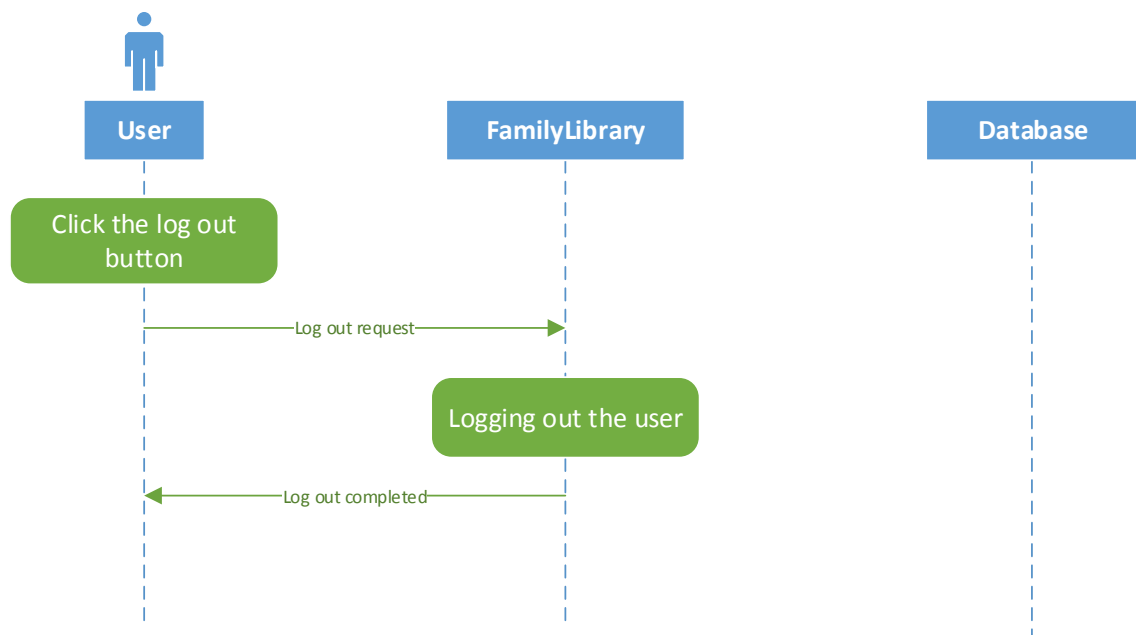


Figure 14. Manual Logout

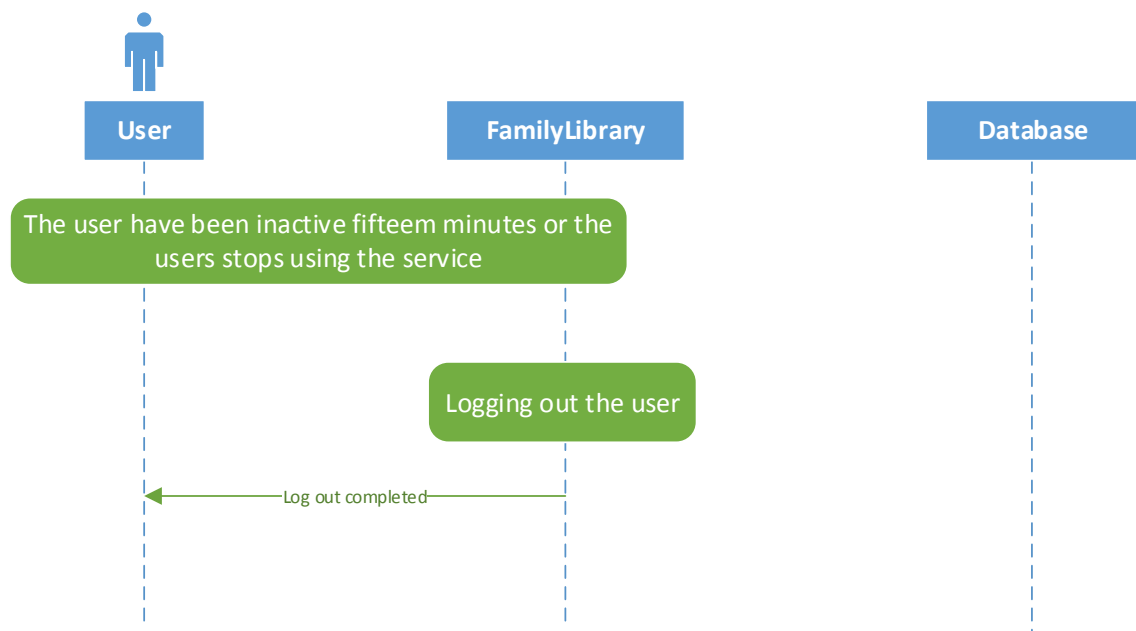


Figure 15. Automatic Logout

3.3.12 Activity Timer

The FamilyLibrary service has to have an individual activity timer for each user logged in. The timers start to count fifteen minutes from every activity the users engage in. If the user is inactive for fifteen minutes, automatic logout will occur.

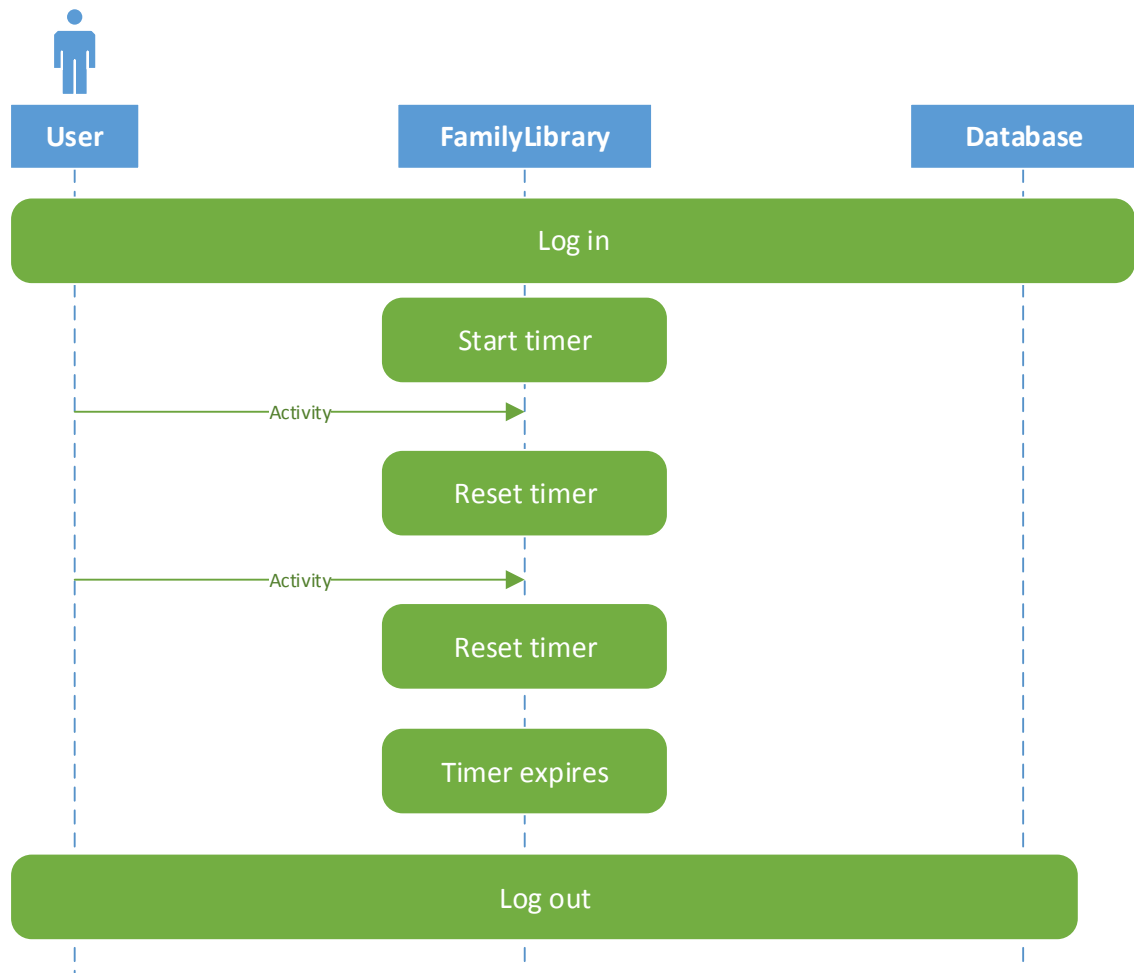


Figure 16. Activity Timer

3.4 Designing of the FamilyLibrary Database

Microsoft Visio was used for database designing. Crow's Foot Database Notation was chosen for designing the FamilyLibrary database.

There are four tables in the design.

- User Table
- Calendar Table
- Children Table
- Children_to_take_care Table

All parameters of tables are introduced in more detail in Appendix 3.

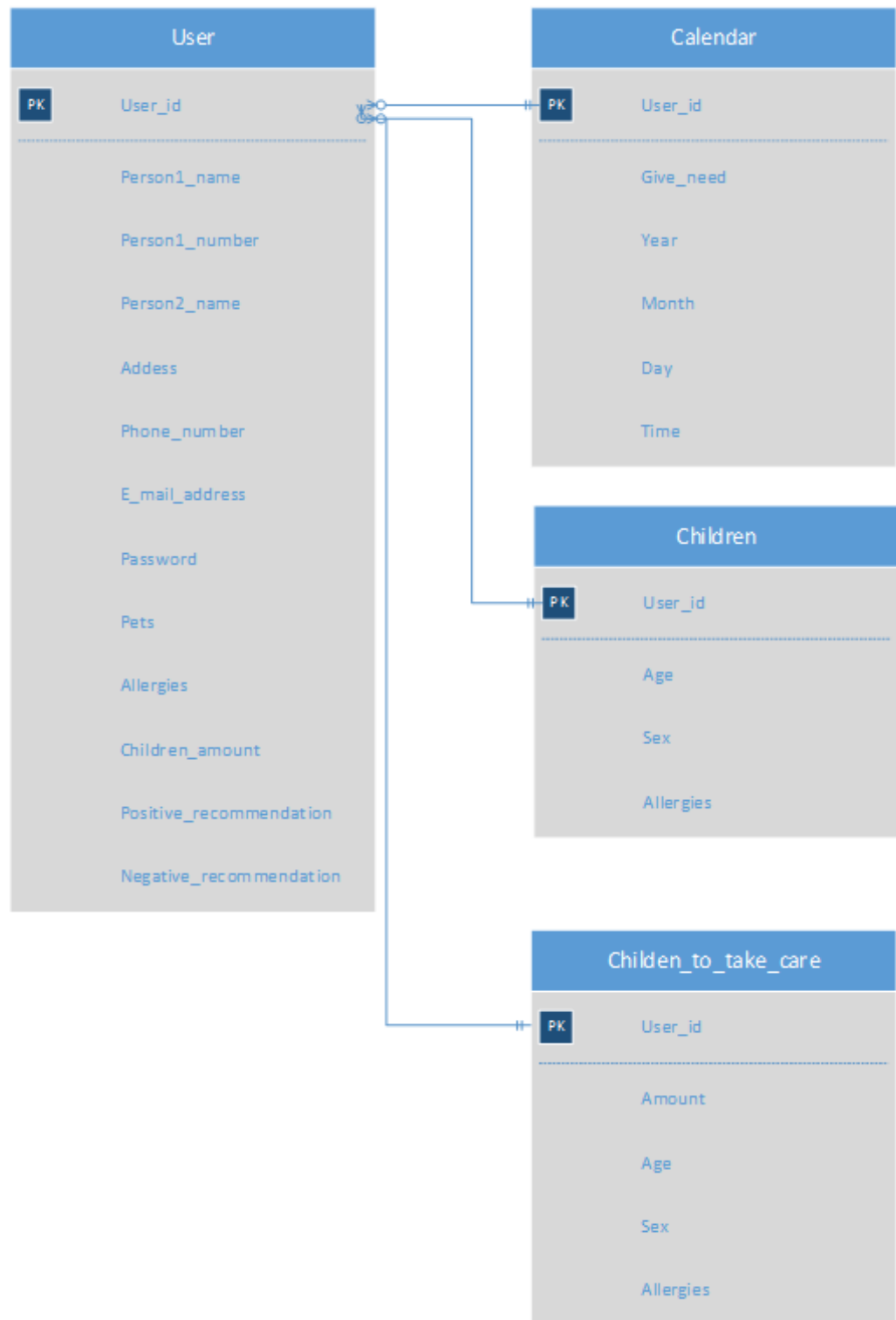


Figure 17. Crow's Foot Database Notation Design of FamilyLibrary Database

3.5 Creating Database Project and Tables with Visual Studio

Visual Studio has a tool for creating a SQL Server database. When a new project is started it has to choose a SQL Server template. Then the name and location for that database has to be selected.

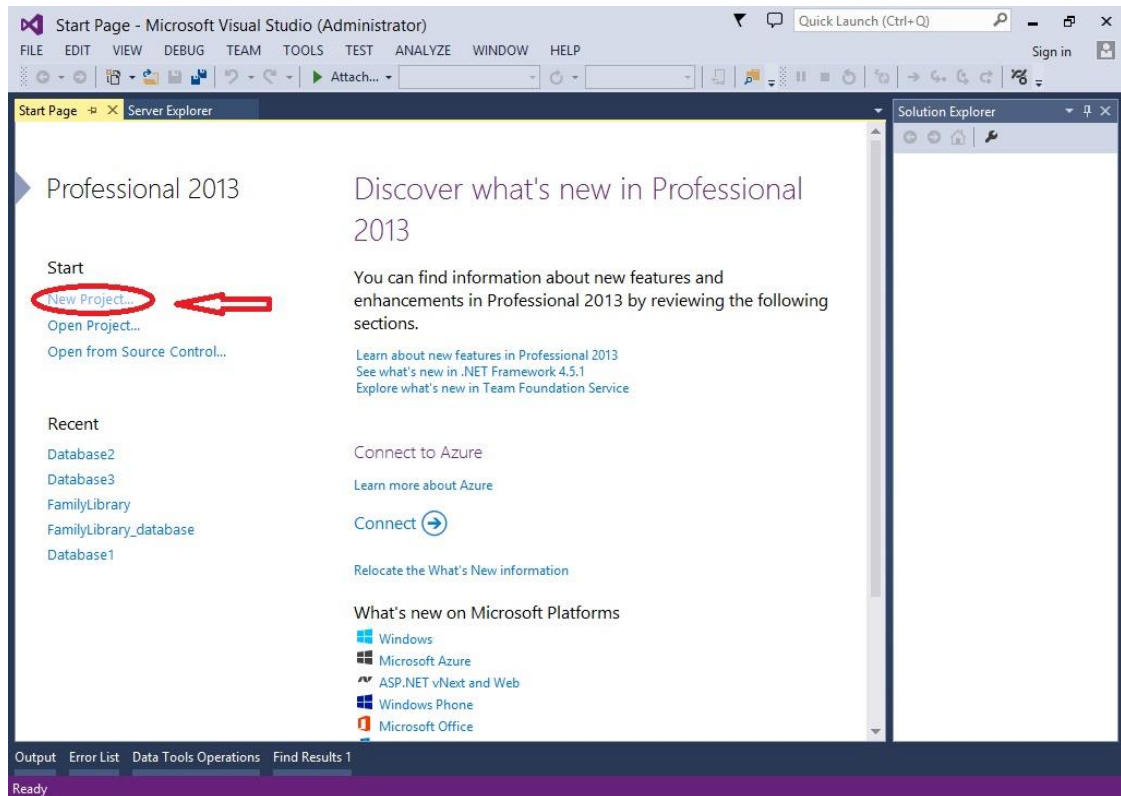


Figure 18. Starting New Project

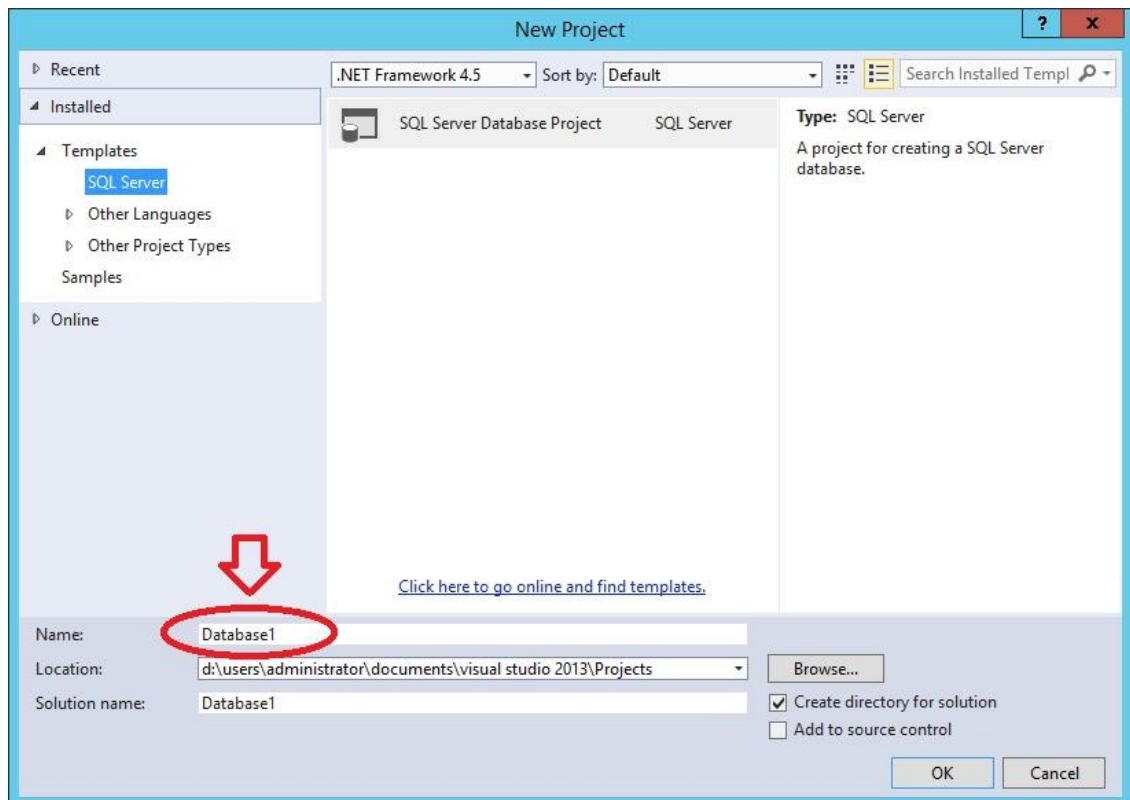


Figure 19. Starting New SQL Server Database Project

There was some problem in Visual Studio or Visual Studio installation because it was not possible to change the name of the project. If the name was changed it caused a problem when new tables were created. Visual Studio just stopped responding. That is why the project name “Database1” is used.

When a new project is created the creation of the tables can be started. An SQL Server Object Explorer can be found in the tools menu and it can be used when wanting to create new tables.

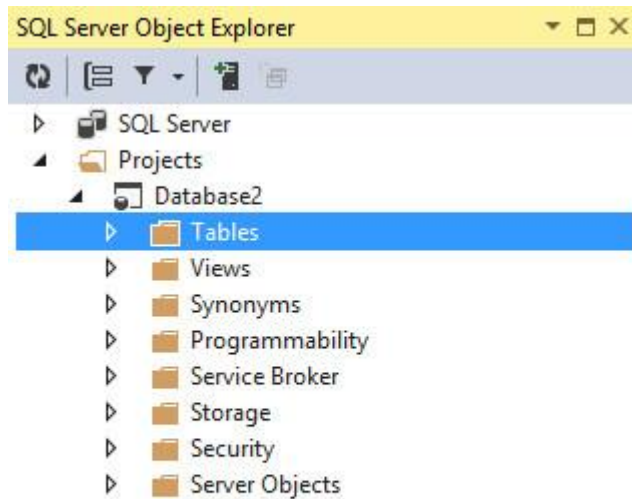


Figure 20. SQL Server Object Explorer

When the table is created it has to have a reasonable name. In the following picture a User table is created.

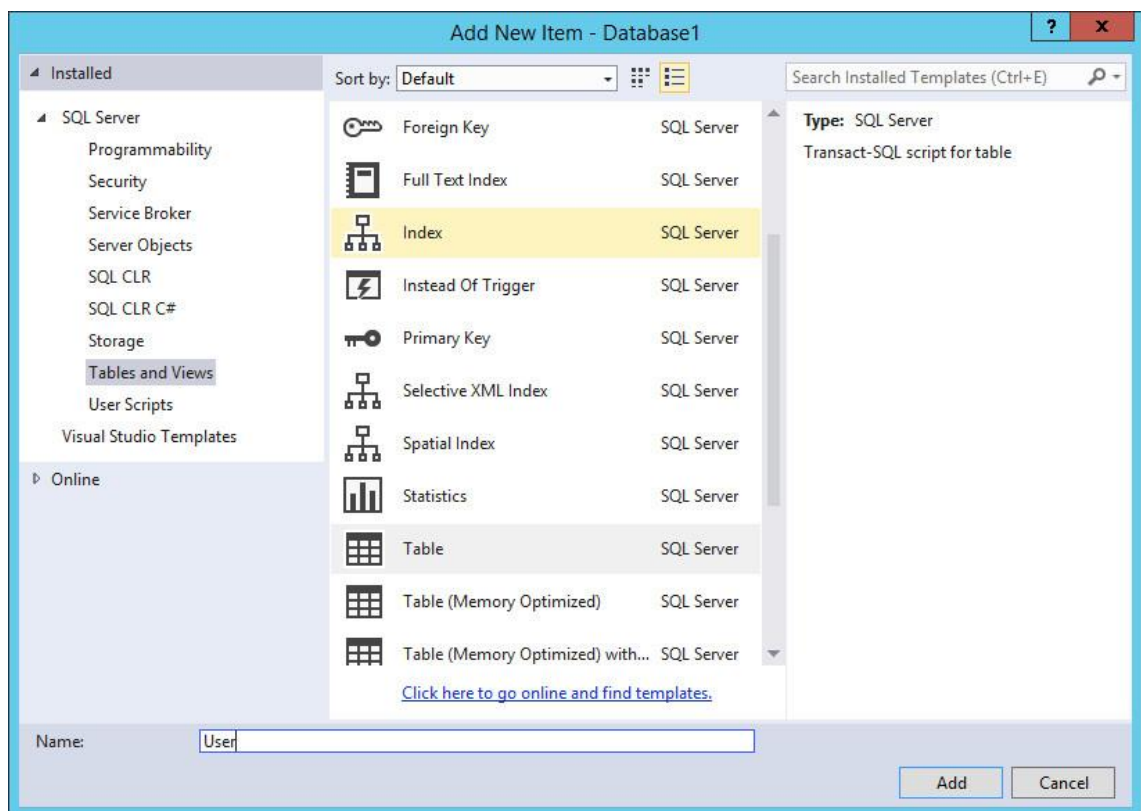


Figure 21. Creation of User Table

With Visual Studio it is possible to create a table and table columns by editor or by written SQL. Both versions are shown at the same time. In the next chapters it will be shown in detail how different tables, used by the FamilyLibrary, are created in the project.

3.5.1 User Table

User table is the main table of the FamilyLibrary service. Most of the data is collected there.

The next picture shows how the User table looks with Visual Studio database designer.

| Name | Data Type | Allow Nulls | Default |
|-------------------------|-------------|-------------------------------------|---------|
| User_Id | int | <input type="checkbox"/> | 0 |
| Person1_name | varchar(50) | <input type="checkbox"/> | NULL |
| Person1_number | varchar(50) | <input type="checkbox"/> | NULL |
| Person2_name | varchar(50) | <input checked="" type="checkbox"/> | NULL |
| Address | varchar(50) | <input type="checkbox"/> | NULL |
| Phone_number | varchar(50) | <input type="checkbox"/> | NULL |
| E-mail_address | varchar(50) | <input checked="" type="checkbox"/> | NULL |
| Pets | bit | <input checked="" type="checkbox"/> | 0 |
| Allergies | varchar(50) | <input checked="" type="checkbox"/> | NULL |
| Children_amount | int | <input checked="" type="checkbox"/> | NULL |
| Positive_recommendation | int | <input checked="" type="checkbox"/> | NULL |
| Negative_recommendation | int | <input checked="" type="checkbox"/> | NULL |

Keys (1)
 <unnamed> (Primary Key, Clustered: User_Id)
 Check Constraints (0)
 Indexes (0)
 Foreign Keys (0)
 Triggers (0)

Figure 22. Snapshot from User Table (Design)

The next picture shows SQL code of User table creation.

```

CREATE TABLE [dbo].[User]
(
    [User_Id] INT NOT NULL PRIMARY KEY DEFAULT 0,
    [Person1_name] VARCHAR(50) NOT NULL DEFAULT NULL,
    [Person1_number] VARCHAR(50) NOT NULL DEFAULT NULL,
    [Person2_name] VARCHAR(50) NULL DEFAULT NULL,
    [Address] VARCHAR(50) NOT NULL DEFAULT NULL,
    [Phone_number] VARCHAR(50) NOT NULL DEFAULT NULL,
    [E-mail_address] VARCHAR(50) NULL DEFAULT NULL,
    [Pets] BIT NULL DEFAULT 0,
    [Allergies] VARCHAR(50) NULL DEFAULT NULL,
    [Children_amount] INT NULL DEFAULT NULL,
    [Positive_recommendation] INT NULL DEFAULT NULL,
    [Negative_recommendation] INT NULL DEFAULT NULL
)

```

Figure 23. Snapshot from User Table (SQL)

3.5.2 Calendar Table

A Calendar table is a place where possible times for child care offered are collected.

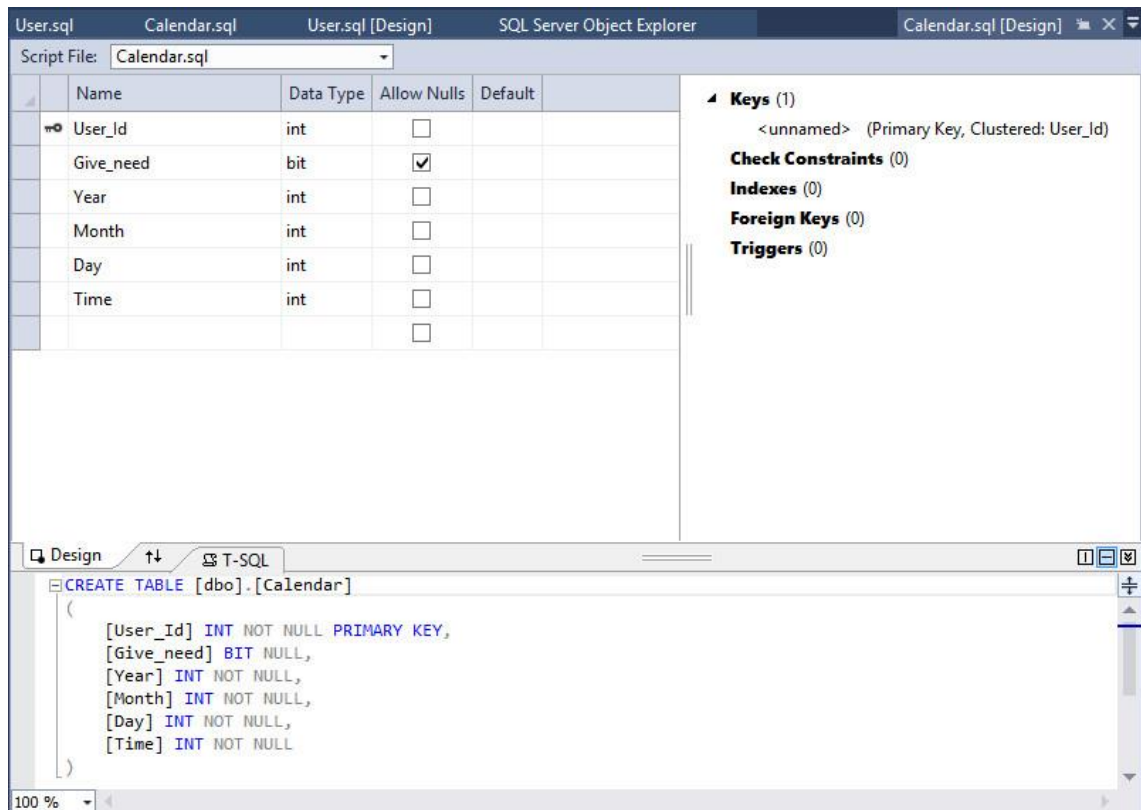


Figure 24. Snapshot from Calendar Table

3.5.3 Children Table

A Children table is a place where information of the children of a family is collected.

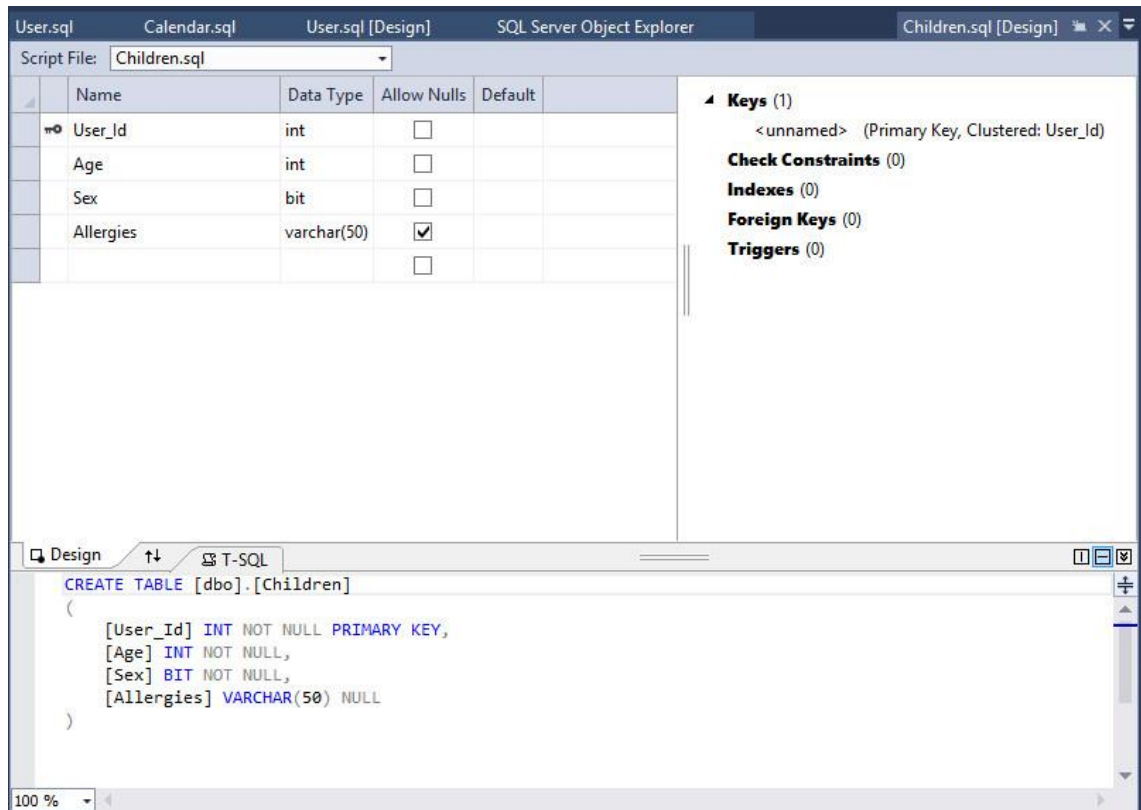


Figure 25. Snapshot from Children Table

3.5.4 Children_to_take_care Table

A Children_to_take_care table is a place where data of possible visitor children is collected.

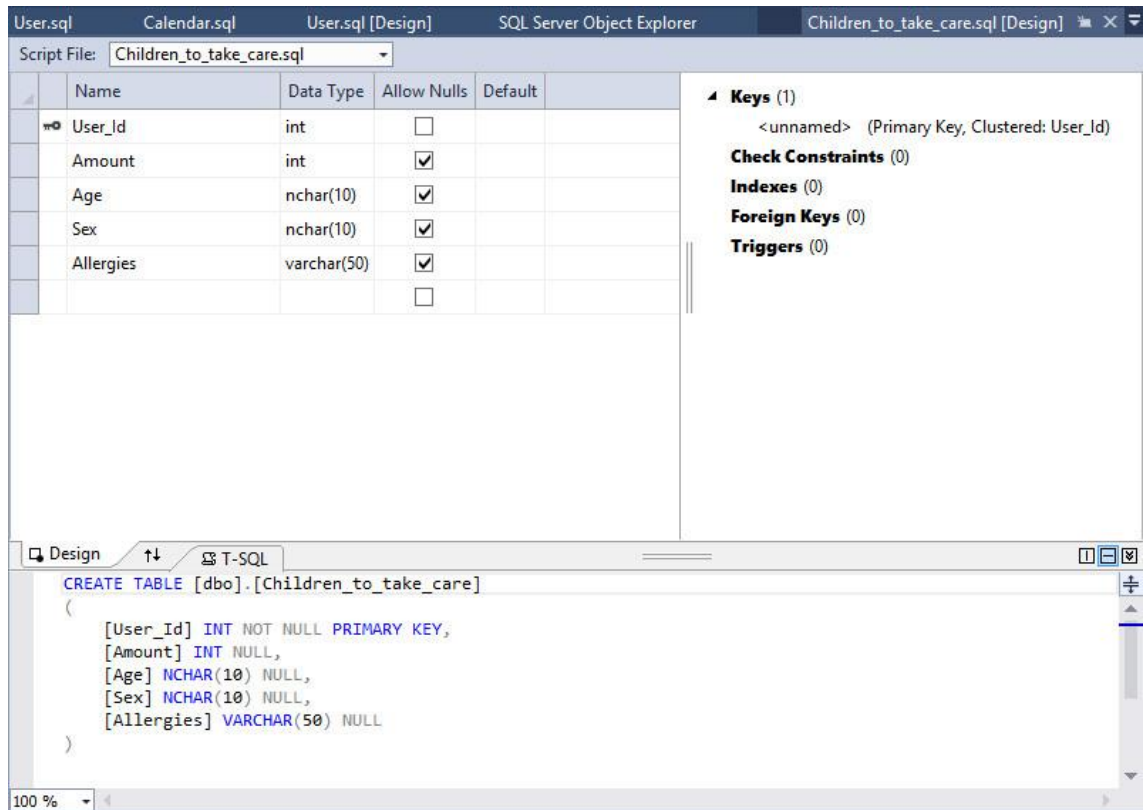


Figure 26. Snapshot from Children_to_take_care Table

3.6 Creating Database and Tables with Microsoft SQL Server Management Studio

In this section a database and a User table are created with Microsoft SQL Server Management Studio. Microsoft SQL Server Management Studio was a part SQL Server 2014 Express installation.

When the SQL Server Management Studio is opened, that SQL Server which is wanted to be managed, needs to be connected. There might be on-premises databases or databases in the network. In this thesis a test database is created on-premises. With the SQL Server Management Studio it is also possible to create databases to network servers.

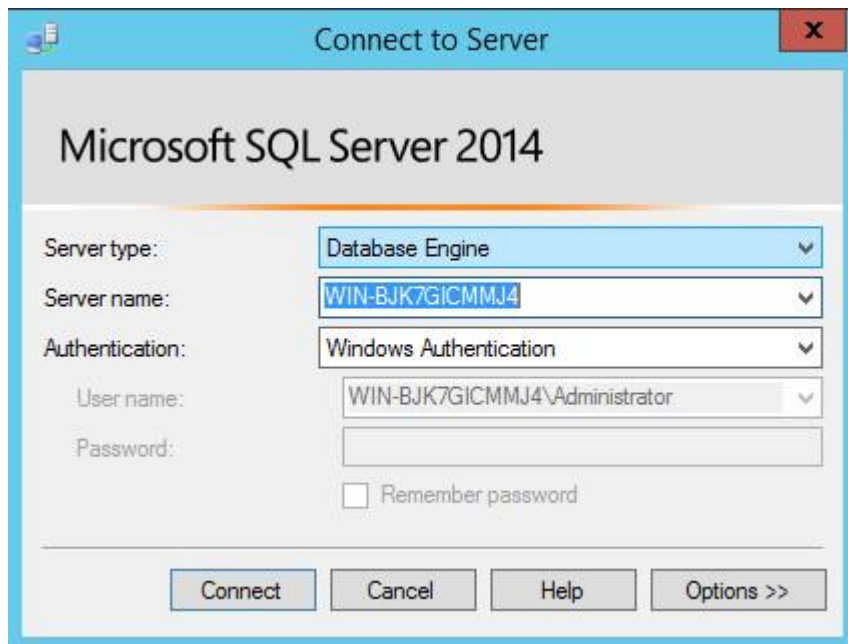


Figure 27. Connect to SQL Server with SQL Server Management Studio

When a new database is needed to create, right-click on the mouse in Object Explorer on top of the text “Database”.

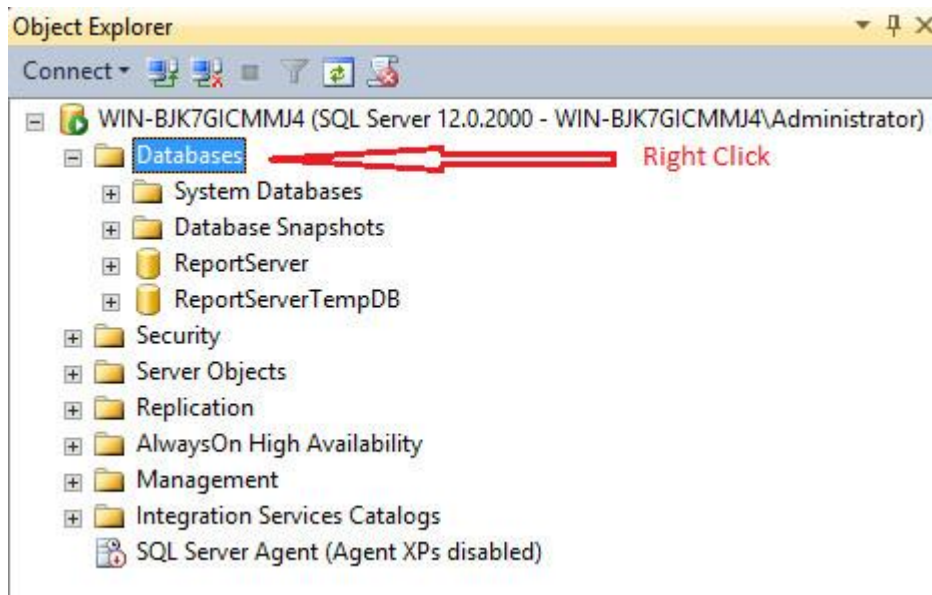


Figure 28. Creating New Database with SQL Server Management Studio

A New Database sheet opens now and then a name of the new database will be chosen. In this chapter the FamilyLibrary database creation is tested and then the name of the database is FamilyLibrary. Add the name and click “ok”.

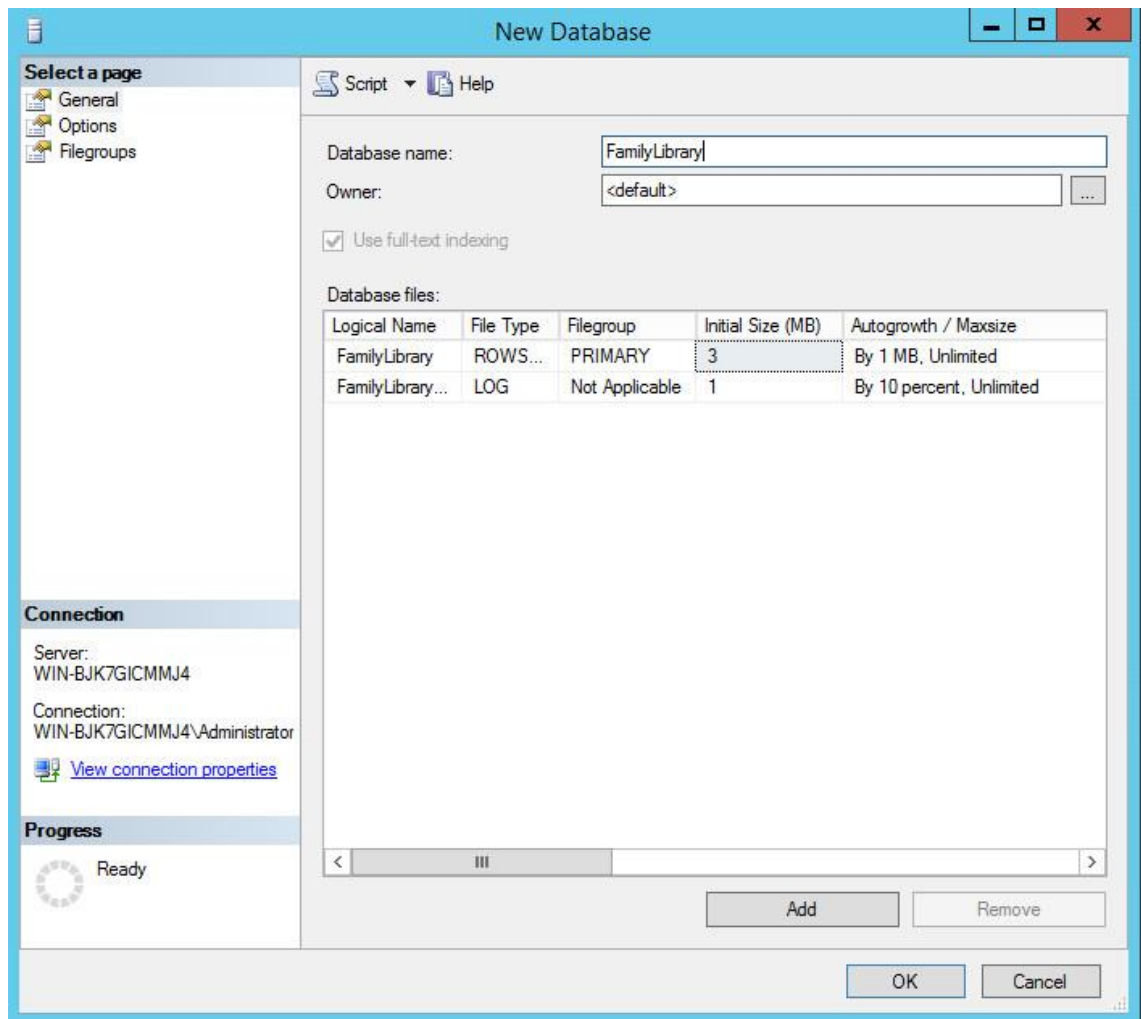


Figure 29. FamilyLibrary Database Creation with SQL Server Management Studio

Now the FamilyLibrary database can be seen in the Object Explorer.

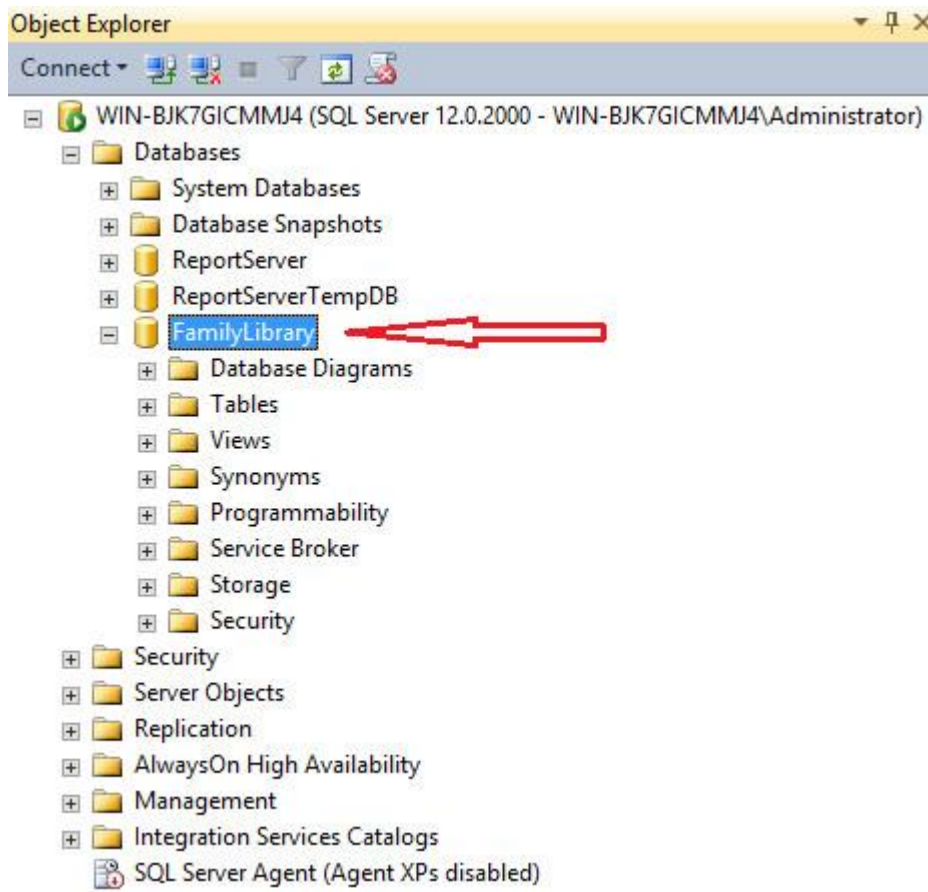


Figure 30. New FamilyLibrary Database Is Created

After the new database is created the creation of new tables can be started. There are several tables designed to the FamilyLibrary service but in this section only a User table is created. The table creation can be started by right clicking the text "Tables" under the FamilyLibrary database. Then choose New -> Table...

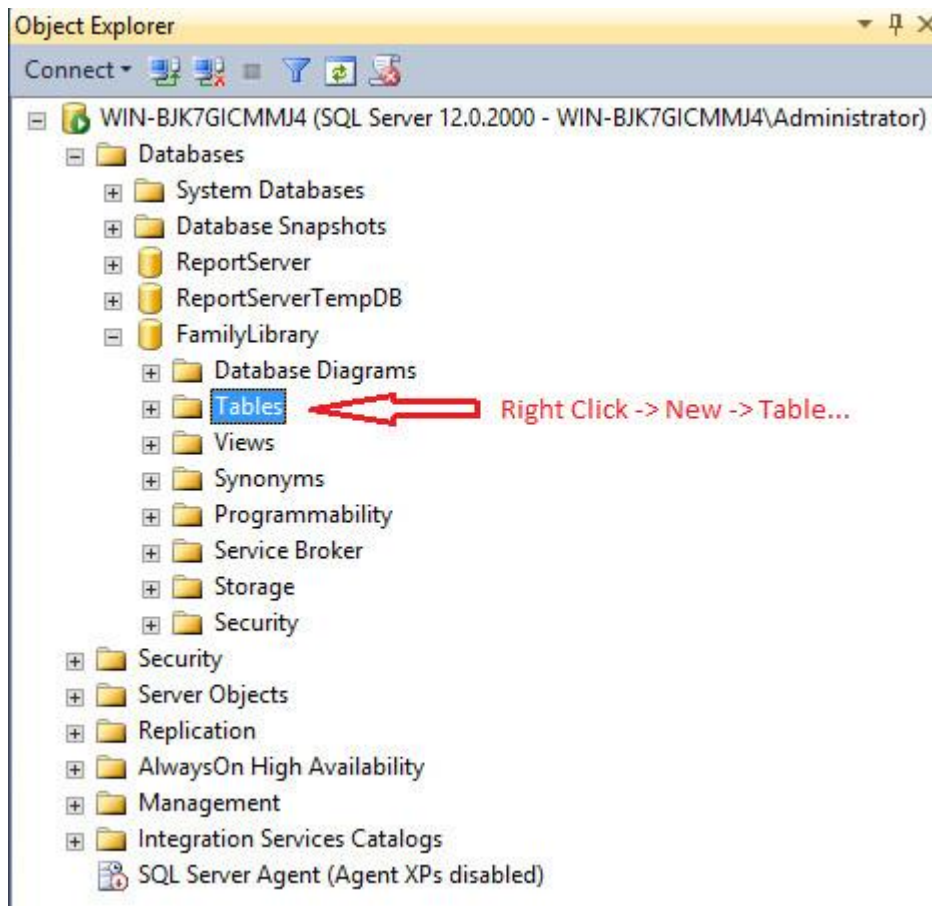


Figure 31. Creating the New Table with SQL Server Management Studio

Now the table editor opens and adding columns can be started.

The screenshot shows the Table Designer window for a new table named 'Table_1'. The table has the following columns and data types:

| Column Name | Data Type | Allow Nulls |
|-------------------------|-------------|-------------------------------------|
| User_id | int | <input type="checkbox"/> |
| Person1_name | varchar(50) | <input type="checkbox"/> |
| Person1_number | varchar(50) | <input type="checkbox"/> |
| Person2_name | varchar(50) | <input checked="" type="checkbox"/> |
| Address | varchar(50) | <input type="checkbox"/> |
| Phone_number | varchar(50) | <input type="checkbox"/> |
| E_mail_address | varchar(50) | <input type="checkbox"/> |
| Pets | bit | <input checked="" type="checkbox"/> |
| Allergies | varchar(50) | <input checked="" type="checkbox"/> |
| Children_amount | int | <input checked="" type="checkbox"/> |
| Positive_recommendation | int | <input checked="" type="checkbox"/> |
| Negative_recommendation | int | <input type="checkbox"/> |

Figure 32. Adding Columns to the New Table Created with SQL Server Management Studio

SQL Server Management Studio gives a default name of “Table_1” to a new table and that name can be changed when the table is saved.

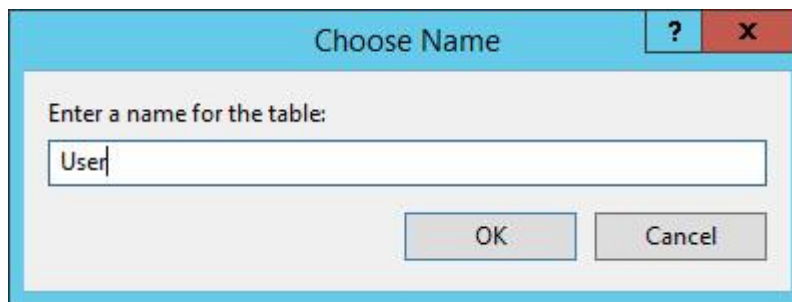


Figure 33. Choosing of the Name When New Table Is Saved with SQL Server Management Studio

A user_id column is meant to be as a primary key. Primary key setting is done by right clicking the User_id column and choosing “Set Primary Key”.

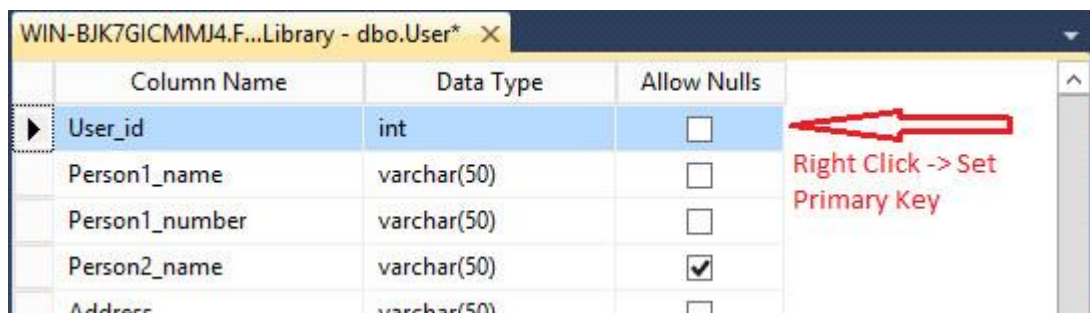


Figure 34. Primary Key Setting with SQL Server Management Studio

When the primary key setting is successfully done, a key icon can be seen in front of the User_id column.

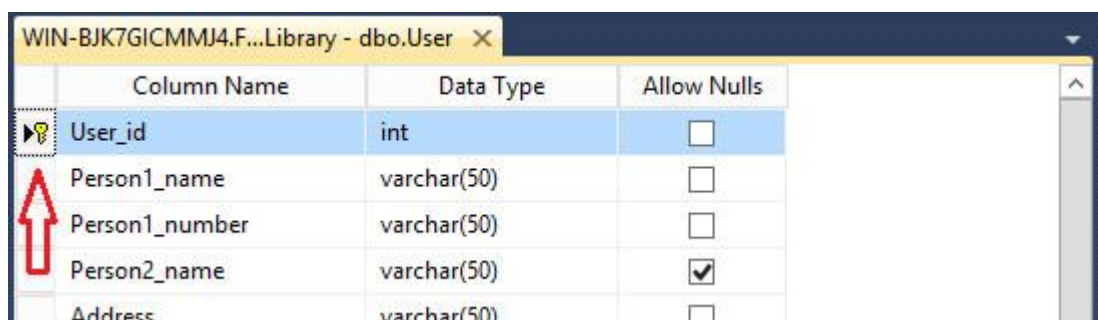


Figure 35. User_id Column is Set to Be as a Primary Key

After the table is saved it can be seen in the Object Explorer. The object Explorer has to be refreshed first.

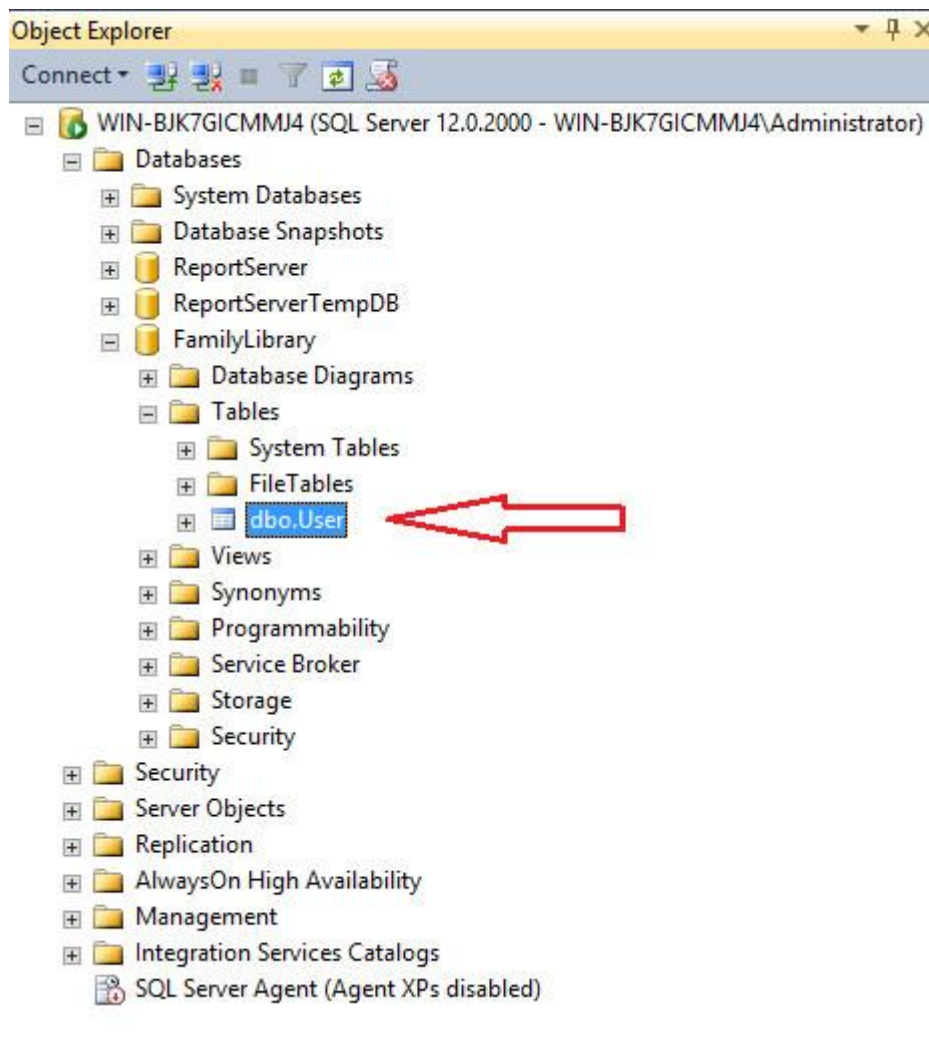


Figure 36. The New Created User Table Can Be Seen in Object Explorer of SQL Server Management Studio

3.7 Implementation of the FamilyLibrary Service

The implementation of the FamilyLibrary service is not covered in this thesis. The implementation is the second phase of this project. It could be done by some other enthusiastic student writing their thesis. If the needed database is done in-premises it will need a service provider. A service provider is also needed if bank account identification is going to be used.

Summary

The topic of the thesis was to design a service for occasionally needed and offered child care help and seeking friend families. The database design was the greatest part of the thesis. The service was named FamilyLibrary and that name was kept all along during designing. During designing it was defined what kinds of features are needed for the FamilyLibrary service and what kind of database design is needed for those features. The outcome of the feature and database design can be found in this thesis.

It seems that SQL Server Management Studio is an easier and lighter way to manage a database and tables compared with doing the same with Visual Studio 2013.

The implementation of the FamilyLibrary service was not included in this thesis. During the design it was seen that implementation is not an easy case. The use of a bank account identification itself requires a great deal of work and also contracts with several banks. Without any financing that might be impossible to implement. In any case this thesis can be continued by creating the service itself. This thesis can be exploited when the implementation work is started.

During the feature design it seemed that more features could be added all the time but the number had to be limited to those which are introduced in the thesis. The first version of the software can be implemented by using this thesis. It is a normal situation for any application that corrections are needed and new features will be implemented. So new versions of the software will appear. The designed features are the main features which are needed for giving or getting occasional child care help or seeking friend families.

The database design was done to correspond with the feature design. The designed features will save data to the database. The designed database is also only the first version of the database. When new features are designed later on, the database design might need updates at that point.

It could be said that this thesis is version 1 design of the FamilyLibrary service.

References

- 1 Transact-SQL Reference. 2014. Web document. Microsoft. <[http://technet.microsoft.com/en-us/library/aa299742\(v=sql.80\).aspx](http://technet.microsoft.com/en-us/library/aa299742(v=sql.80).aspx)>. Updated 27 Nov 2014. Read 27 Nov 2014.
- 2 Microsoft. 2014. SQL Server 2014 & the Data Platform. Windows_Server_2014_Datasheet.pdf.
- 3 FINLEX ® - Ajantasainen lainsäädäntö: 22.4.1999/523. 1999. Web document. FINLEX ®. <<http://www.finlex.fi/fi/laki/ajantasa/1999/19990523#L1>>. Updated 27 Nov 2014. Read 27 Nov 2014.
- 4 FK/Finassialan Keskusliitto. 2011. Pankkien Tupas-tunnistuspalvelun tunnistusperiaatteet v2.0b 28.3.2011. Tupas-tunnistusperiaatteet_v20b.pdf
- 5 Download Microsoft SQL Server 2014 Express. 2014. Web document. Microsoft. <<http://msdn.microsoft.com/en-us/evalcenter/dn434042.aspx>>. Updated 12 Apr 2014. Read 4 Dec 2014.
- 6 Kompozure. 2011. Mikä on Azure, ja miten yrityksestä voi hyötyä siitä? versio 1.0 – 14.11.2011. Kompozure-What_Is_Azure-1.0.pdf
- 7 What is Azure? 2014. Web document. Microsoft. <<http://azure.microsoft.com/en-us/overview/what-is-azure/>>. Updated 12 Oct 2014. Read 10 Dec 2014.
- 8 SQL Database. 2014. Web document. Microsoft <<http://azure.microsoft.com/en-us/services/sql-database/>>. Updated 12 Oct 2014. Read 10 Dec 2014.

Basic SQL Commands

We can use a following test database for example when basic SQL commands are introduced. This is not exactly the same database which is used in the FamilyLibrary service.

| User Number | User Name | Area | Address | Phone Number | E-mail Address | Sex | Pets | Number of Own Kids | Age of Own Kids | Number of Other Kids | Age of Other Kids |
|-------------|-------------------|---------------------|--------------------------------------|---------------|--|--------|------|--------------------|-----------------|----------------------|-------------------|
| 1 | Jane Smith | Kirkkonummi Central | TheFirstStreet 1, 02400 Kirkkonummi | +358401234567 | jane.smith@somemail.com | female | 1 | 2 | 4,6 | 2 | 2-8 |
| 2 | John Smith | Kirkkonummi Central | TheFirstStreet 1, 02400 Kirkkonummi | +358407654321 | john.smith@somemail.com | male | 1 | 2 | 4,6 | 2 | 2-8 |
| 3 | Maija Meikäläinen | Kirkkonummi North | TheSecondStreet 2, 02400 Kirkkonummi | +358401726354 | majja.meikalainen@so.meothermail.com | female | 0 | 1 | 7 | 2 | 5-9 |

Figure 37. Table of test database (User table)

SELECT Retrieves rows from the database and allows the selection of one or many rows or columns from one or many tables. The full syntax of the SELECT statement is complex, but the main clauses can be summarized as: [1]

```
SELECT select_list
[ INTO new_table ]
FROM table_source
[ WHERE search_condition ]
[ GROUP BY group_by_expression ]
[ HAVING search_condition ]
[ ORDER BY order_expression [ ASC | DESC ] ]
```

[] = Optional

Example:

```
SELECT user_name
FROM User
WHERE pets = 0;
```

This returns all user names from user table which don't have pets.

FROM Specifies the tables, views, derived tables, and joined tables used in DELETE, SELECT, and UPDATE statements. [1]

Syntax

[FROM { < table_source > } [,...n]]

Arguments

<table_source>

Specifies a table or view, both with or without an alias, to use in the Transact-SQL statement. A maximum of 256 tables can be used in the statement.

WHERE Specifies the condition for the rows returned by a query. [1]

Syntax

WHERE < search_condition >

Arguments

<search_condition>

Defines the condition to be met for the rows to be returned. There is no limit to the number of predicates in <search_condition>.

This clause is optional and when it is used it specifies which data values will be returned or displayed. The criteria will be specified after the keyword where. With where conditional selections are used.

- = Equal
- > Greater than
- < Less than

- >= Greater than or equal
- <= Less than or equal
- <> Not equal to

Example from User table:

```
SELECT user_name FROM User WHERE sex = 'female';
```

This example returns all female persons from the User table.

LIKE

Determines whether or not a given character string matches a specified pattern. A pattern can include regular characters and wildcard characters. During pattern matching, regular characters must exactly match the characters specified in the character string. Wildcard characters, however, can be matched with arbitrary fragments of the character string. Using wildcard characters makes the LIKE operator more flexible than using the = and != string comparison operators. If any of the arguments are not of character string data type, Microsoft® SQL Server™ converts them to character string data type, if possible.

Syntax

```
match_expression [ NOT ] LIKE pattern [ ESCAPE escape_character ]
```

Arguments

match_expression

Is any valid SQL Server expression of character string data type.

pattern

Is the pattern to search for in *match_expression*, and can include these valid SQL Server wildcard characters.

| Wildcard character | Description | Example |
|--------------------|-------------|---------|
|--------------------|-------------|---------|

| | | |
|----------------|--|--|
| % | Any string of zero or more characters. | WHERE title LIKE '%computer%' finds all book titles with the word 'computer' anywhere in the book title. |
| _ (underscore) | Any single character. | WHERE au_fname LIKE '_ean' finds all four-letter first names that end with ean (Dean, Sean, and so on). |
| [] | Any single character within the specified range ([a-f]) or set ([abcdef]). | WHERE au_lname LIKE '[C-P]arsen' finds author last names ending with arsen and beginning with any single character between C and P, for example Carsen, Larsen, Karsen, and so on. |
| [^] | Any single character not within the specified range ([^a-f]) or set ([^abcdef]). | WHERE au_lname LIKE 'de[^i]%' all author last names beginning with de and where the following letter is not i. |

Table 1. Wildcard Characters when LIKE is used

escape_character

Is any valid SQL Server expression of any of the data types of the character string data type category. *escape_character* has no default and must consist of only one character.

Result Types

Boolean

Result Value

LIKE returns TRUE if the *match_expression* matches the specified *pattern*. [1]

Some example from "User" table:

```
SELECT user_name FROM User WHERE address LIKE 'TheFirst%';
```

This SQL statement will match any user names with matching address. In this case it lists all users whose address starts with "TheFirst" and rest of the letters can be what ever.

CREATE TABLE

This statement is used to create a new table. The format is following: [2]

```
CREATE TABLE "tablename" ("column1" "data type", "column2" "data type");
```

Optional constraints can be used also:

```
CREATE TABLE "tablename" ("column1" "data type" [constraint], "column2" "data type" [constraint]);
```

[] = optional

In this case the User table will be created like this:

```
CREATE TABLE User ( usernumber, username, area, address, phonenumber, emailaddress, sex, pets, numberofownkids, ageofownkids, numberofotherkids, ageofotherkids );
```

If constraints are used, table creation can be like this:

```
CREATE TABLE User (user_number number(5),
                    user_name varchar(30),
                    area varchar(30),
                    address varchar(50),
                    phone_number varchar(15),
                    email_address varchar(40),
                    sex Boolean,
                    pets Boolean,
                    number_of_own_kids number(2),
                    age_of_own_kids varchar(20),
                    number_of_other_kids number(2),
                    age_of_other_kids varchar(20));
```

INSERT / INSERT INTO The insert is used to insert or add a row of data into a table. To insert records into a table the followed format is used:

```
INSERT INTO "table_name" (first_column,... last_column)
values (first_value,... last_value);
```

Take a look in used test database:

```
INSERT INTO User
(usernumber, username, area, address, phonenumber,
emailaddress, sex, pets, numberofownkids, ageofownkids,
numberofotherkids, ageofotherkids)
values ( 00001,
        'Jane Smith',
        'Kirkkonummi Central',
```

'TheFirstStreet 1, 02400 Kirkkonummi',
'+358401234567',

1,

1,

2,

'4,6',

2,

'2-8');

Common SQL Datatypes in Microsoft SQL Server

Each column, local variable, expression and parameters has a related data type. That data type is an attribute that specifies the type of data that the object can hold. The type of data can be integer, character, money, and so on. SQL Server supplies a set of system data types that define all of the types of data that can be used with SQL Server.

In Microsoft SQL Server also user-defined data types, which are aliases for system-supplied data types, can be defined. [1]

System data types:

- **Exact Numerics**

- **Integer**

- (a) bigint

Integer (whole number) data from -2^{63} (-9,223,372,036,854,775,808) through $2^{63}-1$ (9,223,372,036,854,775,807).

- (b) int

Integer (whole number) data from -2^{31} (-2,147,483,648) through $2^{31} - 1$ (2,147,483,647).

- (c) smallint

Integer data from -2^{15} (-32,768) through $2^{15} - 1$ (32,767).

- (d) tinyint

Integer data from 0 through 255.

- **bit**

- (a) bit

Integer data with either a 1 or 0 value.

- **decimal and numeric**

- (a) decimal

Fixed precision and scale numeric data from $-10^{38} + 1$ through $10^{38} - 1$.

(b) numeric

Functionally equivalent to **decimal**.

– **money and smallmoney**

(a) money

Monetary data values from -2^{63} (-922,337,203,685,477.5808) through $2^{63} - 1$ (+922,337,203,685,477.5807), with accuracy to a ten-thousandth of a monetary unit.

(b) smallmoney

Monetary data values from -214,748.3648 through +214,748.3647, with accuracy to a ten-thousandth of a monetary unit.

– **Approximate Numerics**

(a) float

Floating precision number data with the following valid values: - $1.79E + 308$ through $-2.23E - 308$, 0 and $2.23E + 308$ through $1.79E + 308$.

(b) real

Floating precision number data with the following valid values: - $3.40E + 38$ through $-1.18E - 38$, 0 and $1.18E - 38$ through $3.40E + 38$.

– **datetime and smalldatetime**

(a) datetime

Date and time data from January 1, 1753, through December 31, 9999, with an accuracy of three-hundredths of a second, or 3.33 milliseconds.

(b) smalldatetime

Date and time data from January 1, 1900, through June 6, 2079, with an accuracy of one minute.

• **Character Strings**

- char
Fixed-length non-Unicode character data with a maximum length of 8,000 characters.
- varchar
Variable-length non-Unicode data with a maximum of 8,000 characters.
- text
Variable-length non-Unicode data with a maximum length of $2^{31} - 1$ (2,147,483,647) characters.
- **Unicode Character Strings**
 - nchar
Fixed-length Unicode data with a maximum length of 4,000 characters.
 - nvarchar
Variable-length Unicode data with a maximum length of 4,000 characters. **sysname** is a system-supplied user-defined data type that is functionally equivalent to **nvarchar(128)** and is used to reference database object names.
 - ntext
Variable-length Unicode data with a maximum length of $2^{30} - 1$ (1,073,741,823) characters.
- **Binary Strings**
 - binary
Fixed-length binary data with a maximum length of 8,000 bytes.
 - varbinary
Variable-length binary data with a maximum length of 8,000 bytes.
 - image
Variable-length binary data with a maximum length of $2^{31} - 1$ (2,147,483,647) bytes.

- **Other Data Types**

- cursor

- A reference to a cursor.

- sql_variant

- A data type that stores values of various SQL Server-supported data types, except **text**, **ntext**, **timestamp**, and **sql_variant**.

- table

- A special data type used to store a result set for later processing.

- timestamp

- A database-wide unique number that gets updated every time a row gets updated.

- uniqueidentifier

- A globally unique identifier (GUID).

Contents of Tables

In this appendix all parameters of tables are introduced.

User Table

In this section all columns of a User table are introduced.

User_Id

User_Id is an individual Id number for each family registered to the FamilyLibrary service. Later on families can be identified by using this number. Relationship between tables are done by using this id. This column is a unique key which identify the family.

Person1_name

This column is for a name of the adult member of the family. Different families are noticed at this part of the table by not to use column names such as mother_name or father_name. This column can't be empty because every family must have at least one adult member.

Person1_number

This column is for the personal identity number of the adult member of the family. The personal identity number is never shown to other families. At this column different kind of families are noticed by not to use a column names such as mother_number or father_number. This column can't be empty because every family must have at least one adult member.

Person2_name

This column is for a name of the second adult member of the family. This column can be empty because there is families with one adult member only.

Address

This column is for an address where the family lives. If family care help is offered on some other address than on home address of the family, that address can be added here. This column is used when a child care help is searched from some specific area. This column can't be empty. The home address is not seen by other users. It is only used by search tools. This is cause of security reasons.

Phone_number

This column is for a phone number from where the family can be contacted. It is good if families can be contacted via a phone when a child care help is needed. More detailed information can be asked and discussed for details. Maybe also pre-visiting can be planned for getting to know the family. The phone_number column can be left empty cause of security reasons. Some families have for some reason secret phone number. The first contact with the family can be done by using e-mail.

E_mail_address

This column is for an e-mail address from where the family can be contacted. It is good if the families can be contacted by not only via a phone but also via e-mail when child care help is needed. Sometimes it is easier to make the first contact with the e-mail. More detailed information can be asked and discussed for details. Maybe also pre-visiting can be planned for getting to know the family. The e_mail_address column can't be empty.

Password

This column is for a password which the user have been selected.

Pets

This column is for information, which tells does the family have any pets. It is not identified what pets the family have. That information is a detail which can be asked when two families contacted with each other. If this column is empty that means that the family have no pets.

Allergies

This column tells if there are some allergies on the family. This column can be empty if there is no allergies on the family.

Children_amount

This column tells the amount of the children of the family. This column can be empty when there is no children on the family.

Positive_recommendation

This column tells how many positive recommendations the family have got. In other words, how many times "+" button of this family have been pressed.

Negative_recommendation

This column tells how many negative recommendations the family have got. In other words, how many times "-" button of this family have been pressed.

Calendar Table

In this section all columns of a Calendar table are introduced. The calendar table is used for collecting times when the family can give or need the child care help.

User_Id

With this column the family is identified and relationship to the User table is defined.

Give_need

This column tells is the row for time for needed or given help. If the column is empty it means that row is used for time of given child care help.

Year

This column tells the year when the child care help is given or needed. This column can't be empty.

Month

This column tells the month when the child care help is given or needed. This column can't be empty.

Day

This column tells the day when the child care help is given or needed. This column can't be empty.

Time

This column tells the time of the day when the child care help is given or needed. This column can't be empty.

Children Table

In this section all columns of a Children table are introduced. There might be families with no children and they will give some child care help. Maybe own children has grown up and moved away and parents will now help other families. Those families doesn't have relationship from the User table to the Children table. Every child of the family have their own row on the Children table.

User_Id

With this column the family is identified and relationship to the User table is defined.

Age

This column tells an age of the child. This column can be empty if the column amount tells that there is no children on the family.

Gender

This column tells the gender of the child. This column can be empty if the column amount tells that there is no children on the family.

Allergies

This column tells if the child have some allergies. This column can be empty if the child have no allergies.

Children to take care Table

In this section all columns of a Children_to_take_care table are introduced. If the family will only need a child care help, not to give it, then there is no relationship from the User table to this table for that family. When the family informs that they can give some child care help they can tell also some information about what kind of child care they are able to give. For example how many children they can take care at once and what ages children can be.

User_Id

With this column the family is identified and relationship to the User table is defined.

Amount

This column tells how many children the family is able to take care at once. This column can be empty. It can be the case that amount of children the family is able to take care is dependent on ages of children. Then the exact amount is not known.

Age

This column tells the range of the age of children the family is able to take care at once. This column can't be empty.

Gender

This column tells the gender of the children to take care. This is not meant to discriminate some gender but some cases it might be reasonable to offer child care help only for example for girls. Maybe the family have only girls itself and it might be easier if visitor child is the same gender. This column can't be empty. It can have value boy, girl or both.

Allergies

This column tells if the family is able to give child care help also for children which have some allergies. Of course detailed information will be clarified by a phone or an e-mail. This column can't be empty. It can have values yes or no.