ISM 6124.901 Fall 2017

Group Project No. 20

KeePass

BY:

Leonardo Deartiaga

Charu Mathur

Janvi Vora

TABLE OF CONTENTS

INTRODUCTION

KeePass is an OSI Certified Open Source Software distributed under the terms and conditions of GNU General Public License Version 2. The problem of memorizing password is resolved by KeePass. The feature of KeePass is to keep all the user's passwords, data, emails account login, and the URL stored in an encrypted database, secured and protected by Master Password. The best is that the system is small enough to be transferred from one computer to another. One of the feature helps the user to keep the database secured by Master Password and no backup is needed if the data is lost.

OBJECTIVES

A number of key characteristics of a password manager are very important for securely managing passwords and need to be kept in mind before designing an application for the same:

1. **Encryption:**
   A password manager should implement strong encryption techniques using algorithms so that the user credentials are stored in an encrypted form. Even if the device used to store the passwords is stolen or hacked the thief/hacker is unlikely to be able to break into the system and retrieve or recover the user's passwords. KeePass uses AES encryption to encrypt its password databases, SHA-256 password hash, protection against dictionary and guessing attacks, in-memory protection.

2. **Self-contained functionality:**
   A lot of software is not written with absolute data security in mind thus any password management software should not trust the security of outside applications. The security of such a system is questionable if the decrypted passwords would just be passed directly to another application that stores data in tempfiles and may never be explicitly deleted. A successful password manager should have a self-contained functionality instead of relying on outside applications.

3. **Usability:**
   The password manager should be Quick, simple, and easy use for the day-to-day tasks of a user. Along with increasing user ease and satisfaction, it would also ensure that the password manager is used regularly. A complex and less usable password manager may get neglected in favor of less secure options by a user making its purpose redundant.

4. **Interoperable & Extensible:**
   Interoperability is extremely important for any system is with the different platforms available on our desktops, phones, and iPad. A password manager should be compatible for use with the different operating systems on PC (Windows, Linux, Mac OS X) and with Android for phones along with iPhone and iPad i.e. apple products. The data stored

in the database should also be compatible for import and export with all the different file formats - TXT, HTML, XML, and CSV. Language is also a key criterion which makes a password manager more usable if it is available in multiple languages and is thus extensible.

5. **Portable:**
   A password manager should be portable so that the user can have access to their different credentials for every website even when they are on the go and travelling. User access cannot be dependent on the system they are using for desktop access. These managers can even work on smartphones and tablets. KeePass can be carried on a USB stick and can be run or directly installed on Windows if user prefers desktop access.

6. **Verifiable design:**
   One can only trust an application that manages the data used to access other applications if it is verifiable which means that not only is the source available for scrutiny, but verifiably the same as the source code used to produce the actual executable program itself. This would ensure the user's trust in the software. This may be termed as Security through visibility which often requires open source softwares.

7. **Secure resource usage:**
   A number of possible vulnerabilities involving unsecured resource usage are possible such as using a secure memory that will not be written to a page file or swap partition on disk guards against the danger of a decrypted password being dumped onto the disk where it can be recovered later by a malicious security cracker.

REQUIREMENTS

1. Provide simple UI to the user to create a master key for all accesses and also for creating a database, database groups and database subgroups.

2. All system-level passwords (for example, root, enable, NT admin, application administration accounts, and so on) must be changed on at least a quarterly basis.

3. All user-level passwords (for example, email, web, desktop computer, and so on) must be changed at least every six months. The recommended change interval is every four months.

4. The User should have an option to alternatively use key files instead of the master password. Key files provide better security than master passwords in most cases

5. A user should have access to a password database that consists of only one file that can be transferred from one computer to another easily.

6. The entire system and not just the password should be encrypted. Similarly, the complete database should be encrypted and not only the password fields.

7. A user should be able to can create, modify and delete groups, in which passwords can be sorted.

8. The system should provide features such as Auto-Type for frequently visited websites and softwares.

9. There should be proper session tracking such that if a password is copied, it remains in memory for only 10 seconds

10. The password management application should also track windows i.e. when a function is performed like adding, editing or deleting at this time the main database window would be inactive and cannot be accessed unless the currently active window is closed.

11. A repair functionality should help users repairing the database file if the database contained on a USB is damaged due to untimely removal of the USB and cannot be opened

12. Regular backups of the user passwords and databases should be taken

13. The password manager should be able to generate strong encrypted random passwords for the user to be used in different websites and applications

14. The system should automatically renew the user passwords at regular intervals of time and also provide the user with notifications for periodic password changes.

15. The system should always verify the user’s identity before resetting any password

FEATURES

There are multiple key features of KeePass for which it is known. They are as follows:

1. Strong Security

- KeePass uses two secure support Advanced Encryption Standard (AES, Rinjdael) and the Two fish algorithm to protect its databases. Both of these are approved by National Security Agency (NSA).
- The whole database including username, notes are encrypted and not only the password.
- To hash the master key component SHA-256 is used, as it is a 256 bit-cryptographically secure single hash function. SHA-256 is free from attacks. A key derivation function is used to transform the output.
- On the desktop, the secure master key dialogue is shown and there no keylogger works.
- KeePass can get the security password edited and it's the first password manager for it.
- If the operating systems dump the KeePass process to disk, still the passwords are encrypted while KeePass is running and passwords are not revealed.

2. Multiple User Keys

- Complete database can be decrypted with one master password.
- Key files can be used in alternative to the master passwords in most cases, as key files have better security. Key files can be carried in such as floppy disk, USB stick or a CD.
- To have a stronger security both of the above methods can be used as the database requires a key file and the password to get it unlocked. Thought the key file will be lost, the database will be secured.
- The database can be locked by a current Windows user and it can be opened by the same person who created it.

3. Portable and No Installation Required

- It is easy to carry on a USB drive and can run on Windows without being installed.
- The installer packages are available and can create shortcuts on Windows start menu and the desktop.
- Ports for a system like Android and iOS are available.
- The best part about KeePass is that it does not store anything on the system and does not create any initialization files in the Windows directory or any create any registry keys.
- Even if KeePass directory is deleted, still no traces of KeePass are found on the system.

4. Export to TXT, HTML, XML and CSV files

- The password list can be set up in various formats such as HTML, TXT, CSV and XML.
- The XML output can be applied in various applications.
- The cascading style sheets (CSS) are used by HTML to format the tables, so the layout can be easily changed.
- The othe password safes such as the closed-source Password Agent and the commercial closed-source Password Keeper are fully compatible with CSV output.
- The other file formats are properly supported by KeePass plugins.

5. Import from many file formats

- KeePass uses common CSV export format and exports from Password Keeper and Password agent are easily put in KeePass database.
- KeePass can easily parse and import TXT outputs of CodeWallet Pro, a commercially closed source password safe.
- KeePass can import 35 formats.

6. Support of Password Groups

- The groups can be edited, created and modified in which passwords need to be sorted.
- The groups can be a tree and then divided into multiple subgroups.

7. Open Source

- The KeePass is kept free and will have access to the source code.
- KeePass is open source and it prevents backdoors. The source code can be looked for and then complied together.
- KeePass is OSI Certified Open Source Software and is a mark of open source Initiative.

8. Strong Random Password Generator

- KeePass helps the user to generate strong and random passwords for the users.
- The possible output characters of the generator can be possibly defined

OVERALL SYSTEM ANALYSIS

The following section will include the following:

1. Technical analysis of the KeePass information system.
   - Open Source license
   - Programming language
   - Acknowledgments
2. System history and current status.
   -Version release statistics
   -Download statistics
3. Quality of the system.
   - Bug tracking
   - User satisfaction
   - Knowledge base

1. Technical Analysis:

➢ License:

Keepass has a GPL (General Public License) which gives people consent to review, share, or modify the software. The source files are hosted on SourceForge.com, and available to view by members of the SourceForge community.

➢ Programing Language:

Version 1.x : C++.
Version 2.x : C#

➢ Acknowledgments:

Keepass pulls from several open source classes and libraries which includes the following:

Author	**Class / Library**
Szymon Stefanek	C++ implementation of the AES/Rijndael encryption algorithm.
Niels Ferguson	C implementation of the Twofish encryption algorithm.

Brian Gladman	C implementation of the SHA-2 (256/384/512) hashing algorithms.
Alvaro Mendez	MFC class for validating edit controls (CAMSEdit).
Brent Corkum	MFC class for XP-style menu (BCMenu).
Davide Calabro	MFC class for buttons with icons (CButtonST).
Zorglab, Chris Maunder, Alexander Bischofberger, James White, Descartes Systems Sciences Inc.	MFC class for color pickers (CColourPickerXP).
Peter Mares	MFC class for window side banners (CKCSideBannerWnd).
Chris Maunder	MFC class for system tray icons (CSystemTray).
Hans Dietrich, Chris Maunder	MFC class for hyperlinks in dialogs (XHyperLink).
Lallous	Class for sending simulated keystrokes to other applications (CSendKeys).
PJ Naughter	MFC classes for checking for single instance (CSingleInstance) and version information (CVersionInfo).
Bill Rubin	Command line C++ classes.
Boost Developers	Boost C++ libraries.
Silktide	Cookie Consent JavaScript plugin.

Mark Burnett	List of 10000 Top Passwords, which KeePass uses in its password quality estimation algorithm.

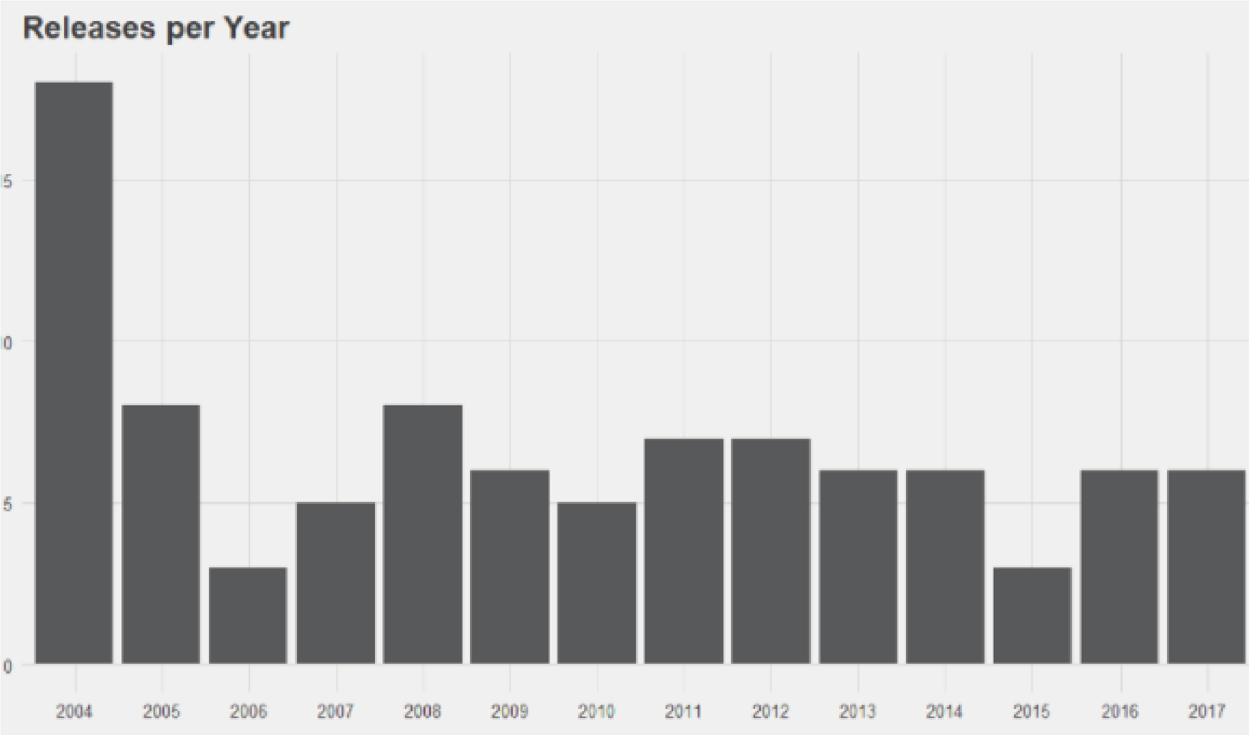
Table taken from https://keepass.info/help/base/credits.html

2. System History, and Current Status:

➢ History:

- Keepass was initially released on November 16, 2003.
- The oldest release of Keepass available on the sourceforge website is version 0.86.
- Version 0.86 was released on January 1, 2004
- Since version 0.86 there has been 94 number of version updates. This includes production, alpha, and beta releases. The alpha and beta releases were released for testing purposes.
- Of the 94 releases, 7 were alpha or beta releases.

The chart below shows the number of releases throughout the life of the KeePass project:



- The project has been releasing updates on a consistent basis for 14 years.
- This shows that the owners of the project are committed to keeping the software system healthy and functioning. The project averages 6.714286 releases per year.

- **Current Status:**

6,613,443 downloads year to date (11/5/2017).

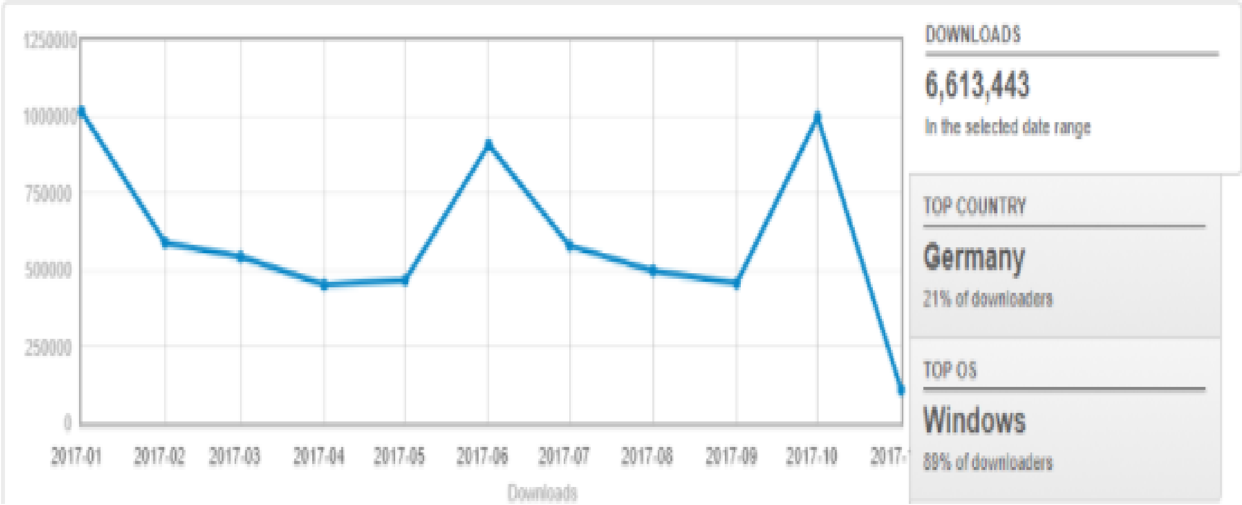


Chart taken from https://sourceforge.net/projects/keepass/files/stats/timeline

3. Quality of the Information System:

The criteria used to measure the quality of KeePass:

Bug Tracking:

- Bugs are tracked and resolved in a timely fashion. Below you will see the ticket statistics found at SourceForge.
- tickets: 1607

open tickets: 17

closed tickets: 1590

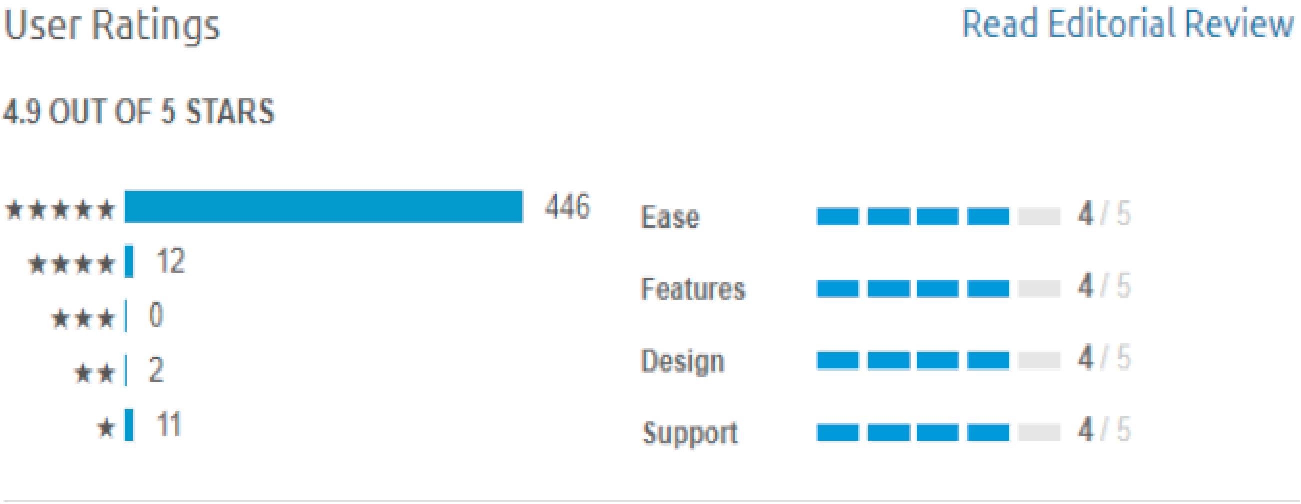
- **New tickets in the last...**
  - 7 days: 1
  - 14 days: 5
  - 30 days: 13

- # of comments on tickets: 7084
- # of new comments on tickets in last...
  - 7 days: 5

- 14 days: 35
- 30 days: 80

➢ User Satisfaction:

KeePass has a high user rating of 4.9 out of 5 stars.



User Ratings

Read Editorial Review

4.9 OUT OF 5 STARS

★★★★★	446
★★★★	12
★★★	0
★★	2
★	11

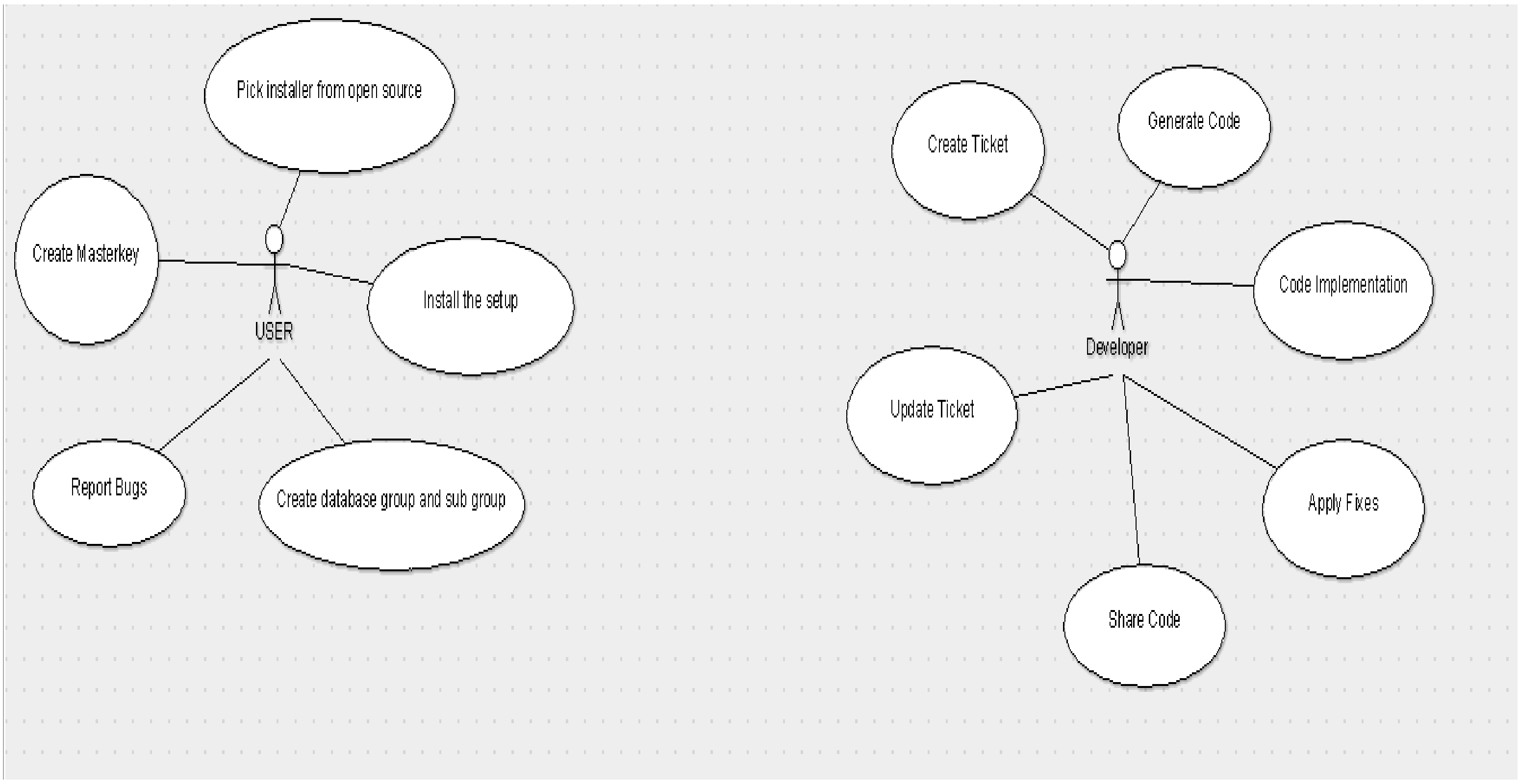
Ease	4 / 5
Features	4 / 5
Design	4 / 5
Support	4 / 5

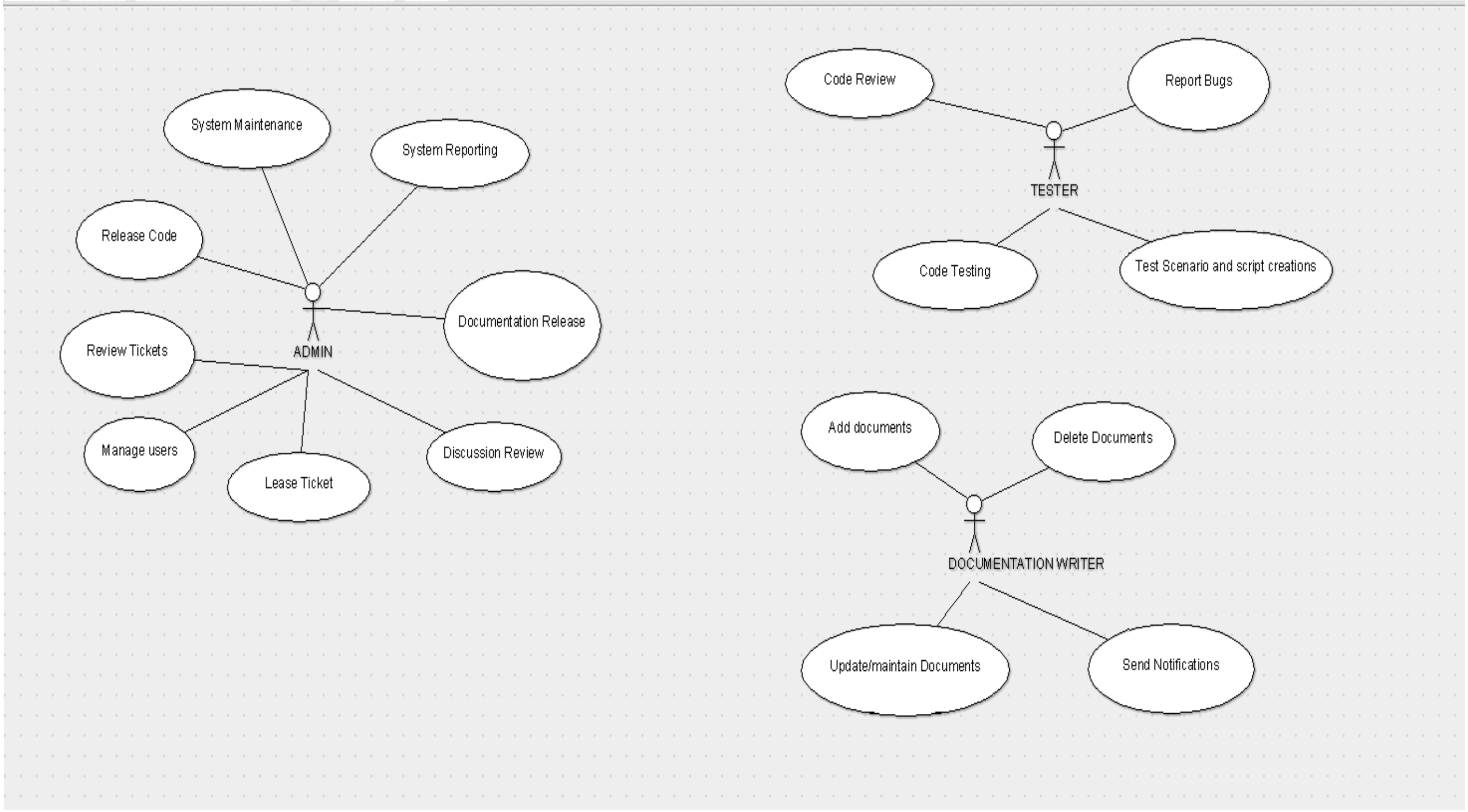
- 446/471 users gave KeePass a 5 star rating. The software is well liked among the community which speaks to the quality of the product.
- Users rate the software system with the following categories: Ease, Features, Design, and Support.

➢ Documentation:

- The KeePass website includes an extensive and easily accessible documentation.
- The site includes a knowledge base, Frequently asked questions, and user tutorials.

USE CASE ANALYSIS





The actors of the Use Case Analysis are as follows:

1. User

The user will be the one to use the system and take the advantage of all the features that KeePass has. The user performs the following function.

- Pick installer from open source
- Install the setup without actually installing it on the Windows
- Create the database groups and sub groups
- Report bugs if there are any problems accesing the database.
- Create Masterkey to keep the database protected.

2. DEVELOPER

The developer is the one who will setup the whole setup and make sure the users are able to use it smoothly. The developer performs the following functions:

- Create ticket
- Generate Code
- Code implementation
- Apply fixes incase of any disruptions in the software.
- Share code
- Apply fixes

3. ADMIN

The admin is the one who looks after the system and makes sure everything is on track and the processing is working good. The admin functions are as follows:

- System Maintenance
- System Reporting
- Documentation Release
- Discussion Review
- Lease Ticket
- Manage users
- Review Tickets
- Release Code

4. TESTER

The tester runs the system on daily basis and make sure it is working Fine. The tester performs the following function:

- ➢ Code Review
- ➢ Report Bugs
- ➢ Test scenario and script creations
- ➢ Code Testing

5. DOCUMENTATION WRITER

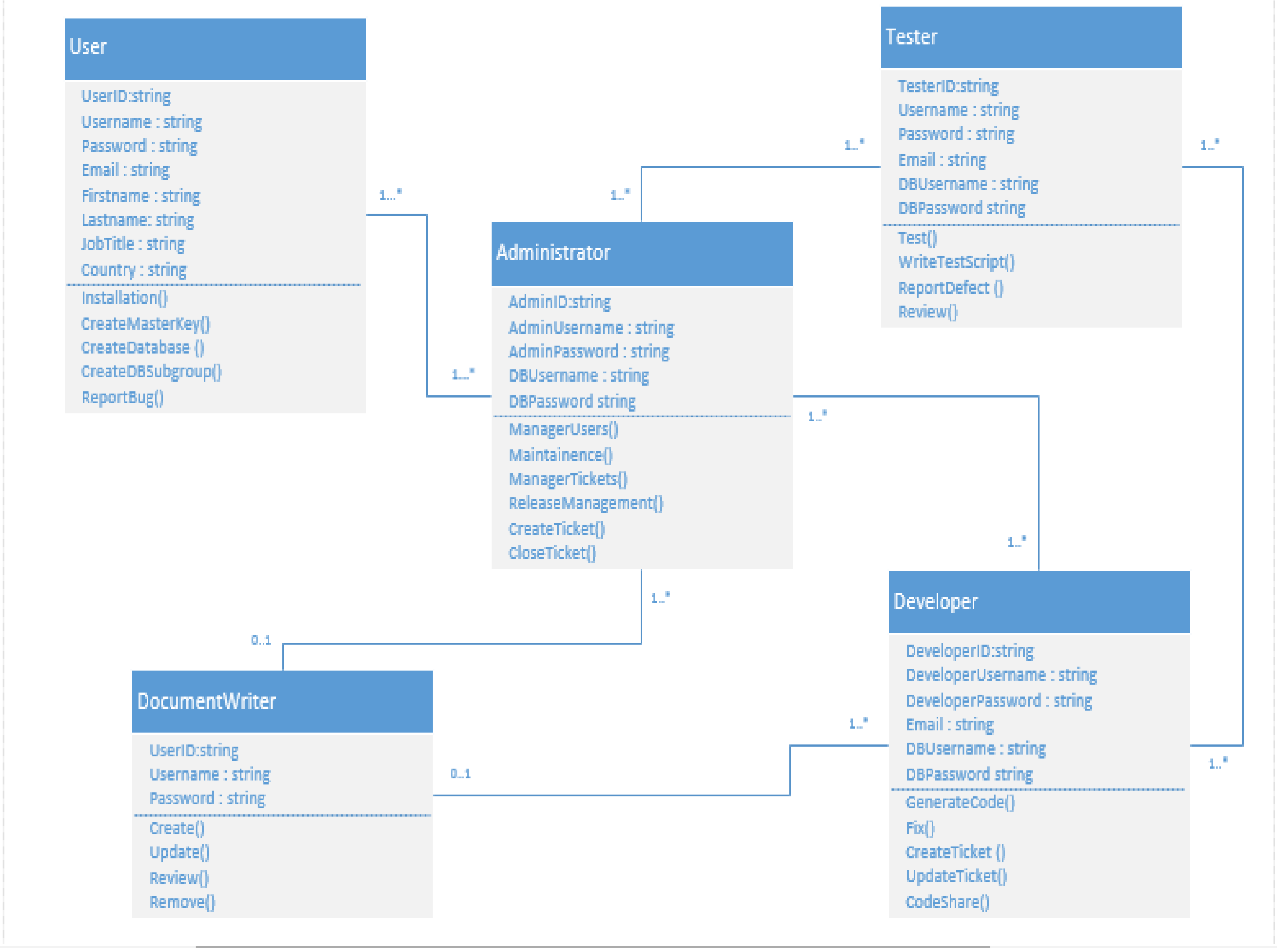
The doucmentation writer plots the system initially and then implements it so that is developed in the same way. The documentation writer performs the following function.

- ➢ Add doucments
- ➢ Delete documents

6. Send Notifications

- ➢ Update/Maintain documents

CLASS DIAGRAM



Below is a description of the classes with their attributes and methods

1. USER

- Attributes

Attribute Name	Data Type	Description
UserID	string	Unique ID is given to user
UserName	string	Preferred name of User in the system
Password	string	Password for the User
Email	string	Contact Email address of the User
Firstname	string	First name of the user
Lastname	string	Last name of the user
JobTitle	string	User Job Title
Country	string	Current country from where the user account has been created

- Methods

Method Name	Description
Installation()	Used to install the Keepass application on user system
CreateMasterKey()	Used by the User to create a Master key for the application
CreateDatabase()	Used to create a database for the user
CreateDBSubgroup()	User to create a sub group in the user database
ReportBug()	Used by the user to report bugs in the application

2. DEVELOPER

- Attributes

Attribute Name	Data Type	Description
DeveloperID	string	Unique ID is given to a developer
DeveloperUserName	string	Name of developer login name
DeveloperPassword	string	Developer Login password
Email	string	Developer contact Email address
DBUsername	string	Developer username for database access
DBPassword	string	Developer password for database access

- Methods

Method Name	Description
GenerateCode()	Used to write new code for new features and functionalities
Fix()	Used to modify existing code to fic defects and bugs
CreateTicket()	Used to create a ticket for tracking of defects or bugs
UpdateTicket()	Used to update an existing ticket
CodeShare()	Used to share the source code for the open source system

3. ADMINISTRATOR

- Attributes

Attribute Name	Data Type	Description
AdminID	string	Unique ID is given to an Administrator
AdminUserName	string	Name of Administrator login name
AdminPassword	string	Administrator Login password
DBUsername	string	Administrator username for database access
DBPassword	string	Administrator password for database access

- Methods

Method Name	Description
ManageUsers()	Used by the Administrator manager different users
Maintenance()	Used to carryout maintenance jobs and outages
CreateTicket()	Used to create a ticket for tracking of defects or bugs
ManageTicket()	Used to manage tickets
CloseTicket()	Used to close tickets that have been actioned upon or rejected
ReleaseManagement()	Used by the Administrator to send out new product versions/ releases

4. TESTER

- Attributes

Attribute Name	Data Type	Description
TesterID	string	Unique ID is given to Tester
UserName	string	Preferred name of Tester in the system
Password	string	Administrator password for database access
Email	string	Contact email address
DBUsername	string	Tester username for database access
DBPassword	string	Tester password for database access

- Methods

Method Name	Description
Test()	Used by testers to test code and functionality
WriteTestScript()	Used by testers to write test scripts and scenarios
ReportDefect()	Used by the testers to report a defect found to the admin and developer
Review()	Used by the tester to review if code fix has fixed the defect in the functionality

5. DOCUMENTWRITER

- Attributes

Attribute Name	Data Type	Description
UserID	string	Unique ID is given to each document writer
UserName	string	Preferred name of Doc writer in the system
Password	string	Administrator username for database access

- Methods

Method Name	Description
Create()	Used to create new documentation for sharing on open source system
Update()	Used to update existing documentation for sharing on open source system
Review()	Used to review documentation for sharing on open source system
Remove()	Used to remove existing documentation for sharing on open source system

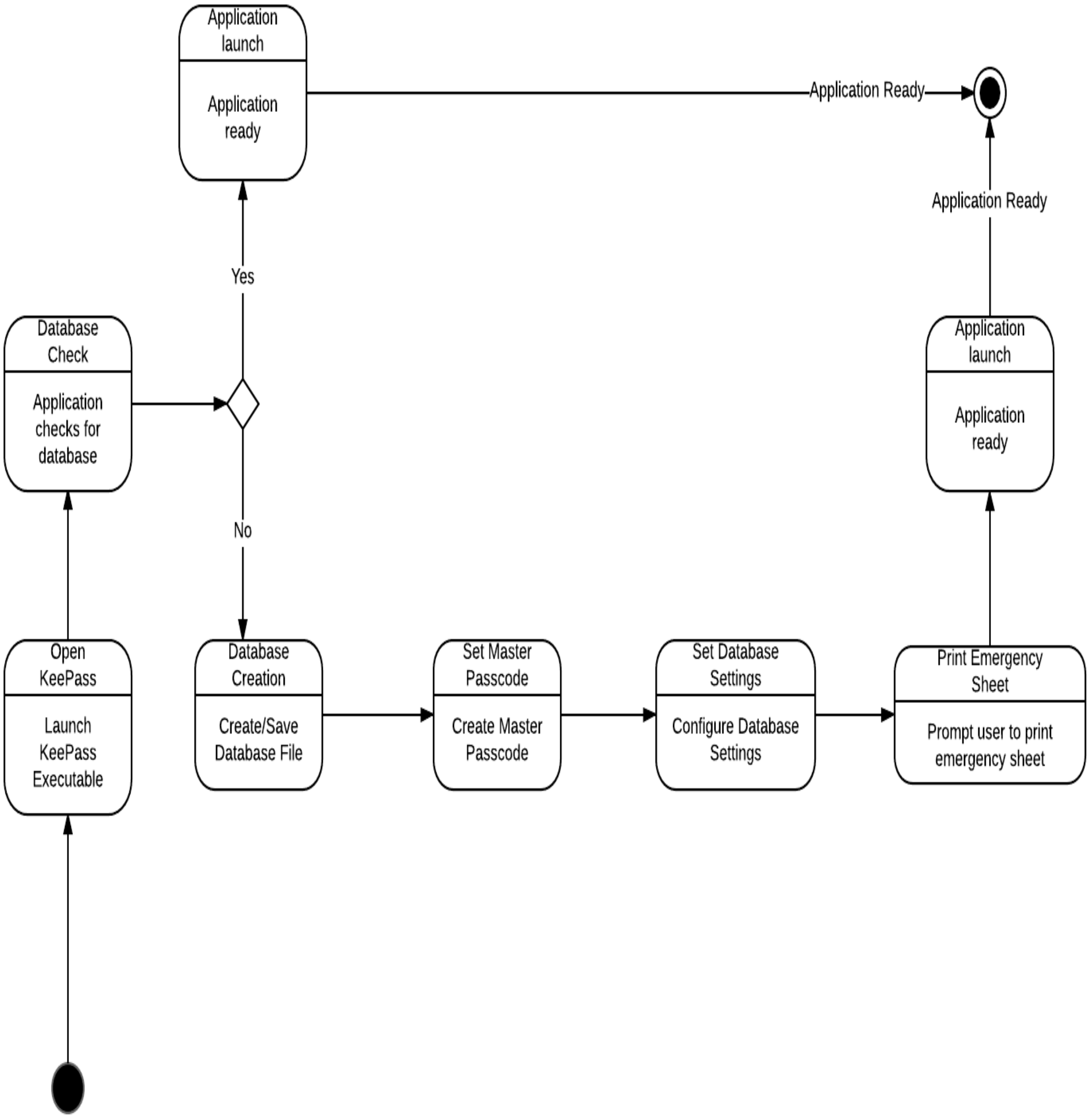
RELATIONS

Entity Relation	Association	Description
User - Administrator	1…* to 1...*	Multiple users can be managed by multiple administrators
Developer - Administrator	1…* to 1...*	Multiple developers can be managed by multiple administrators
Tester - Administrator	1…* to 1...*	Multiple testers can be managed by multiple administrators
Document Writer - Administrator	0…1 to 1...*	Each Administrator can manage one or no document writer
Developer - Testers	1…* to 1...*	Multiple developers and testers can coordinate with the same team
Document Writer - Developer	0…1 to 1...*	Each developer can coordinate with one ot no document writer to share their code

STATE CHART DIAGRAM

The State diagram describes the behavior of KeePass when you launch the KeePass executable.

.



State Chart Description:

1. **Open KeePass –** The keypass executable is initialized by the user of the software.

2. **Database Check –** The KeePass system checks to see if there is a database file.
   Yes: Application Launch – Database file is located, and KeePass system launches.

3. **Application Ready**

4. **No: Database Creation –** The system prompts you to create and save a database file.

5. **Set Master Passcode –** The user will create a master passcode

6. **Set Database Settings –** The user will configure the database settings

7. **Print Emergency Sheet –** System prompts user to print emergency sheet

8. **Application Launch –** KeePass system launches

9. **Application Ready**

PASSWORD MANAGER COMPARISION

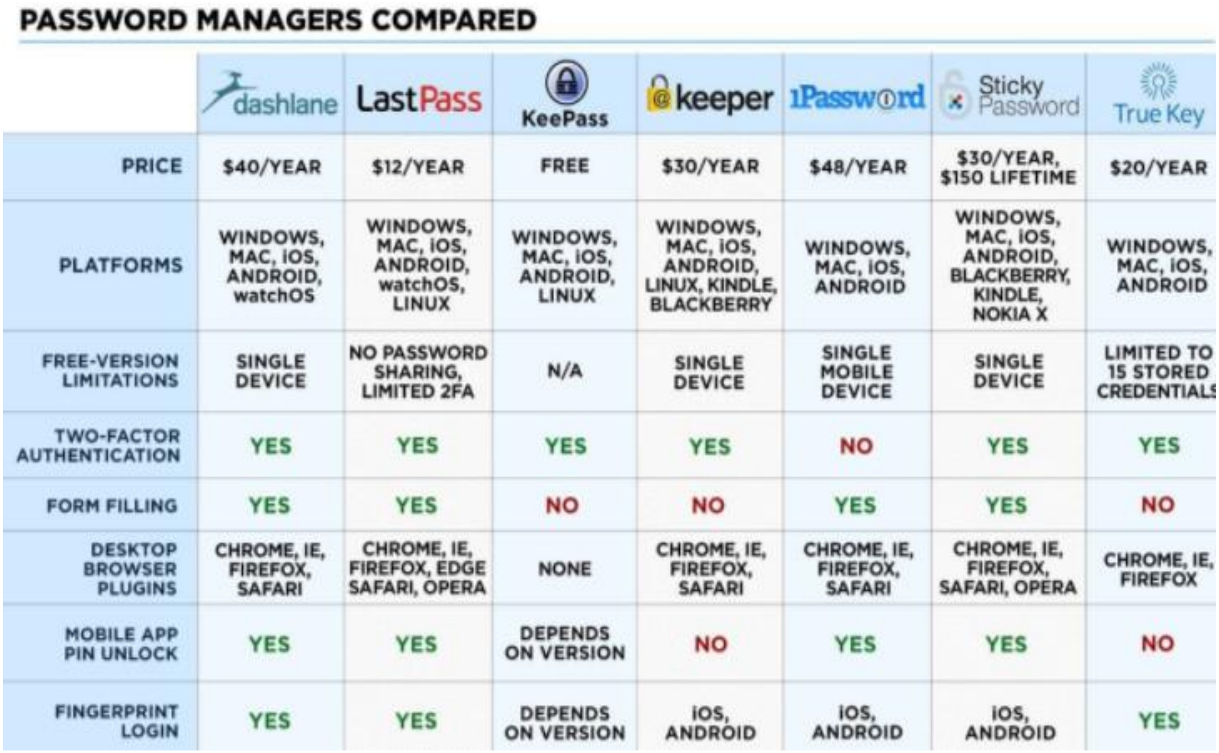
There are multiple password managers out in the market. Some password managers cost money, while others are free. Even the password managers that cost money usually provide basic/limited features for free.

Where KeePass stands out is the fact that it is an open source software that is managed by a small team who accept donations.

Besides the price for the software most of the password managers have similar features, but do they differ in certain aspects. For instance, the password managers might use different encryption algorithms to secure the software. Another difference is that some password managers store the user's passwords on the cloud, and others store them locally on the hard drive.

Most password managers are available on the major operating systems like windows, mac os, android, and linux.

The figure below shows a detailed comparison of the most popular password managers out in the market. The figure was taken from tomsguide.com



PASSWORD MANAGERS COMPARED

	dashlane	LastPass	KeePass	keeper	1Password	Sticky Password	True Key
PRICE	$40/YEAR	$12/YEAR	FREE	$30/YEAR	$48/YEAR	$30/YEAR, $150 LIFETIME	$20/YEAR
PLATFORMS	WINDOWS, MAC, iOS, ANDROID, watchOS	WINDOWS, MAC, iOS, ANDROID, watchOS, LINUX	WINDOWS, MAC, iOS, ANDROID, LINUX	WINDOWS, MAC, iOS, ANDROID, LINUX, KINDLE, BLACKBERRY	WINDOWS, MAC, iOS, ANDROID	WINDOWS, MAC, iOS, ANDROID, BLACKBERRY, KINDLE, NOKIA X	WINDOWS, MAC, iOS, ANDROID
FREE-VERSION LIMITATIONS	SINGLE DEVICE	NO PASSWORD SHARING, LIMITED 2FA	N/A	SINGLE DEVICE	SINGLE MOBILE DEVICE	SINGLE DEVICE	LIMITED TO 15 STORED CREDENTIALS
TWO-FACTOR AUTHENTICATION	YES	YES	YES	YES	NO	YES	YES
FORM FILLING	YES	YES	NO	NO	YES	YES	NO
DESKTOP BROWSER PLUGINS	CHROME, IE, FIREFOX, SAFARI	CHROME, IE, FIREFOX, EDGE SAFARI, OPERA	NONE	CHROME, IE, FIREFOX, SAFARI	CHROME, IE, FIREFOX, SAFARI	CHROME, IE, FIREFOX, SAFARI, OPERA	CHROME, IE, FIREFOX
MOBILE APP PIN UNLOCK	YES	YES	DEPENDS ON VERSION	NO	YES	YES	NO
FINGERPRINT LOGIN	YES	YES	DEPENDS ON VERSION	iOS, ANDROID	iOS, ANDROID	iOS, ANDROID	YES

RECOMMENDATION

1. Sync Support

Sync support means that a user does not need to configure the software manually on all their computers or mobile devices. With the advancement in technology, cloud storage is freely available to all and if the user could configure one and have it set to sync, the data would automatically appear in other devices linked to the account. Aside from convenience, this automation means your passwords are not exposed during transfer between computers.

KeePass maintains a local database, which many prefer to prevent their passwords residing “in the cloud," but they both optionally sync through third-party accounts, such as Dropbox. Although database and password storage in the cloud would pose higher and sever security threats to user data they would also provide more ease of access to the users. The KeePaas security over the cloud can be improved with more encryption and security layers to their system while syncing the local data to the cloud account online.

2. Cloud Sync

KeePass currently does not have a cloud sync feature. Meaning that if the user’s hard drive crashes or if the USB hosting the database file is lost, the database file cannot be replaced. That is why we are proposing a cloud sync feature where the user has the option to either host, or backup the database file on a server hosted by the KeePass team. The feature will add extra trust in the system as the user will trust that his/her passwords will never be truly lost. This feature will take some of the responsibility away from the user, and add it to the KeePass system. The cloud sync feature would be a useful addition to an already sound system.

3. Forget master password option

In the KeePass, the user has to setup a strong Master password in order to protect the database. Once the Master password is set that is always used to login the data. There are issues that has been reported, the users forget their Master password and then it is impossible for them to access the account and use the database.

We recommend, the KeePass should have an option to FORGET THE PASSWORD, just like the other sites. This will help the user to setup another Master password that will be strong password too and access the database again. This will help the user not to stress a lot and remember the password. There can be security questions too, that will help to retrieve the password again. If this one feature is added again, then the KeePass will be leading password protector in the market and because of its easy accessibility it will be used more.

REFERENCES

- https://sourceforge.net/p/keepass//
- https://keepass.info/
- https://www.tomsguide.com/us/best-password-managers,review-3785.html