Software Requirements Specification

for

Makagiga

Version 6.9-beta

Prepared by Dekas Dimitrios

Aristotle University of Thessaloniki, Department of Informatics

February 2020

Table of Contents

Revision History

Name	Date	Version
Makagiga	*11/04/2006*	0.9.3 Beta
Makagiga	*04/09/2006*	0.9.4 Beta
Makagiga	*17/10/2006*	0.9.5 Beta
Makagiga	*18/11/2006*	0.9.6 Beta
Makagiga	*21/12/2006*	1.0
Makagiga	*07/02/2007*	1.1 Beta
Makagiga	*14/03/2007*	1.2
Makagiga	*30/07/2007*	1.9 Beta (2.0 Beta 1)
Makagiga	*22/09/2007*	1.9.1 Beta
Makagiga	*31/10/2007*	1.9.2 Beta
Makagiga	*11/11/2007*	1.9.3 Beta
Makagiga	*22/11/2007*	1.9.4 Beta
Makagiga	*28/12/2007*	2.0
Makagiga	*13/02/2008*	2.2
Makagiga	*27/03/2008*	2.3.1 Beta
Makagiga	*22/04/2008*	2.4
Makagiga	*15/06/2008*	2.9 Beta
Makagiga	*13/07/2008*	2.9.1 (3.0 Beta 2)
Makagiga	*01/09/2008*	2.9.2 Beta
Makagiga	*02/10/2008*	2.9.3 Beta
Makagiga	*10/11/2008*	2.9.4 Beta
Makagiga	*26/12/2008*	2.9.5 Beta
Makagiga	*31/12/2008*	3.0
Makagiga	*23/01/2009*	3.1 Beta
Makagiga	*04/02/2009*	3.2
Makagiga	*15/04/2009*	3.3 Beta
Makagiga	*26/04/2009*	3.4
Makagiga	*06/06/2009*	3.6

Makagiga	09/07/2009	3.7 Beta
Makagiga	24/07/2009	3.7.1 Beta
Makagiga	28/10/2009	3.7.3 Beta
Makagiga	16/11/2009	3.8
Makagiga	29/12/2009	3.8.1
Makagiga	24/02/2010	3.8.2
Makagiga	31/03/2010	3.8.3
Makagiga	29/05/2010	3.8.5
Makagiga	10/08/2010	3.8.6
Makagiga	29/09/2010	3.8.7
Makagiga	08/11/2010	3.8.8
Makagiga	22/11/2010	3.8.9
Makagiga	27/12/2010	3.8.10
Makagiga	23/01/2011	3.8.11
Makagiga	27/02/2011	3.8.12
Makagiga	12/06/2011	3.8.13
Makagiga	09/08/2011	4.0 Beta 1 (3.99.0)
Makagiga	01/09/2011	4.0 Beta 2
Makagiga	09/09/2011	4.0
Makagiga	06/12/2011	4.1 Beta
Makagiga	05/03/2012	4.2
Makagiga	06/08/2012	4.3.1 Beta
Makagiga	20/09/2012	4.4
Makagiga	05/08/2013	4.9 Beta
Makagiga	04/09/2013	4.9.1 Beta
Makagiga	24/09/2013	4.10
Makagiga	20/11/2013	4.11 Beta
Makagiga	07/12/2013	4.12
Makagiga	31/03/2014	4.99 Beta
Makagiga	26/05/2014	4.12.1
Makagiga	16/06/2014	4.99.1 Beta
Makagiga	03/08/2014	5.0
Makagiga	01/09/2014	4.12.2
Makagiga	23/10/2014	5.1 Beta
Makagiga	22/12/2014	5.2

Makagiga	*02/03/2015*	5.3 Beta
Makagiga	*01/04/2015*	5.4
Makagiga	*23/07/2015*	5.5 Beta
Makagiga	*10/09/2015*	5.5.1 Beta
Makagiga	*04/10/2015*	5.6
Makagiga	*10/01/2016*	5.7 Beta
Makagiga	*03/02/2016*	5.8
Makagiga	*16/09/2016*	5.8.1
Makagiga	*26/04/2017*	5.8.2
Makagiga	*14/01/2018*	5.8.3
Makagiga	*22/03/2018*	5.99 Beta
Makagiga	*05/04/2018*	6.0
Makagiga	*06/06/2018*	6.1 Beta
Makagiga	*18/07/2018*	5.8.4
Makagiga	*18/07/2018*	6.2
Makagiga	*03/10/2018*	6.3 Beta
Makagiga	*20/11/2018*	6.4
Makagiga	*28/01/2019*	6.5-beta
Makagiga	*25/02/2019*	6.6
Makagiga	*27/03/2019*	6.7-beta
Makagiga	*06/08/2019*	6.7.1-beta
Makagiga	*02/09/2019*	6.8
Makagiga	*30/11/2019*	6.9-beta

1.   Introduction

1.1. Purpose

The purpose of this document is to present a detailed description of the open-source software Makagiga. It is going to present its features, its interfaces, its capabilities and the constraints under which it should operate. The document is intended to introduce the software to its regular users and also assist developers who are willing to contribute to the further development of Makagiga.

1.2. Document Conventions

This document was created based on the IEEE template for System Requirement Specification documents and follows its conventions.

1.3. Intended Audience and Reading Suggestions

Intended Audience:

- Typical Users who wish to use Makagiga for daily organization and RSS reading. (Making a daily plan, keeping notes).

- Advanced/Professional Users who want to use Makagiga in order to unify everyday tasks in one single software.

- Developers who are interested in contributing to the project by further developing its features or fixing already existing bugs.

For the reader's convenience it is suggested to start by reading the **Introduction** section to familiarize with the general purpose of Makagiga. After that, using the **System Features** section the reader can get a hang of the software's features while using it in real time. If the reader is willing to contribute to the project, it is suggested to thoroughly review the **Overall Description** and **External Interface Requirements** sections.

1.4. Product Scope

Makagiga is meant to be used as an assistance tool for the user's everyday life. The software provides a plethora of options for the user to choose from. Some of the features of Makagiga are a built in calendar widget for scheduling meetings, a todo list widget, text editor, a note-taking widget as well as a RSS widget that allows the user to keep track of his everyday feeds around the internet inside a single software by just subscribing to them. Finally, all these features come in a portable software that is available in both Linux and Windows.

1.5. References

Makagiga's website:

https://makagiga.sourceforge.io/

Makagiga's GitHub page:

https://github.com/kdt/makagiga

IEEE Template for System Requirements Specification Documents:

https://goo.gl/nsUFwy

2.   Overall Description

2.1. Product Perspective

Makagiga was created for everyone who regularly uses text editors, calendar applications, RSS feeds or is in need of some todo listing and is interested in capitalizing on a software that unifies all the above.

It is an open-source, portable application that was developed to run on both Windows and Linux. It has a very active developer basis and uses feedback from its users to keep growing and adapting to the user base recommendations.

2.2. Product Functions

File:

- Save: Saves the current session.
- Restore Sessions: Restores the last user session.
- Manage Sessions: Manage the active user sessions.
- Recently Closed: Allows the user to re open the recently closed files.
- Import: Imports a variety of known file formats into the active tab.
- Export: Exports the content of the active tab.
- Work Offline: Limits Internet Connections in the application.
- Quit: Closes the application.

Edit:

- Add Widget:
    1. Calendar: Adds a calendar widget to the active tab.
    2. Internet Search: Adds an internet search widget to the active tab.
    3. To-do: Adds a to-do list widget to the active tab.
    4. Notes: Adds a simple notepad widget to the active tab.
    5. RSS Feed: Adds an RSS feed reader widget to the active tab.
- Paste Text as Sticky Notes: Adds a new notepad widget to the active tab and pastes the clipboard's content inside it.

- Get More Widgets: Allows the user to download extra widgets.

View:

- Tool Bars: Used to show/hide application's tool bars.
- Notifications:
- Show Widgets:
- Full Screen: Makes the application run in full screen.
- Lock: Locks the contents of the active tab.
- Settings: Opens the active tab settings.

Tools:

- Console: Opens the console feed as a new active tab.
- Calendar: Opens a calendar as a new active tab.
- Backup: Creates a backup.
- Delete Private Data: Deletes history, cache, logs.
- Application Shortcut: Creates an application shortcut in the desired destination.
- Plugins: Add or remove plugins.
- Settings: Opens application settings.

Go:

Allows the user to navigate throw the open application files.

Help:

- Help: A variety of useful links will appear by clicking this option.
- Check for Updates: Checks for software updates.
- Topics: A quick Q&A for the user to get information from.
- What's This?: Opens an interface for the user to determine the purpose of every element in the screen.
- About: Opens information about the software as a new active tab.

New:

- New Folder: Creates a new folder.
- Quick Paste: Gives a variety of formats for the user to select from to paste the text in.

- To-do: Pops a dialogue for the user to create a new to-do list.

- Notepad: Pops a dialogue for the user to create a new text file document.

- Text/HTML: Pops a dialogue for the user to create a new HTML file.

- Image: Pops a dialogue for the user to create a new Image document.

- Link: Pops a dialogue for the user to save a new link and add a description to it.

- Add RSS Feed: Pops a dialogue for the user to add an RSS feed in the active tab.

- Get more plugins: Allows the user to download extra plugins.

Download and update all RSS feeds: Downloads and updates every RSS feed that the user has added to the application.

Search:

- Name: Makes a search based on letter.

- Date Modified: Makes a search based on the modification date.

- Tags: Makes a search based on tag.

- Icon: Makes a search based on the icon.

- File Type: Makes a search based on file type.

- Color: Makes a search based on color.

- Rating: Makes a search based on rating.

2.3. User Classes and Characteristics

- Typical Users who wish to use Makagiga for daily organization and RSS reading. (Making a daily plan, keeping notes).

- Advanced/Professional Users who want to use Makagiga in order to unify everyday tasks in one single software.

- Developers who are interested in contributing to the project by further developing its features or fixing already existing bugs.

2.4. Operating Environment

- Windows 7

- Windows 8

- Windows 10

- Linux distributions

2.5.   Design and Implementation Constraints

Makagiga is developed in Java. It is created on the premise that there is a single application running at every time and any other feature, widget or plugin runs in its own fully modular code. This way using APIs the plugin development is fully supported and made easy.

2.6.   User Documentation

Makagiga's Wikipedia page:

https://en.wikipedia.org/wiki/Makagiga

Makagiga's Reviews:

https://sourceforge.net/p/makagiga/wiki/Reviews/

Makagiga's FAQ:

https://sourceforge.net/p/makagiga/wiki/FAQ/

Makagiga's Read Me:

https://sourceforge.net/p/makagiga/code/HEAD/tree/trunk/makagiga/README.md

2.7.   Assumptions and Dependencies

Makagiga is developed in Java and that is why it requires Java to be installed on the user's computer. The recommended version of Java is 13. In both Linux and Windows this Java version is included in the Makagiga packages.

3. External Interface Requirements

3.1. User Interfaces

1. Makagiga's Main Screen:
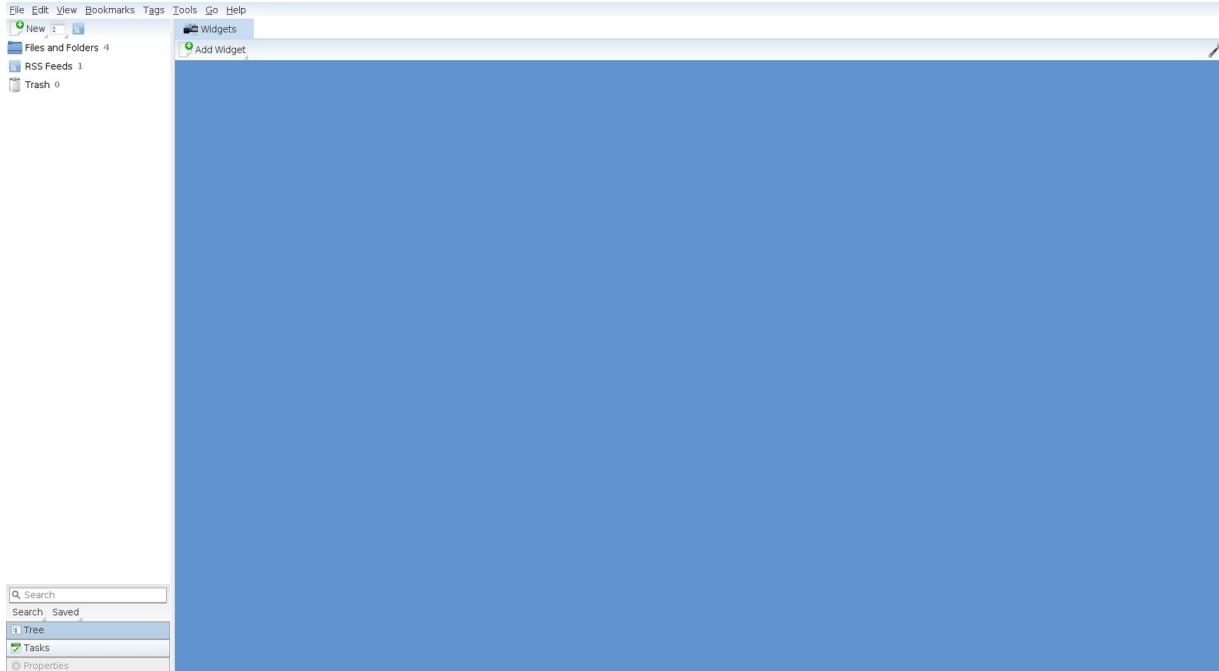
2. Import HTML Files (supported file formats displayed):
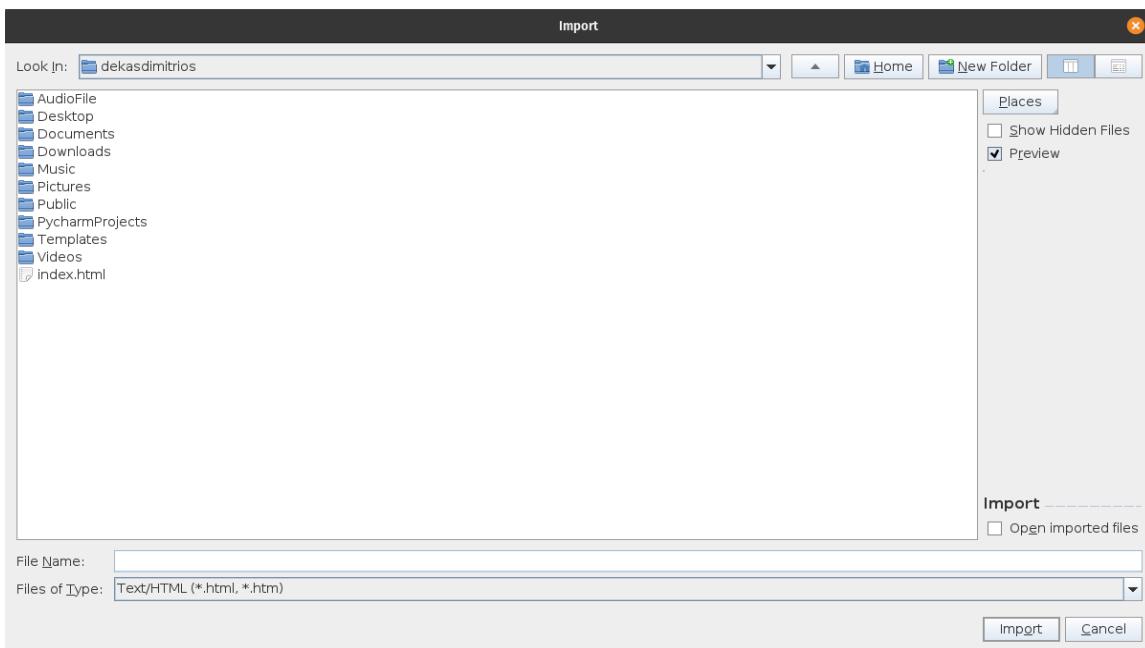
3. Import Image Files (supported file formats displayed):
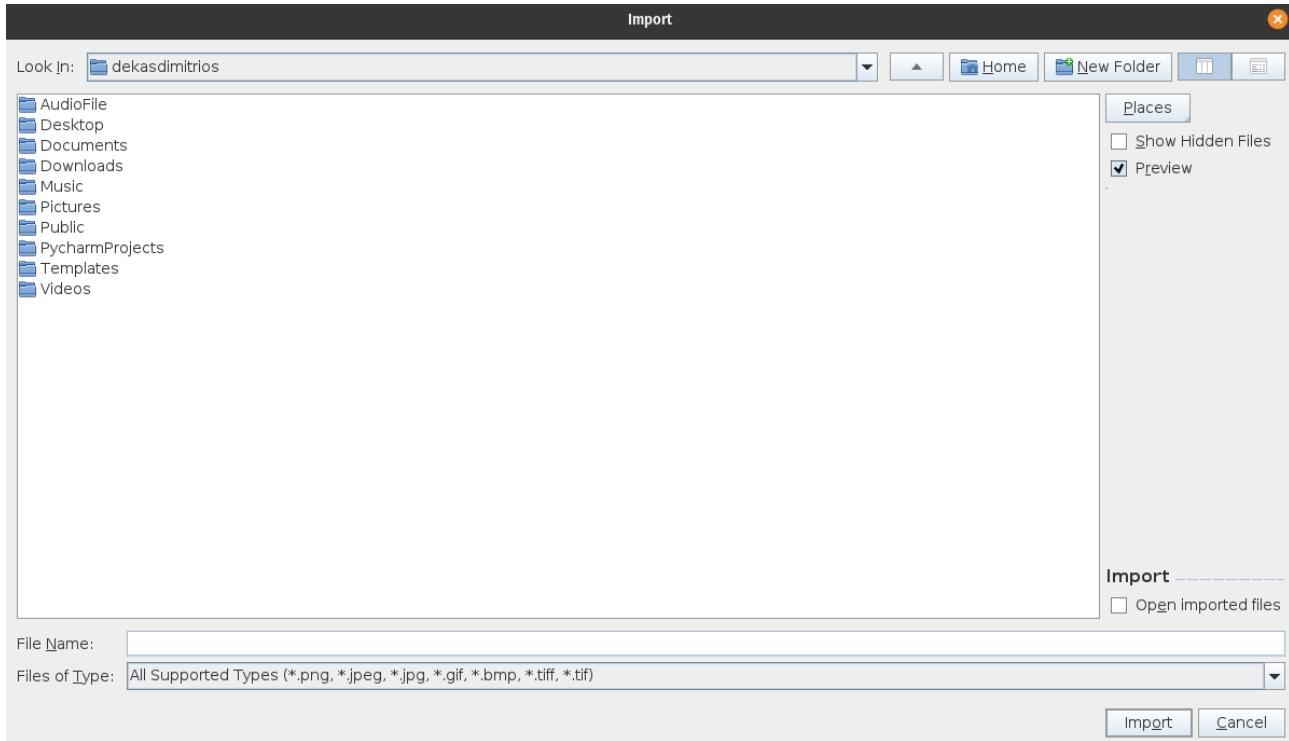
4. Import Link Files (supported file formats displayed):
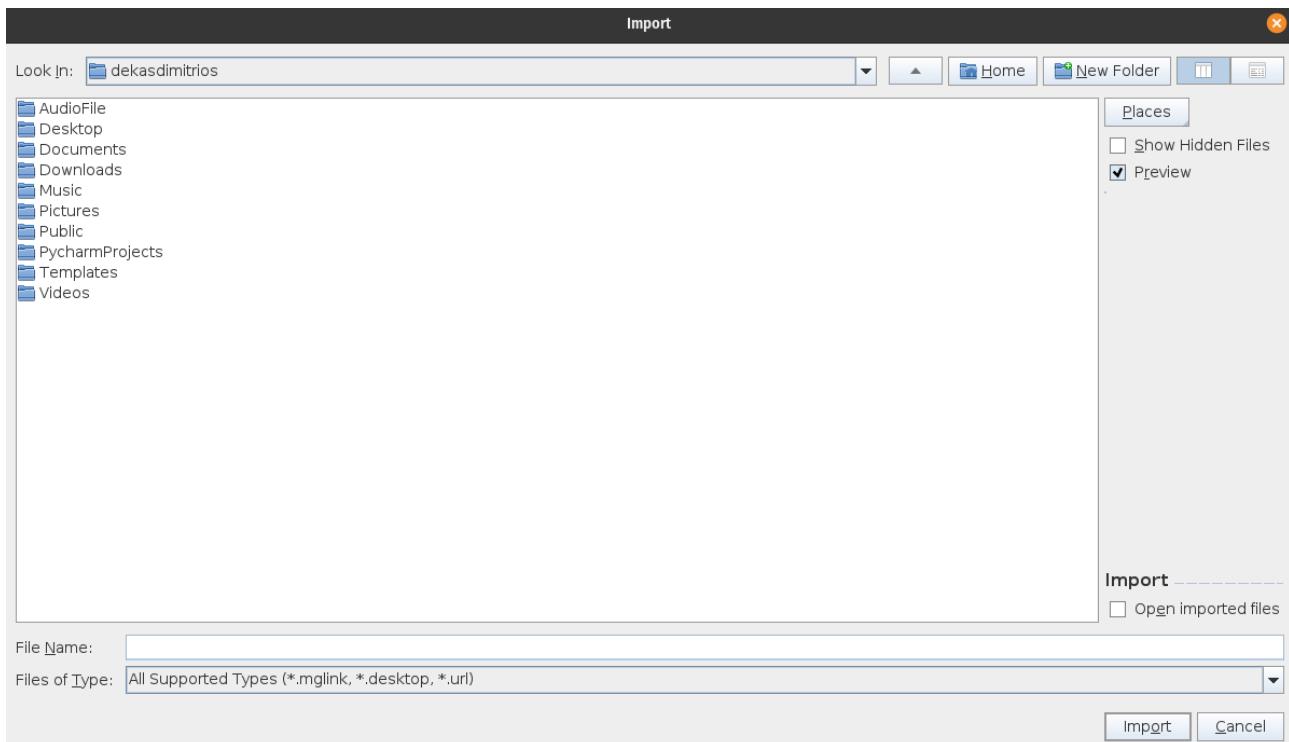
5. Import To-Do List Files (supported file formats displayed):
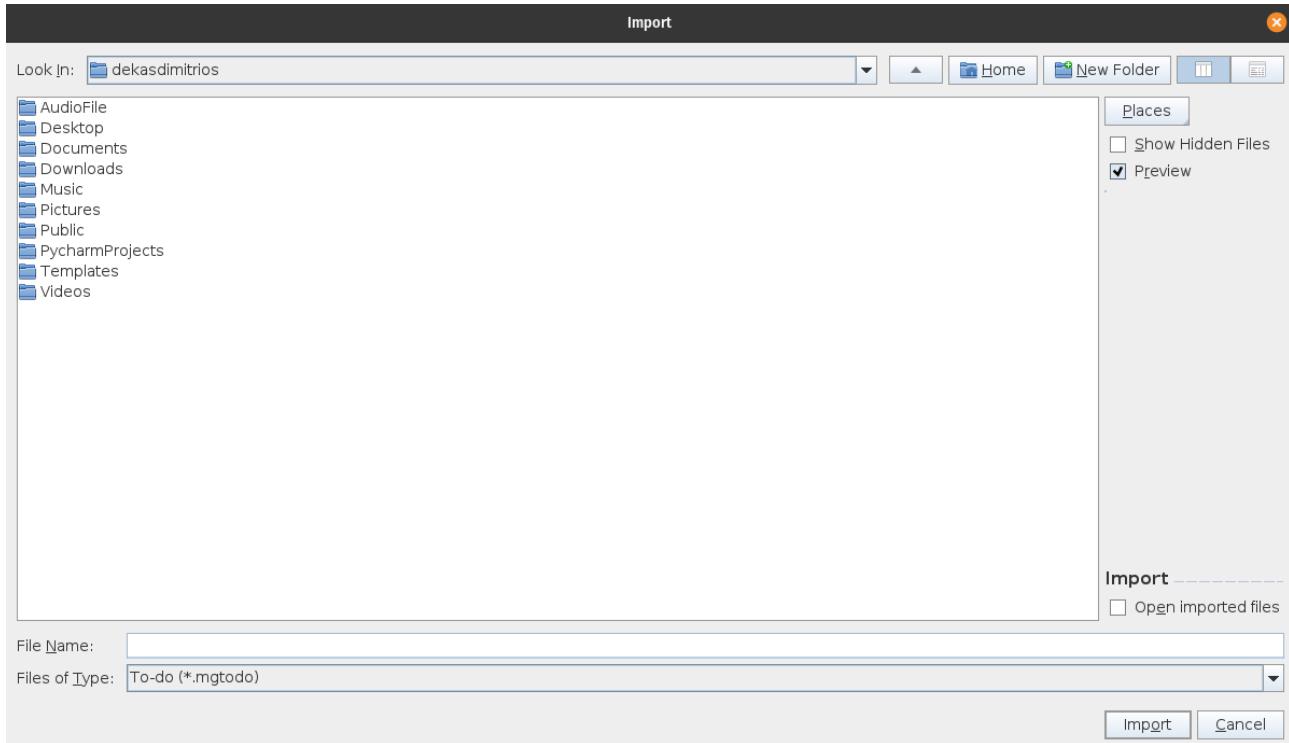
6. Import Notepad Files (supported file formats displayed):
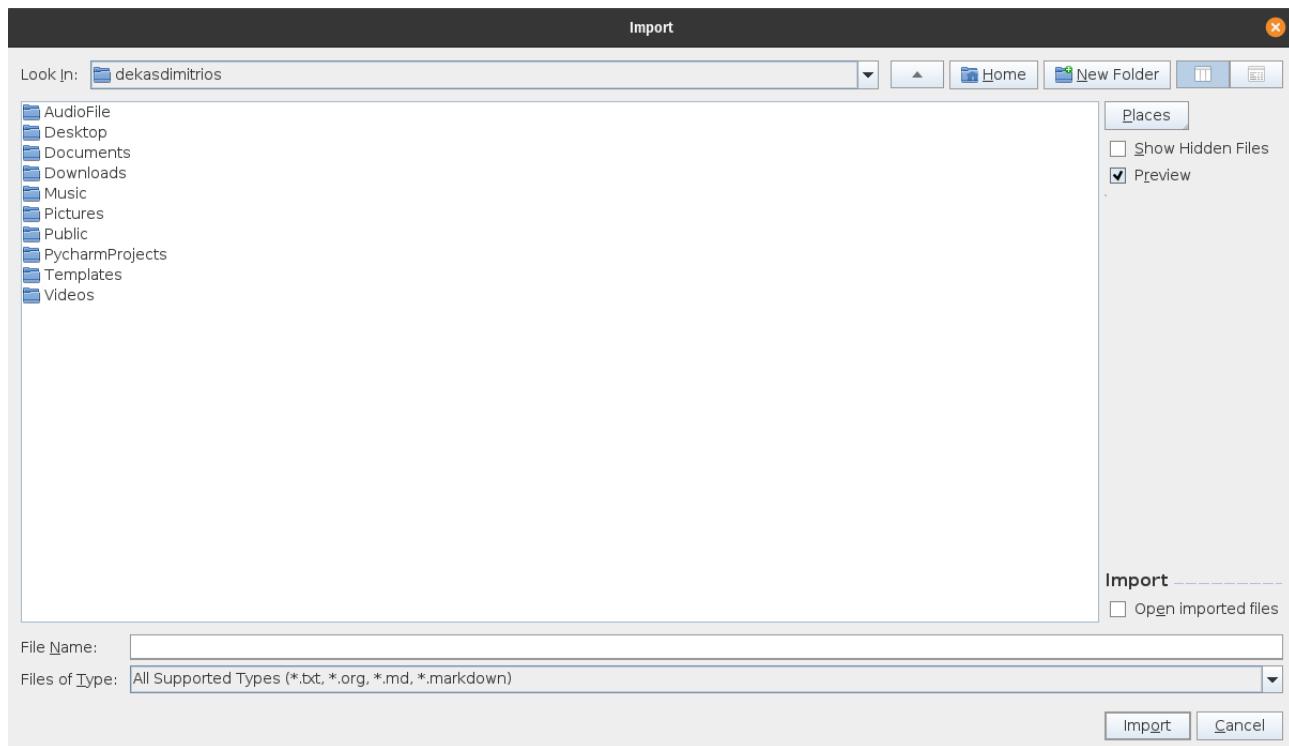
7. New File:
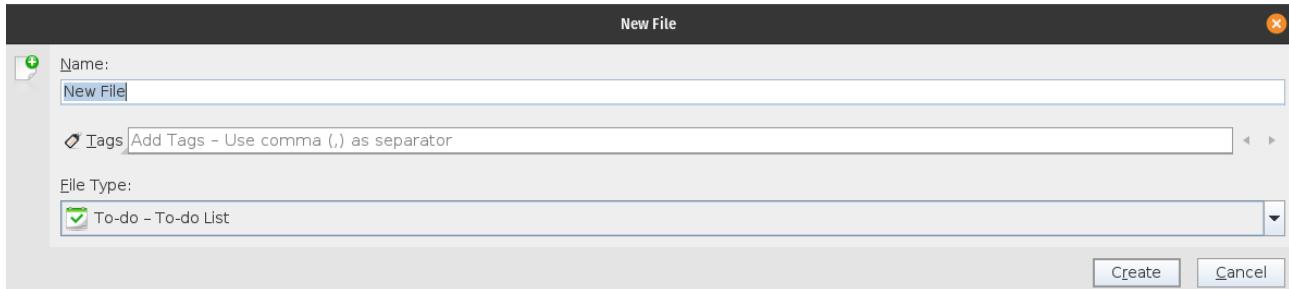
8. *Overview of Calendar. Buttons presented from the top to bottom. Close button (closes the calendar widget), left/right arrows (moves the calendar to the previous/next month accordingly), plus/minus sign (moves the calendar to the previous/next day accordingly), three dots button (displays a menu that allows the user to make selections about the day, month and year of the calendar). The calendar dates are clickable and after a click they display a menu that gives a variety of options to the user about this exact date (e.g. new task creation for that date, tag this date as important etc.)*
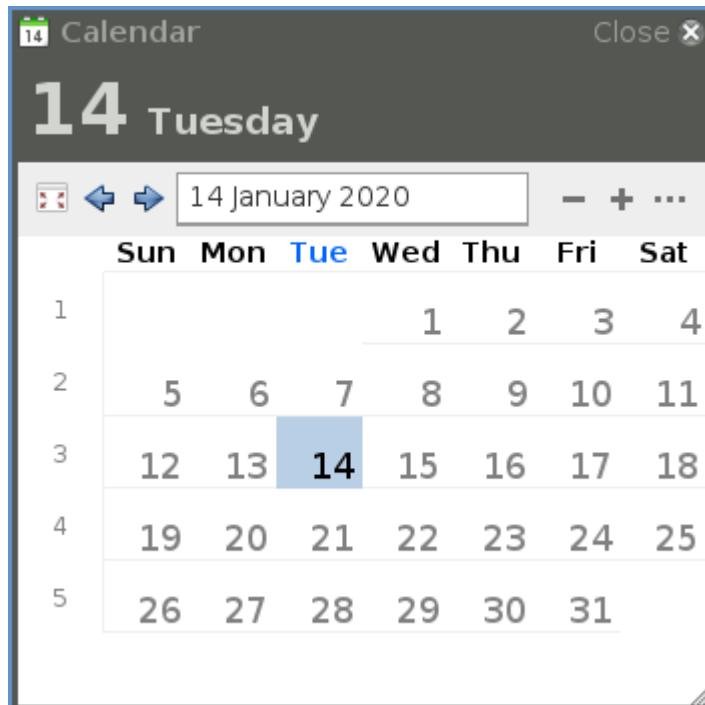
9. Overview of Internet Search. Buttons presented from right to left. Search engine logo (allows the user to change the preferred search engine), three dots button (displays a menu that allows the user to make selections about search method and search settings), check button (completes the searching, opens the search results in the default user's browser)

10. Overview of Notes. Buttons presented from the top to bottom. Close button (closes the calendar widget), printer button (prepares the printing of the note's contents), settings button (opens a settings window that is used to change various things about the note document like font, name and text color), color button (changes the background color of the widget)
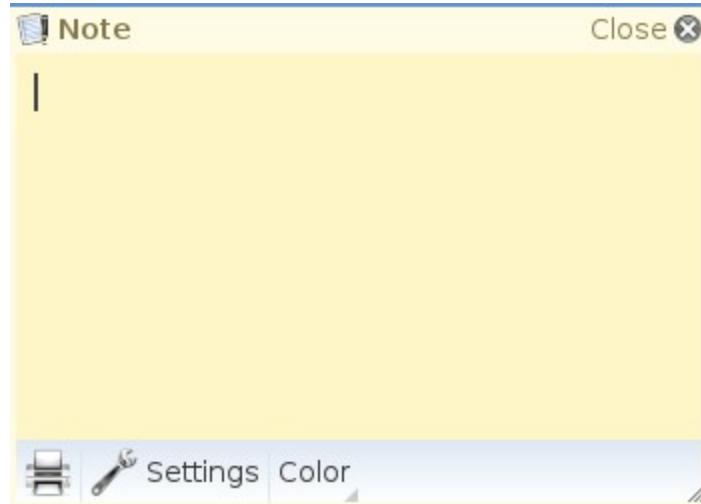
11. Overview of RSS Feed. Variety of clickable links that open the thread in the user's default browser. Buttons presented from right to left. Refresh button (refreshes the RSS feed), settings button (displays a menu that allows the user to make selections about the feed's display and auto refreshing)
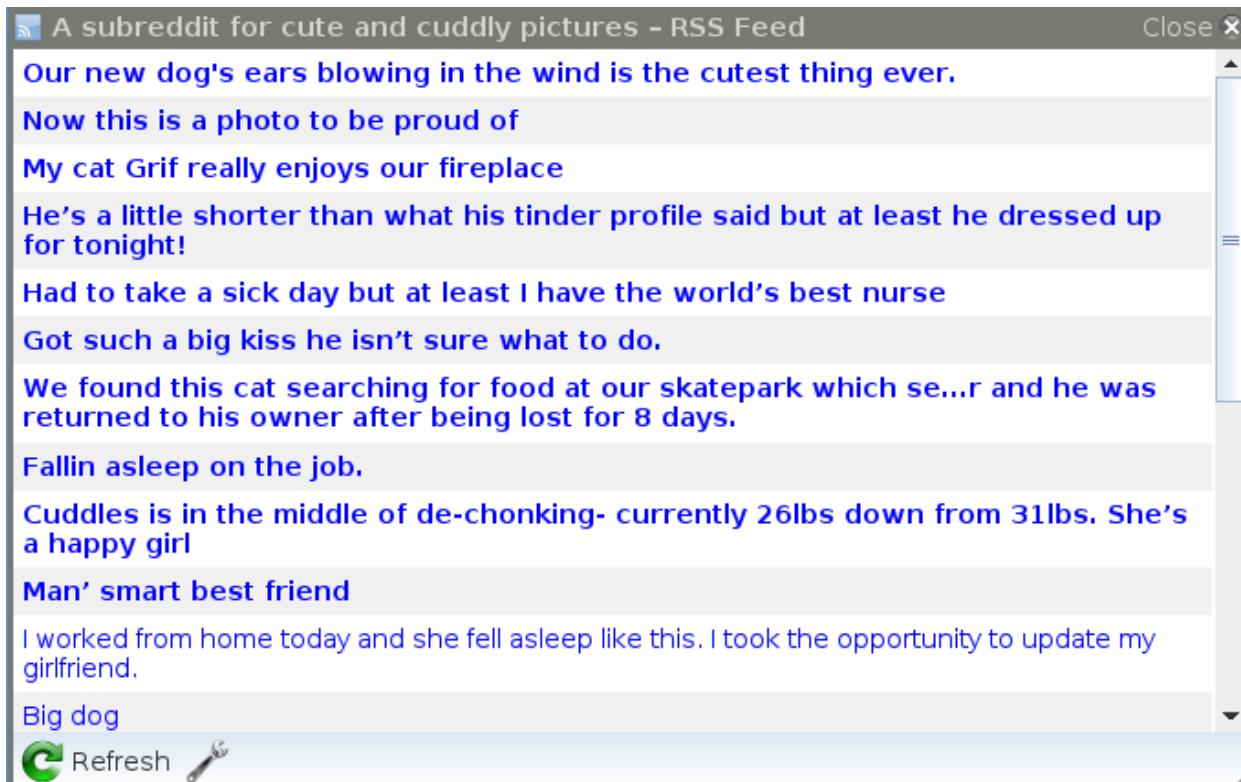
12. Overview of To-Do Lists. Buttons presented from right to left. Printer button (prepares the printing of the todo lists' contents), undo/redo buttons, cut/copy/paste buttons, delete done button (deletes tasks marked as completed), indent/unindent buttons (indent/unindent the task inside the list), page/pencil button (task priorities setter)
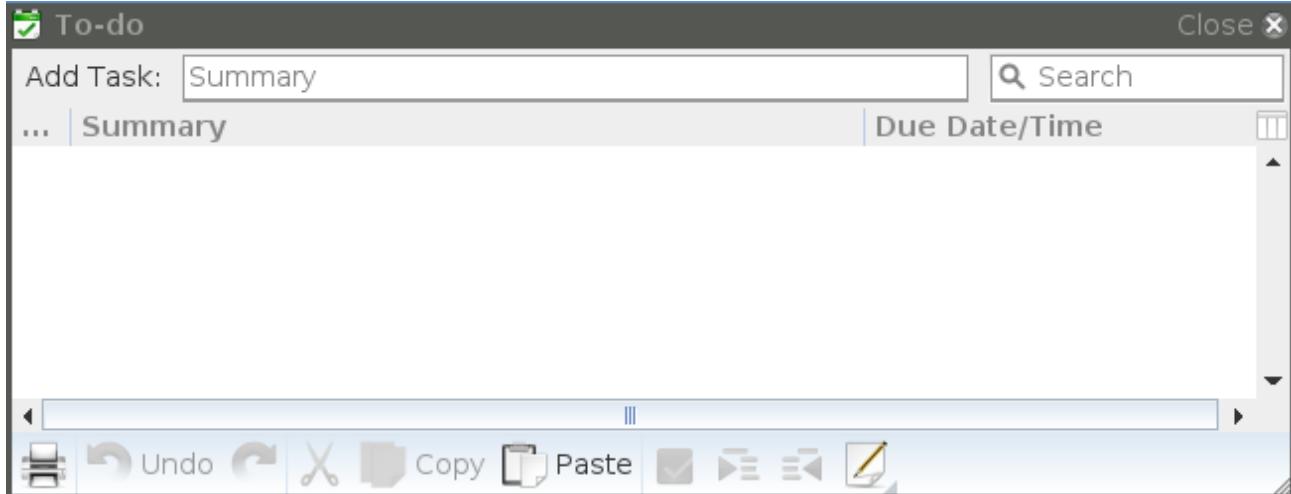
13. Statistics. Displays statistics about user's tasks. In this example it contains information about the due dates of tasks.
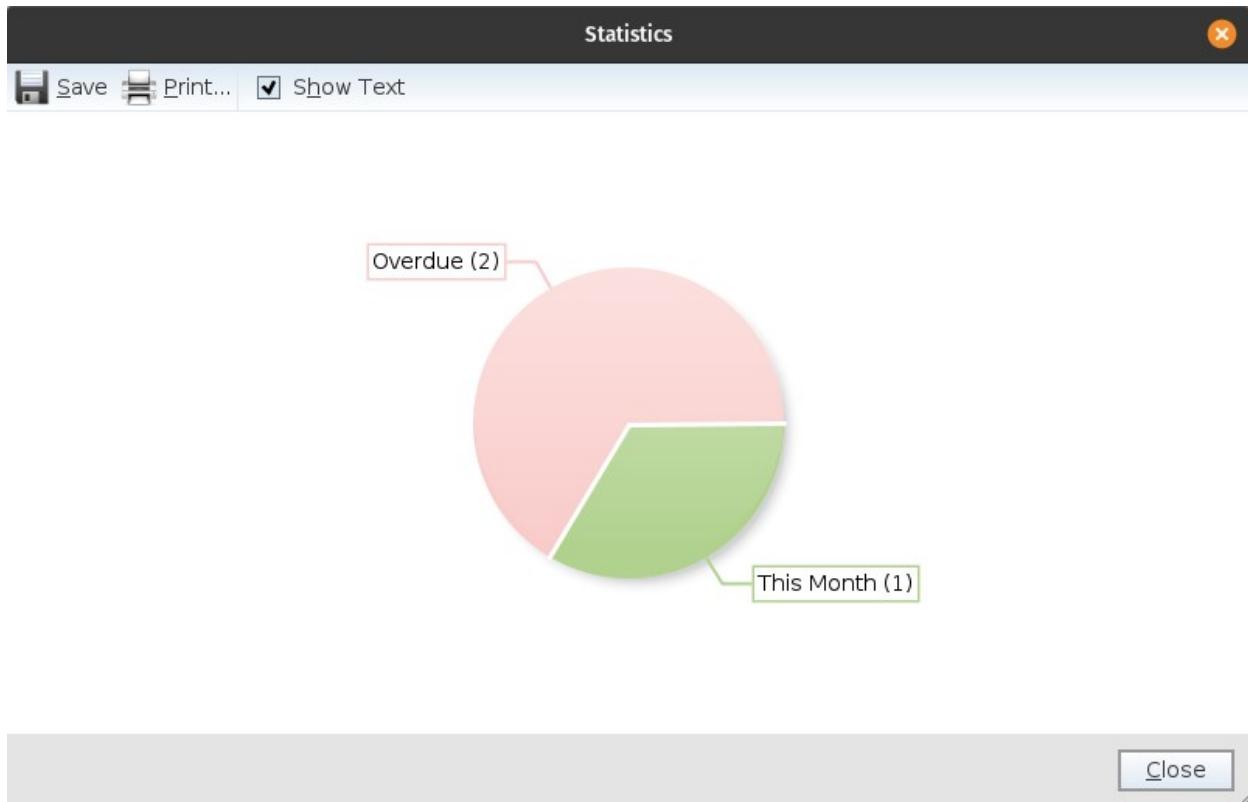
14. Tools -> Backup. Create a backup of the current application state.
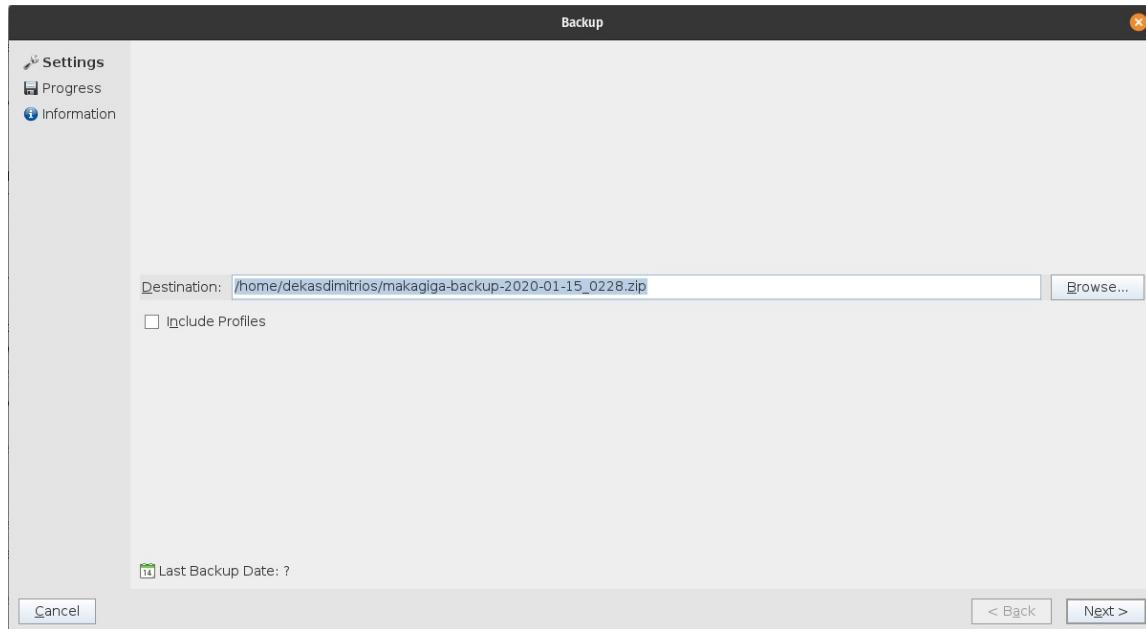
15. Tools -> Plugins. Manage the application's plugins.
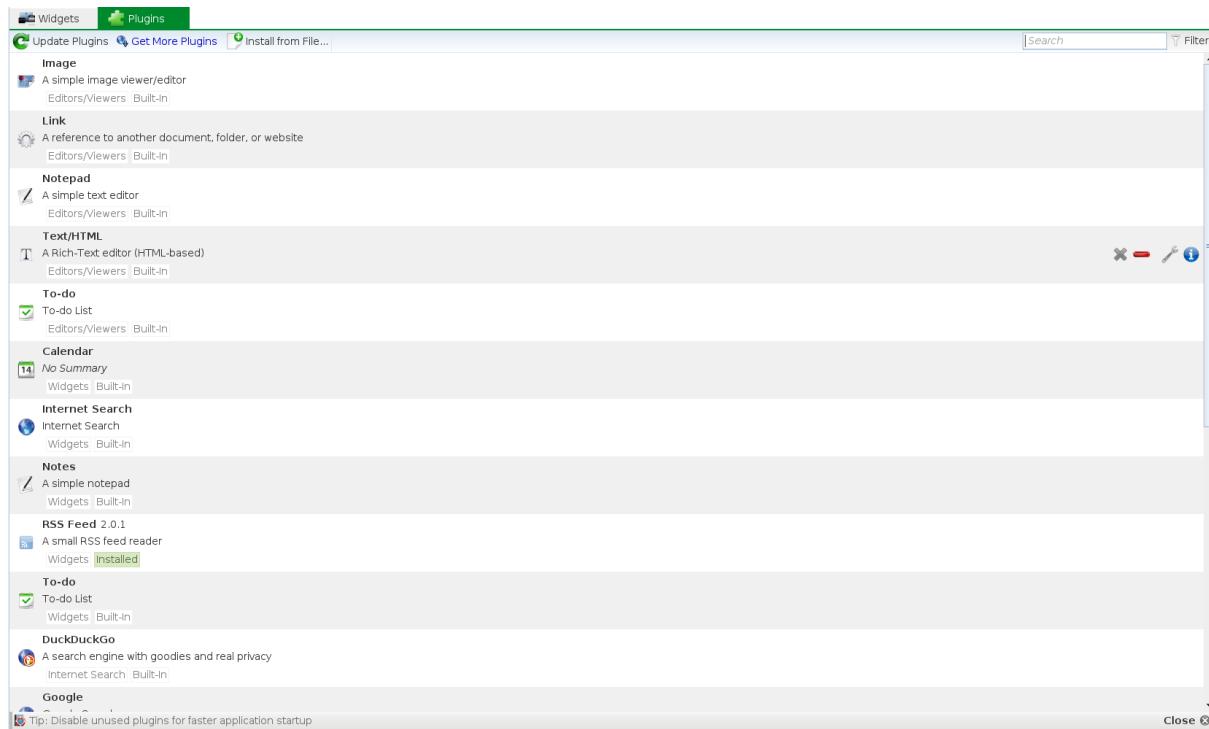
16. Tools -> Settings. Opens the application's settings interface.
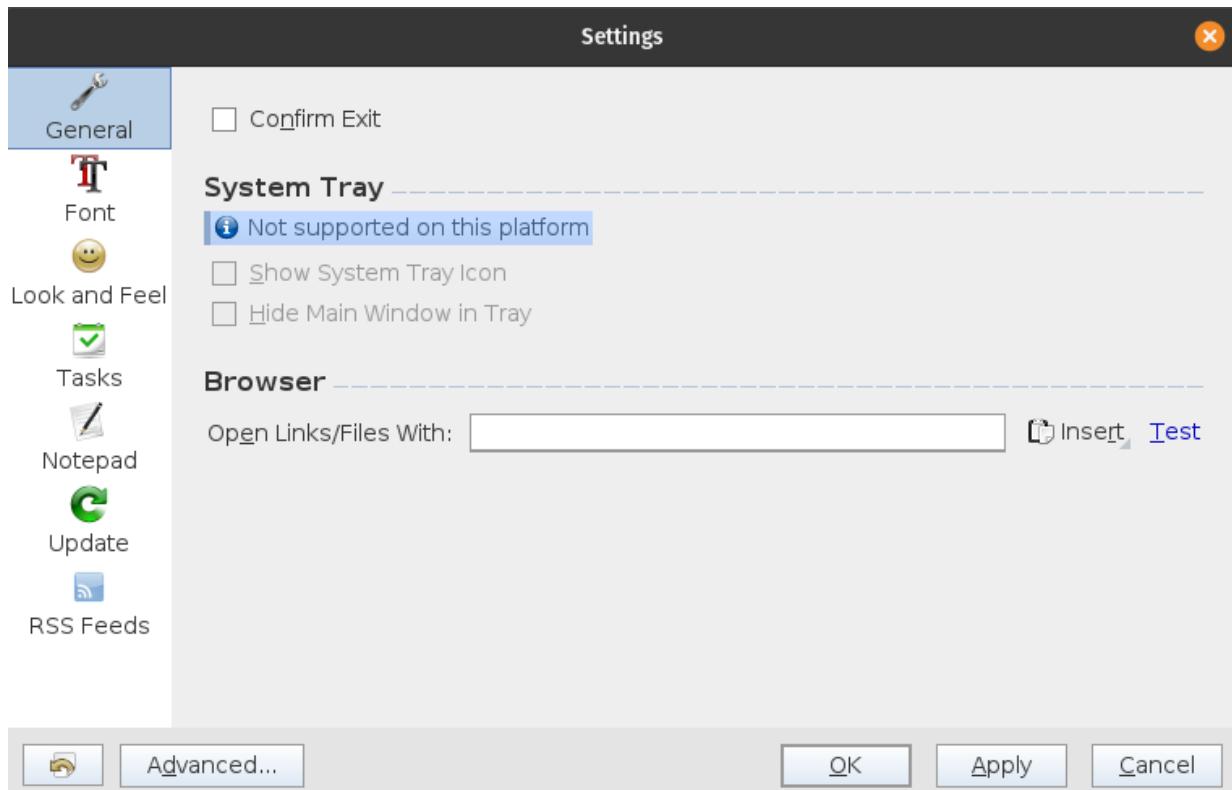
17. Help -> About
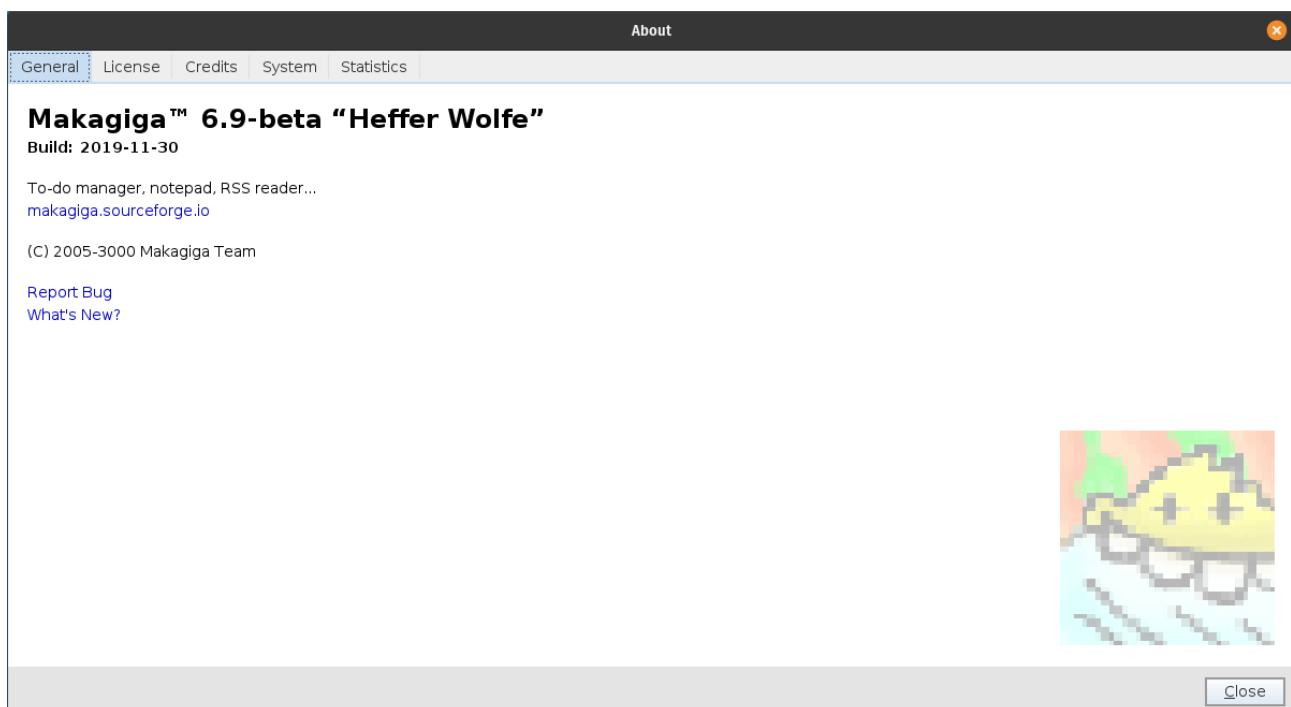
18. Tree-like Interface. A tree-like structure is given as an option to the user in order to better organize the stored notes, tasks, feeds and links. The user is able to create new folders or any new type of document and store it in the appropriate folder by just clicking the **New** button on the top left corner of the interface and selecting the desired option. Any unwanted document is movable and can be disposed inside the **Trash** section or even be deleted.

19. Tasks Interface. In this interface the user is able to print the tasks stored in his application using the top left printer button, see statistics about the stores tasks, search for a task based on letters, or even sort the tasks based on some parameters.

20. Task Manipulation Interface. Through this interface the user is able to fully manipulate the stored tasks in the application.
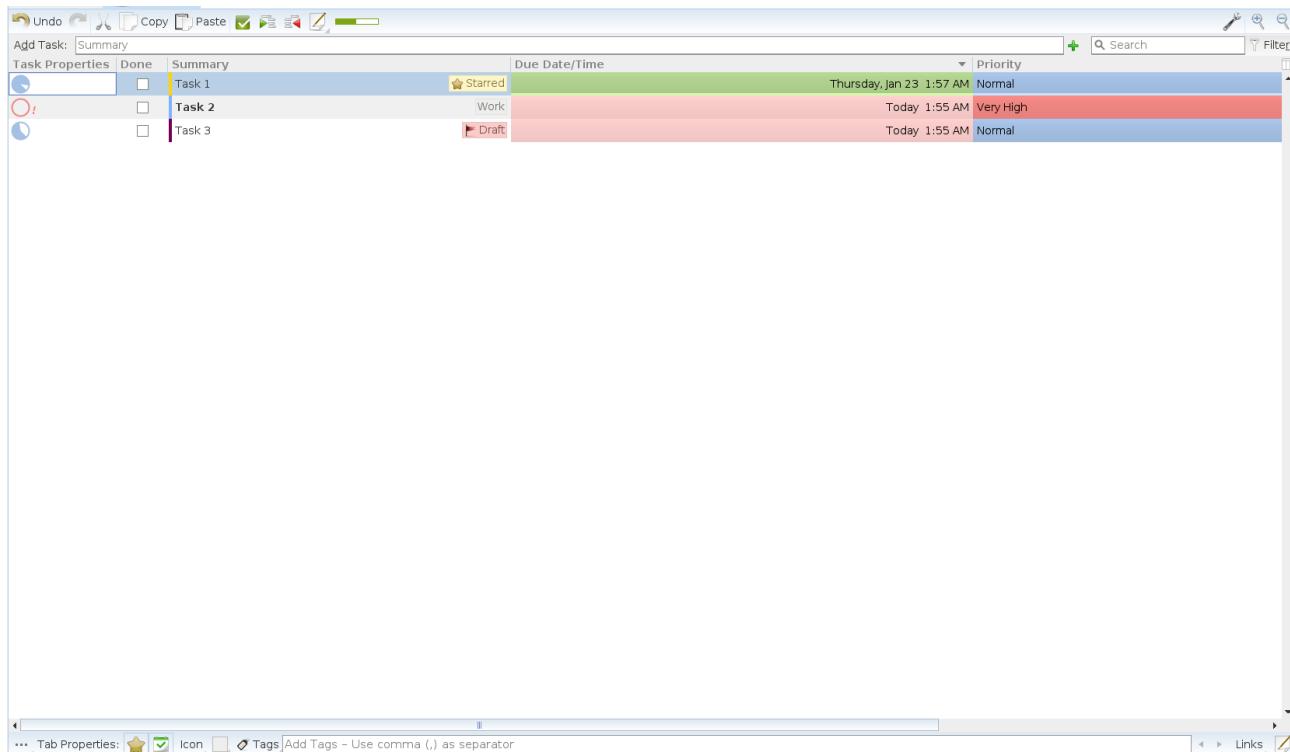
21. RSS Feed Interface. Any RSS Feed added by the user can be accessed through here.
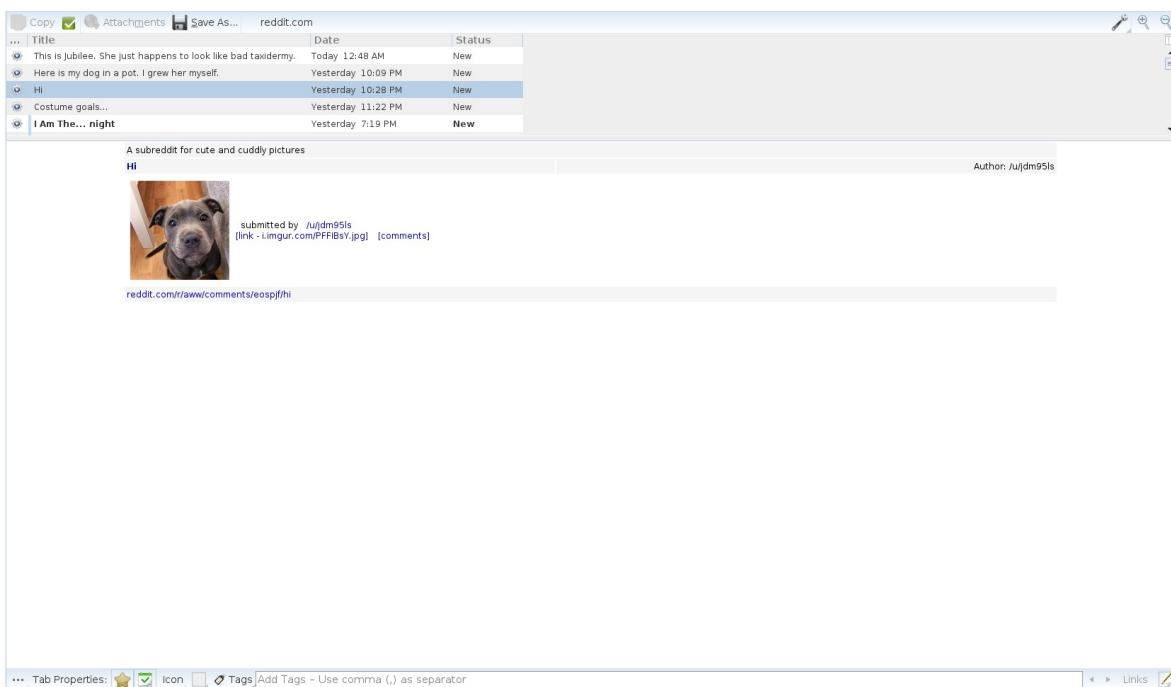
3.2. Hardware Interfaces

Makagiga is a portable application which means is can be run by a USB device. For speed reasons USB 3 is suggested. Since it is a very lightweight software is has no special hardware requirements and can be run in any kind of device with minimum CPU and RAM specifications.
Since the application uses a graphical user interface, a compatible graphics card is required.

3.3. Software Interfaces

Makagiga requires Java to be installed on the user's system in order to function, more specifically Java 13 for its latest release. More information can be found on the **Assumptions and Dependencies** subsection.

Makagiga can be connected with any RSS feed API in order to display the desired feed directly inside the software.

3.4. Communications Interfaces

Makagiga requires an internet connection to install new plugins and widgets or update already installed ones.

4. System Features

This section demonstrates Makagiga's features and how they can be used in order to help the users in the tasks the wish to accomplish with them.

4.1. Add Task

By clicking a date in the Calendar widget a drop out menu appears. Selecting the Add Task option will trigger the appearance of the below user interface windows. In this window users can directly interact with the dates on the calendar by clicking on them and marking them as the active date.

After choosing the date in which the user wishes to add a task, there is the summary box where the user can precisely describe the task and store information about it. The user is able to determine the completion percentage of the task, its priority level or select the **Anniversary** section's button, in order to characterize the task as an anniversary. Besides that, there is an exact hour selection for the task on the top right of the window.
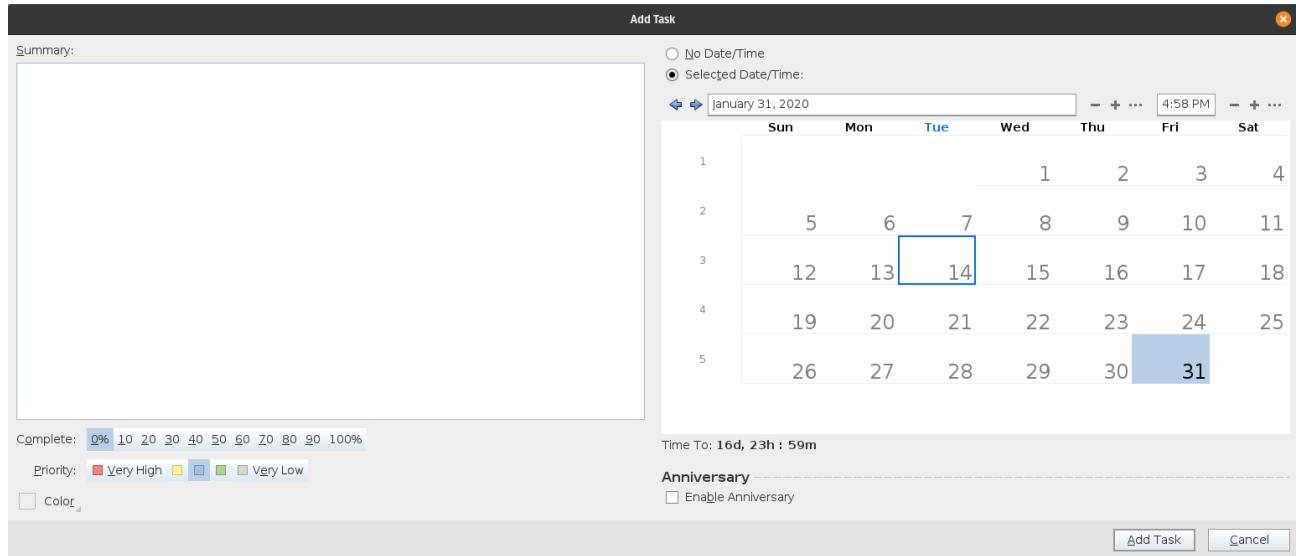
4.2. RSS Feed Insertion

The user is able to insert an RSS Feeds inside the application to keep track of every new post that is made in them. By passing the web address in the dialogue box of the **Settings** window interface, and selecting the desired option, a new section is created in the application in order to host the content of the feed in any given time.
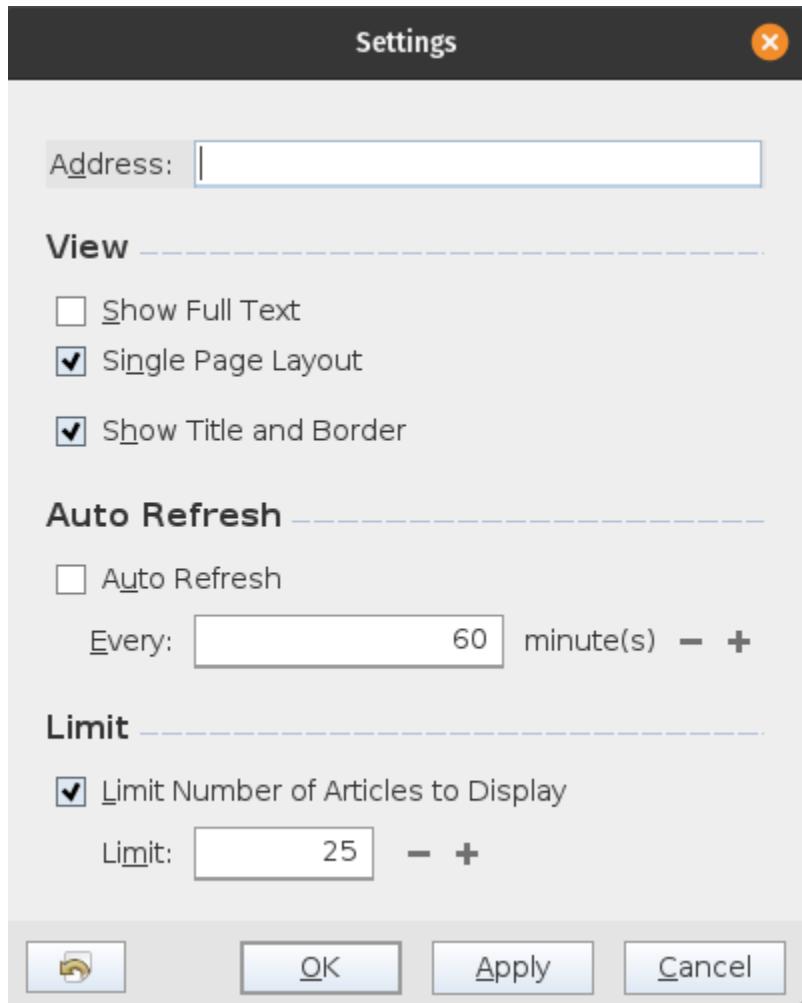
4.3. Note Keeping and Organizing

The user of Makagiga is able to create notes and organize them. Notes can be held inside other notes in order to demonstrate the connection between them. The sticking property of the notes makes the application easy to use and the interface fun and interactive for the user. Being able to drag and drop notes from the sidebar makes storing and using notes an fast and productive way of remembering things by note talking. The notes' color can be changed to make them sharper for the eye.
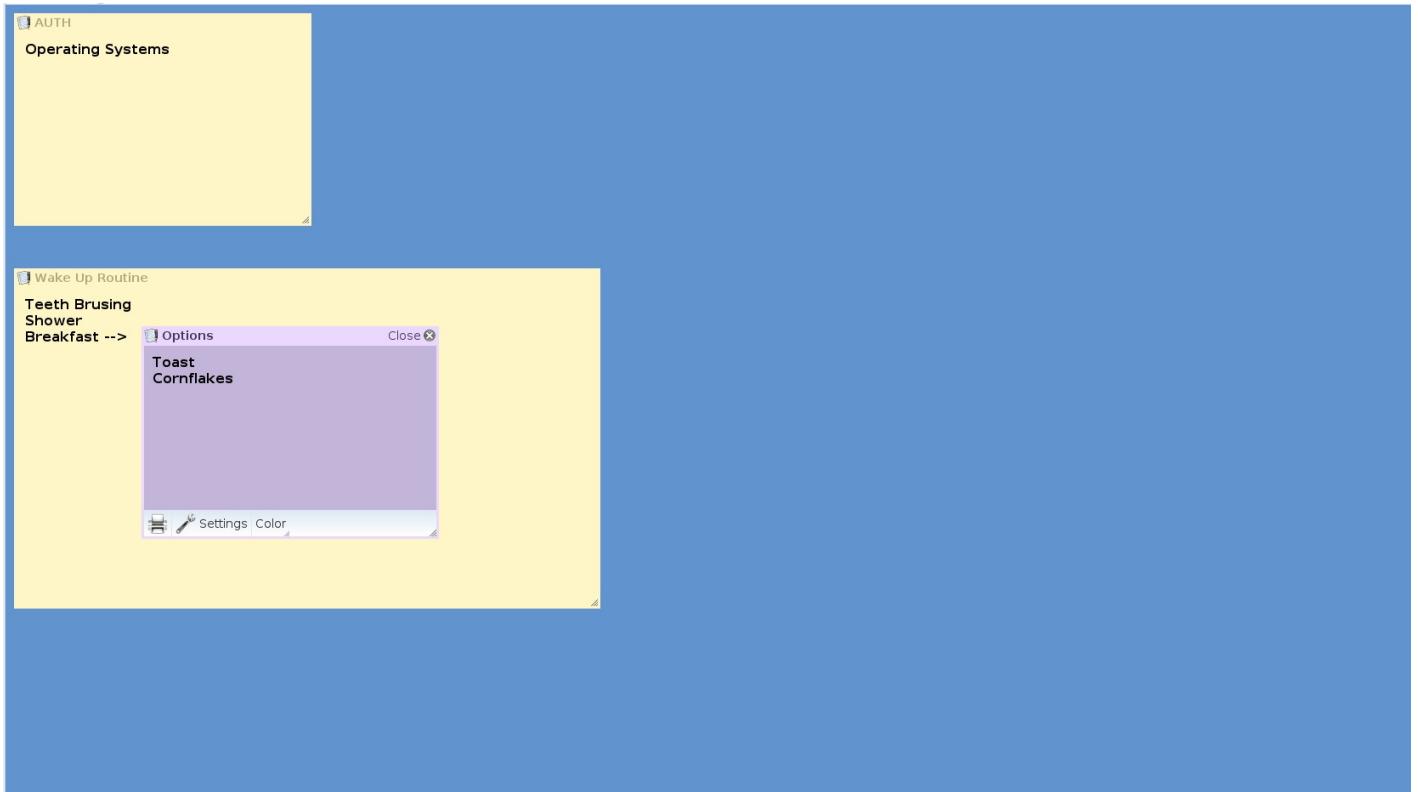
4.4. Calendar Organization

The user is able to organize everyday life through the calendar widget provided by the app. Combining the calendar with the task management system inside the application, the user can program each day by detail.
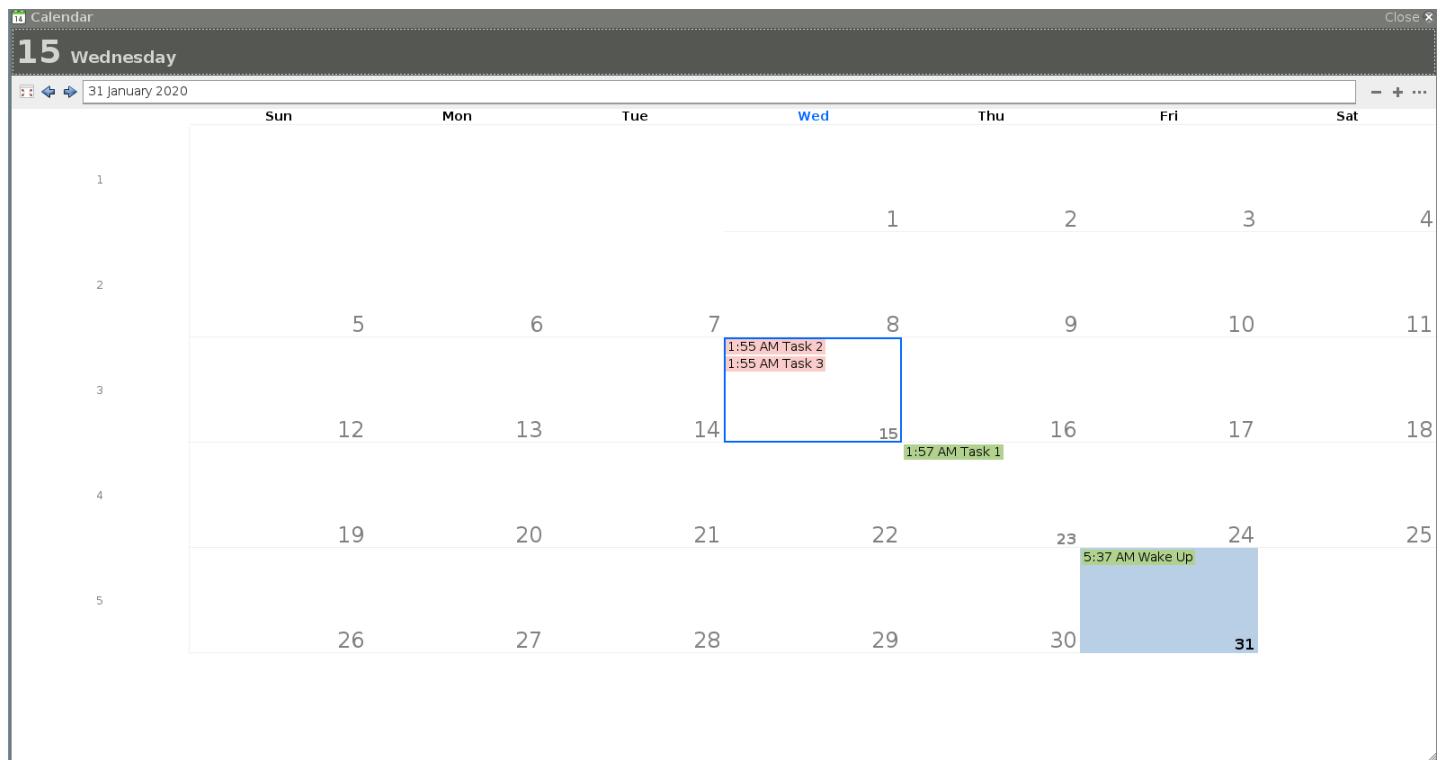
4.5. Storing Links

Storing links is made easy with Makagiga. The user may wish to remember a useful link and a feature is created just for that purpose. The link address is pasted in the dialogue box and a description panel is available for the user to keep notes about the link that is going to be stored.
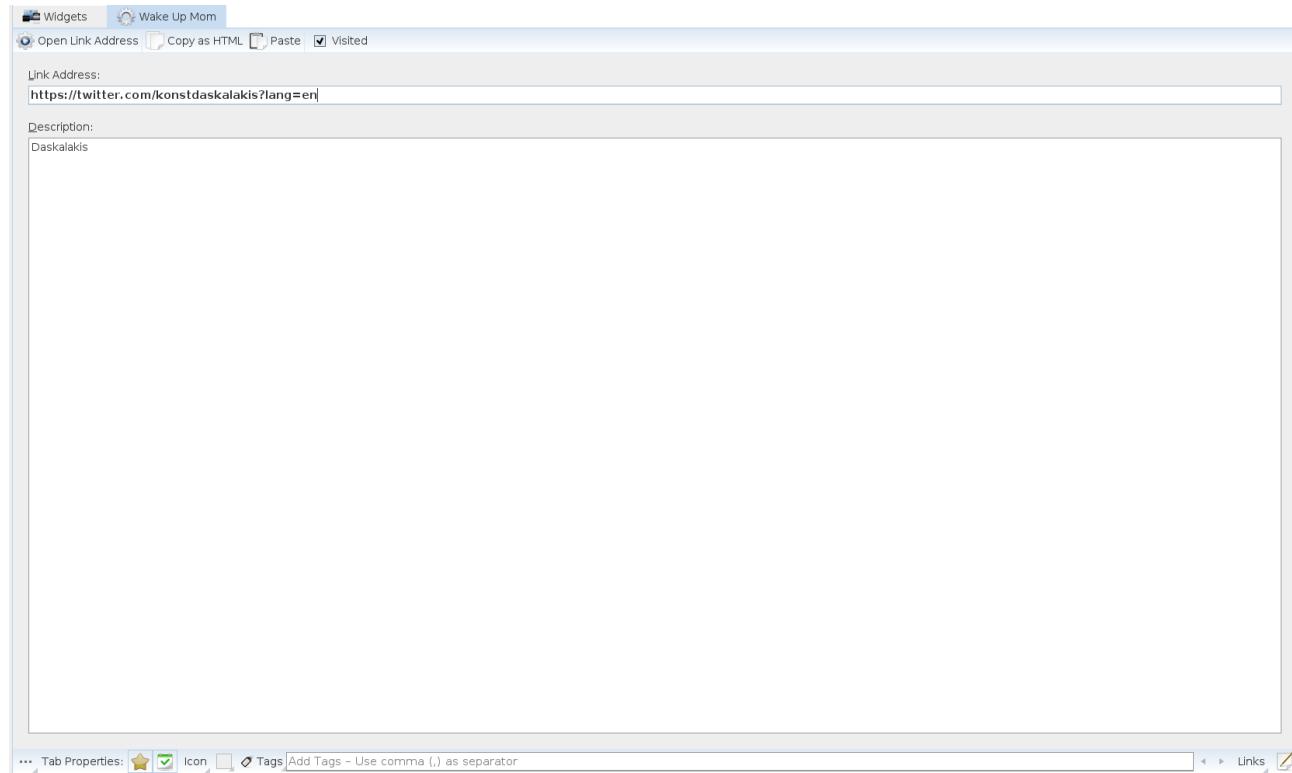
5. Other Nonfunctional Requirements

5.1. Performance Requirements

Makagiga is a portable application and can be run easily in any environment that supports it. It is very light wight and can be run in almost any computer system the user may provide. It is a Java application and so it requires a Java version to be installed in the system. However a Java version in included in Makagiga's packages for the user to freely use.

5.2. Safety Requirements

The Makagiga's development team tries to make sure that no bug or lose of stored data will be a problem for the user while using the application by constantly updating and patching the software. There is a bug tracker available where users can report any bugs they encounter so the development team can take care of them. Make sure to describe the bug's nature, precisely how it occurred to the user, the actions that made it happens and any other useful information the can be provided in the link below:

https://sourceforge.net/p/makagiga/bugs/new/

5.3. Security Requirements

Makagiga has no special security requirements and thus any user could use it freely without any special user privileges.

5.4. Software Quality Attributes

Makagiga is a very easy-to-use application and no special knowledge is required by its users. Having said that, a regular user of an RSS Feed will find the application a bit easier to use and get familiar with. From the developers perspective, a basic knowledge of Java is needed. Besides that, the programmer that wishes to contribute to the project will need basic understanding of the necessary APIs. Therefore, he should be able to get a hang of the project's structure.