

Requirements and Analysis Document for ResuMate (RAD)

Contents

[Contents](#)

[1 Introduction](#)

[1.2 General characteristics of application](#)

[1.3 Scope of application](#)

[1.4 Objectives and success criteria of the project](#)

[1.5 Definitions, acronyms and abbreviations](#)

[2 Requirements](#)

[2.1 Functional requirements](#)

[2.2 Non-functional requirements](#)

[2.2.1 Usability](#)

[2.2.2 Reliability](#)

[2.2.3 Performance](#)

[2.2.4 Supportability](#)

[2.2.5 Implementation](#)

[2.2.6 Packaging and installation](#)

[2.2.7 Legal](#)

[2.3 Application models](#)

[2.3.1 Use case model](#)

[2.3.2 Use cases priority](#)

[2.3.3 Domain model](#)

[2.3.4 User interface](#)

[2.4 References](#)

[APPENDIX](#)

[Use cases](#)

[Use case texts](#)

Version: 1.0

Date: 25-05-13

Author: Danny Lam, Laszlo Sall Vesselenyi, Patricia Paulsson, Sara Nadi

This version overrides all previous versions.

1 Introduction

In today's society, the difficulty to get a profession is continuously increasing due to unemployment. You have to stand out among the other job applicant's resumes in order to make sure that the employer notices you. It is much easier said than done, although there are a lot of tips and tricks on the internet. Nevertheless, the motivation for writing a resume decreases as you do not want to browse around to be apprised the art of well written and readable resume. That is how the office application *ResuMate* comes into the picture.

1.1 Purpose of application

The objective of this application is to provide efficient ways of creating professional resumes and cover letters, including utilities such as easy to use templates and basic photo editing tools. This is due to the fact that people nowadays feel that creating these documents is rather elusive, thus we believe that this application may be the supporting wheels for users to accomplish their goals.

1.2 General characteristics of application

The application will be a desktop, standalone (non-networked), single user application with a graphical user interface for the Windows/Mac/Linux platforms. While being non-networked, functionality may be extended to include rudimentary email capacity for sending resume directly to future employer.

The application will provide some of the simpler tasks of text editing. The documents are not "sand boxes", but are always preconfigured by (changeable) templates. The user can insert an image, which is then automatically scaled and placed into the predetermined area in the document. Personal Information may be stored to make future creation of documents easier. The functions of saving, opening and exporting a document will also apply.

1.3 Scope of application

Writing resumes and cover letters. The application will not allow for free writing (writing without using templates). The application will make use of one or more internal file types for saving data (personal information, application state and the documents), although printing (and potentially email communication) is handled by means of PDF. See Possible future directions.

1.4 Objectives and success criteria of the project

1. It should be possible to write a CV/resume in a facilitated way with various tools.
2. It should be possible to write cover letters in a facilitated way with various tools.
3. It should be possible to ad and grayscale an image.
4. It should be possible to print or export a resume and/or a cover letter.

1.5 Definitions, acronyms and abbreviations

- Doc and/or doc. = Refers to a document (docs in plural) in view (selfexplanatory, but important)
- CV = curriculum vitae/resume
- Java = platform independent programming language.
- JRE = Java Runtime Environment
- Swing = Provides a toolkit for good looking graphical user interfaces in Java

2 Requirements

2.1 Functional requirements

The user should be able to:

1. Create, save, open & export a document
2. Write, edit and format a text.
3. Import & grayscale an image.

2.2 Non-functional requirements

2.2.1 Usability

This application's graphical user interface will follow the principle of previous work meaning that the application will inherit some graphical representation models from other document handling applications. This is since it will in some aspects be familiar to the new user and thus not feel discouraging.

2.2.2 Reliability

This application should not crash in listed use cases.

2.2.3 Performance

This application should take arbitrary amount of time to perform a certain task. Lightweight actions, such as typing, should require “little to no time” to process, opening documents and importing images may take a second or two.

2.2.4 Supportability

This application is based on being used as a desktop application in the JRE with Swing, thus is compatible for all operative systems compatible with the JRE and Swing.

There should be automated test verifying some use cases. Code related to the GUI could be tested manually.

2.2.5 Implementation

To achieve platform independence, the application will use the Java Runtime Environment (JRE). Users must have the JRE installed and configured.

2.2.6 Packaging and installation

The application will be delivered as an executable jar file which follows the concept of plug and play meaning no installation is required.

2.2.7 Legal

The application makes use of the open source external library iText, which is licensed under the Affero General Public License. Readable here: <http://itextpdf.com/terms-of-use/agpl.php>

2.3 Application models

2.3.1 Use case model

See APPENDIX for UML diagram and textual descriptions.

2.3.2 Use cases priority

1. Enter text

2. Save doc
3. Open doc
4. Import image
5. Export Doc
6. Undo action
7. Redo action
8. Cut/copy/paste text
9. Print doc
10. Crop/resize/reorient image
11. Create new doc
12. Find text
13. Replace text
14. Format text (font, size, colour, bold/italic)
15. Change template
16. Navigate open docs
17. Rename doc
18. Grayscale image
19. View help
20. View info

2.3.3 Domain model

See APPENDIX.

2.3.4 User interface

Application will use a fixed (non-skinable, non-themeable) GUI following standard conventions. See APPENDIX for screens.

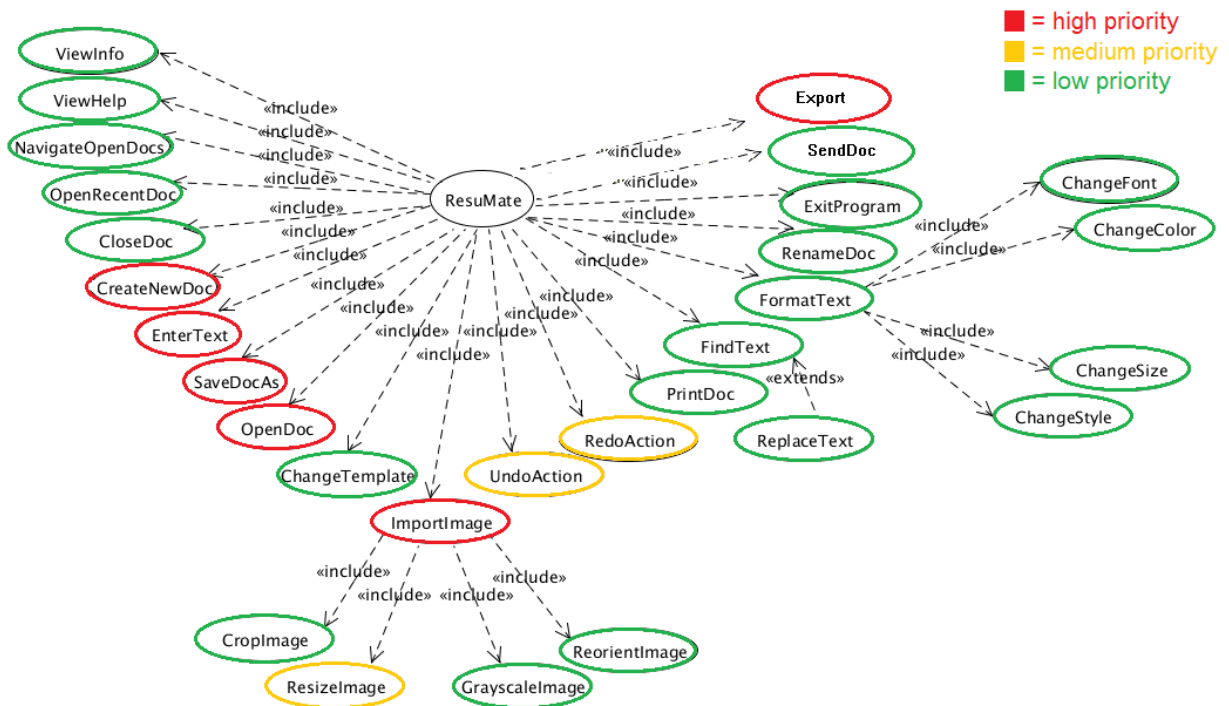
2.4 References

iText: <http://itextpdf.com/>

APPENDIX

Use cases

Overview.



Use case texts

Use Case: Enter Text

Summary: An action in which the user enters a text on the text area and the program corresponds by displaying this text.

Priority: High

Extends: -

Includes: -

Participators: The user

Normal flow of events:

	Actor	System
1	Enters text	
2		Text appears on the working space

Use Case: Change Template

Summary: A function that allows you to use a completed template.

Priority: high

Extends: -

Includes: -

Participators: The user

Normal flow of events:

	Actor	System
1	Selects a template	
2		The template appears on workspace

Use Case: Import Image

Summary: This is how the user imports an image, presumably a photo of oneself, to use in one's resume. The image will be automatically scaled. If a template with a designated area for an image is chosen, the image is automatically placed in that area. For parts of GUI (Import Image button, etc.), see GUI sketch.

Priority: high

Extends: -

Includes: Scale image

Participators: The user

Normal flow of events

Importing an image without making any changes to it.

	Actor	System
1	Clicks Image -> Import in Menu	
2		Opens File browser
3	Finds and selects image to import	
4	Clicks Open button	
5		Closes file browser
6		Places image in doc in designated position

Use Case: Grayscale image

Summary: Used whenever the user would like to grayscale the image in the document.

Priority: low

Extends: -

Includes:

Participators: The user

Normal flow of events:

	Actor	System
	Clicks on the option “Grayscale” in the menu under Image.	
		The image in the document becomes grayscaled.

Use Case: Edit Content (cut)

Summary: Used whenever the user wants to cut a symbol or a text.

Priority: high

Extends: -

Includes: -

Participants: The user

Normal flow of events:

	Actor	System
	Marks a text	
		The selected text is highlighted
	Clicks on “Cut”-button	
		The text copies
		The text deletes

Use Case: Edit Content (copy)

Summary: Used whenever the user wants to copy a symbol or a text.

Priority: high

Extends: -

Includes: -

Participants: The user

Normal flow of events:

	Actor	System
1	Marks a text	
2		The selected text is highlighted
3	Clicks on "Copy"-button	
4		The text is copied

Use Case: Edit Content (paste)

Summary: Used whenever the user wants to paste a copied symbol or a text.

Priority: high

Extends: -

Includes: -

Participants: The user

Normal flow of events:

	Actor	System
1	Clicks on the "Paste"-button	
2		The copied text appears

Use Case: Change Font

Summary: Used whenever the user wants to change the current marked font.

Priority: high

Extends: -

Includes: It continues being written in the selected font even after the user has marked a text.

Participants: The user

Normal flow of events:

	Actor	System
1	Clicks on a drop-down	
2		Shows a list
3	Selects a font	
4		The font is selected
		The font is applied to the text

Use Case: Format Text (Size)

Summary: : An action which formats the text by applying various sizes to enhance its visual representation.

Priority: High

Extends: -

Includes: -

Participators: The user

Normal flow of events:

	Actor	System
1	Clicks on the dropdown arrow of next to the field with a number	
2		A dropdown is displayed with numbers
3	Chooses a number	
4		The number which was in the field is now updated to the chosen one
5	The dropdown is now closed	
6		The text will have the specific size applied to it.

Use Case: Format Text (Colour)

Summary: : An action which formats the text by applying different colours to enhance its visual representation.

Priority: High

Extends: -

Includes: -

Participators: The user

Normal flow of events:

	Actor	System
1	Clicks on the dropdown arrow of the button with a coloured letter.	
2		Dropdown with alternatives is shown
3	Chooses a colour	
4		The dropdown is now closed
5		The colour which was given to the letter is now updated to the chosen one
6		The text will have the specific colour applied to it.

Use Case: Format Text (Style)

Summary: : An action which formats the text by applying different styles to enhance its visual representation.

Priority: High

Extends: -

Includes: -

Participators: The user

Normal flow of events:

	Actor	System
1	Clicks on the button with the icon "B"(stands for bold) "I" (stands for italic) or " <u>"(stands for underlined)</u>	
2		The button is pressed
3		The text will have the specific style applied to it.

Use Case: Undo An Action

Summary: Whenever the user has made an action, this use case lets the user intercept the previous action.

Priority: high

Extends: -

Includes: A possibility to redo a previous action that the user choose to undo.

Participators: The user

Normal flow of events:

	Actor	System
1	Deletes a text	
2		The text is removed
3	Clicks on the "Undo"- button	
4		The removed text appears

Use Case: Redo An Action

Summary: Whenever the user has made an action, this use case lets the user intercept the previous action which he undid.

Priority: high

Extends: -

Includes: A possibility to redo a previous action that the user wants to redo.

Participators: The user

Normal flow of events:

	Actor	System
1	Deletes	
2		The text is removed
3	Clicks on the “Undo”- button	
4		The removed text appears
5	Clicks on the “Redo” - button.	
6		The removed text is removed from the screen once again.

Use Case: Find Text

Summary: Used whenever the user would like to find a certain text in the document.

Priority: medium

Extends: -

Includes:

Participators: The user

Normal flow of events:

	Actor	System
1	Clicks on the option “Find text” in the menu under Edit.	
2		A pop-up is shown for input of the text to be found
3	Types the text to be found	

4		Highlights the text to be found
---	--	---------------------------------

Use Case: ReplaceText

Summary: Used whenever the user would like to replace a text with another.

Priority: medium

Extends: -

Includes:

Participants: The user

Normal flow of events:

1	Clicks on the option "Replace" in the menu under Edit.	
2		A pop-up is shown for input of the the text to be replaced and to be replaced with.
3	Types the text to be replaced and to be replaced with.	
4		Shows the text with the text to be replaced with the text to be replaced with.

Use Case: Save document as

Summary: An action which saves current document in another format in favor of the user.

Priority: High

Extends: -

Includes: -

Participants: The user

Normal flow of events:

	Actor	System
1	Clicks on the option "Save..." in the menu under File or the	

	Save button	
2		Opens a file chooser
3	Browses for desired path	
4		Shows directories
...	Repeat until satisfied	
...		Repeat as demanded
5	Enter desired name for document	
	Press save	
6		The file chooser closes
7		The files are saved

Alternate flows

Flow 3.1: A document with the same name and format does already exist

3.1		The system saves the file with a number appended.
-----	--	---

Use Case: Open document

Summary: An action which opens a document affirmatively of the user.

Priority: High

Extends: -

Includes: Choosing a document from a new window .

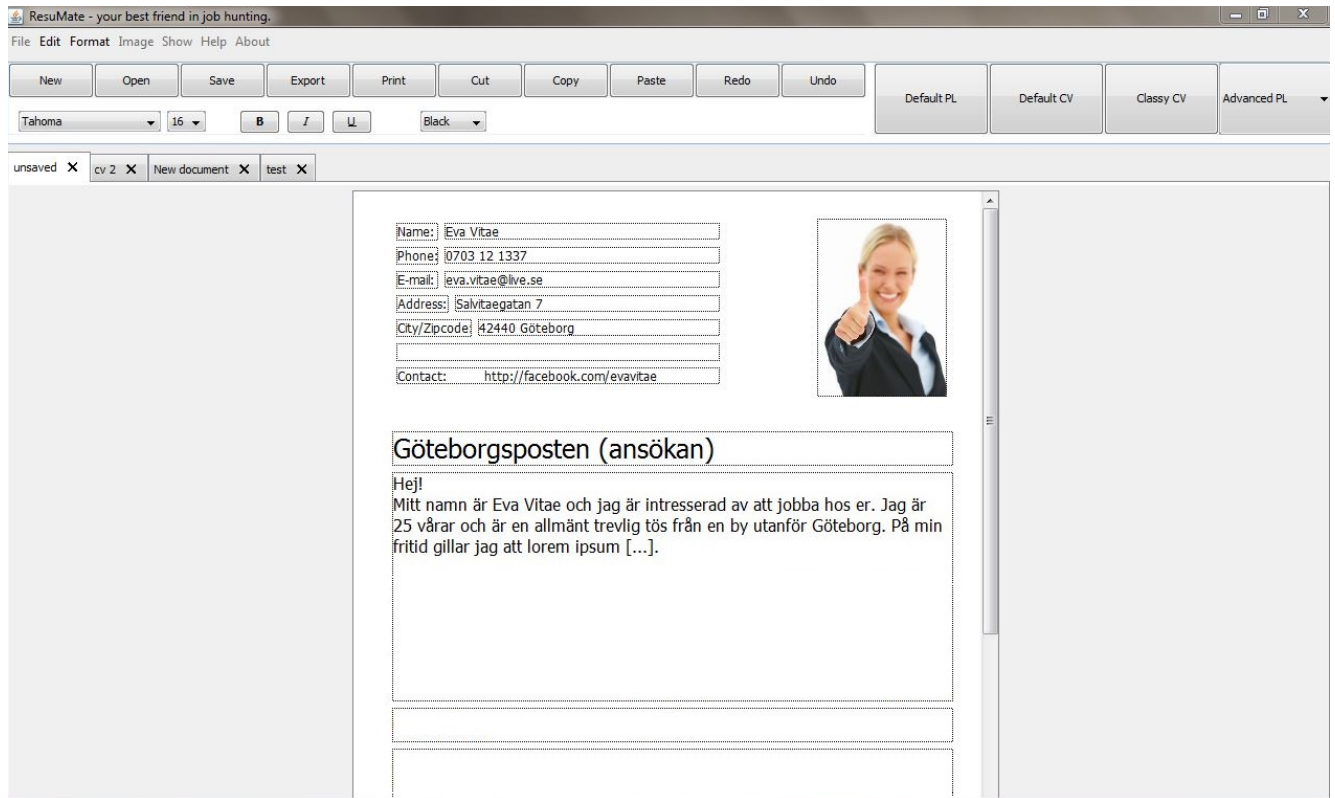
Participators: The user

Normal flow of events:

	Actor	System
1	Clicks on the option "Open" in the menu under File or in the	

	button in the toolbar	
2		Opens a file chooser
3	Browses for file	
4		Shows directories
...	Repeat until satisfied	
...		Repeat as demanded
5	Choose file to open	
6		The file choose closes
7		The chosen document will be opened

GUI



Domain model

