

SE 3XA3: Development Plan
Title of Project

Team #, Team Name
Student 1 name and macid
Student 2 name and macid
Student 3 name and macid

October 3, 2016

Contents

1	Project Drivers	1
1.1	The Purpose of the Project	1
1.2	The Stakeholders	1
1.2.1	The Client	1
1.2.2	The Customers	1
1.2.3	Other Stakeholders	1
1.3	Mandated Constraints	1
1.4	Naming Conventions and Terminology	1
1.5	Relevant Facts and Assumptions	1
2	Functional Requirements	1
2.1	The Scope of the Work and the Product	1
2.1.1	The Context of the Work	1
2.1.2	Work Partitioning	1
2.1.3	Individual Product Use Cases	2
2.2	Functional Requirements	2
3	Non-functional Requirements	4
3.1	Look and Feel Requirements	4
3.2	Usability and Humanity Requirements	4
3.3	Performance Requirements	4
3.4	Operational and Environmental Requirements	4
3.5	Maintainability and Support Requirements	4
3.6	Security Requirements	4
3.7	Cultural Requirements	4
3.8	Legal Requirements	4
3.9	Health and Safety Requirements	4
4	Project Issues	5
4.1	Open Issues	5
4.2	Off-the-Shelf Solutions	5
4.3	New Problems	5
4.4	Tasks	5
4.5	Migration to the New Product	5
4.6	Risks	5
4.7	Costs	5

4.8	User Documentation and Training	5
4.9	Waiting Room	5
4.10	Ideas for Solutions	5
5	Appendix	6
5.1	Symbolic Parameters	6

List of Tables

1	Revision History	ii
---	-----------------------------------	----

List of Figures

1	UML Use Case Diagram	2
---	--------------------------------	---

Table 1: **Revision History**

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

This document describes the requirements for The template for the Software Requirements Specification (SRS) is a subset of the Volere template (Robertson and Robertson, 2012). If you make further modifications to the template, you should explicitly state what modifications were made.

1 Project Drivers

1.1 The Purpose of the Project

1.2 The Stakeholders

1.2.1 The Client

1.2.2 The Customers

1.2.3 Other Stakeholders

1.3 Mandated Constraints

1.4 Naming Conventions and Terminology

1.5 Relevant Facts and Assumptions

User characteristics should go under assumptions.

2 Functional Requirements

2.1 The Scope of the Work and the Product

2.1.1 The Context of the Work

The context of the application is that it will be executed on a windows machine, the machine itself can not be harmed by execution of the application. The developers must be able to understand image manipulation on a software level to tackle many of the problems at hand.

2.1.2 Work Partitioning

For all work partitioning refer to the Gantt Chart.

2.1.3 Individual Product Use Cases

The following image is the Use Case representation of Gifitti using UML.

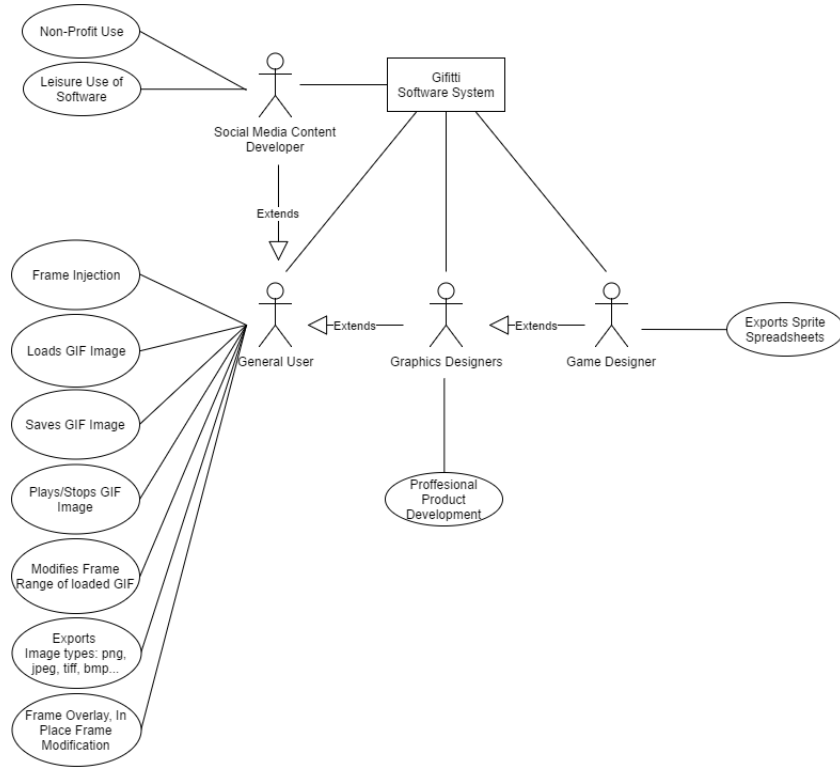


Figure 1: UML Use Case Diagram

2.2 Functional Requirements

1. The user is able to open a GIF from a specified location.
2. The user is able to save a GIF to a specified location.
3. The user can save a GIF to the current location of an opened GIF.
4. The user can specify the saved name of the GIF.
5. The user can specify to export a GIF as a sprite spreadsheet, to a specified location.

6. A Command must exist to allow the GIF to be played, this command only works if the GIF is stopped.
7. A Command must exist to allow the GIF to be stopped, this command only works if the GIF is playing.
8. The user is given control over the stop and start commands.
9. A range of frames may be specified and extracted as another GIF image.
10. Any GIF or specified range of frames in a GIF can be exported as a series of frames.
11. Series of frames may be PNG, JPEG, BMP, TIFF, or any other standard image format.
12. The system must ensure the all files read in is of a proper format.
13. The system can set the playback speed of GIFs.
14. Users have the ability to reset the GIF to what was originally loaded.
15. The system must be integrated with a help context available to the user.
16. The system is to allow for frame injection into the GIF.
17. For frames added to a GIF not of the proper size, the GIF is either scaled to fit, or the frame is scaled to fit.
18. Frame modification should be able to be done in place on the application.
19. Frame modification allows users to draw on or place images on existing frames.
20. All modifications can be placed on adjacent frames on user request.

3 Non-functional Requirements

3.1 Look and Feel Requirements

3.2 Usability and Humanity Requirements

3.3 Performance Requirements

3.4 Operational and Environmental Requirements

3.5 Maintainability and Support Requirements

3.6 Security Requirements

3.7 Cultural Requirements

3.8 Legal Requirements

3.9 Health and Safety Requirements

This section is not in the original Volere template, but health and safety are issues that should be considered for every engineering project.

4 Project Issues

4.1 Open Issues

4.2 Off-the-Shelf Solutions

4.3 New Problems

4.4 Tasks

4.5 Migration to the New Product

4.6 Risks

4.7 Costs

4.8 User Documentation and Training

4.9 Waiting Room

4.10 Ideas for Solutions

References

James Robertson and Suzanne Robertson. *Volere Requirements Specification Template*. Atlantic Systems Guild Limited, 16 edition, 2012.

5 Appendix

This section has been added to the Volere template. This is where you can place additional information.

5.1 Symbolic Parameters

The definition of the requirements will likely call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance.