SE 3XA3: Software Requirements Specification
Ratava

Team 9, Makiam Group
Aidan McPhelim - mcpheima
Alexie McDonald - mcdona16
Illya Pilipenko - pilipeni

December 6, 2018

Contents

List of Tables

List of Figures

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

1 Project Drivers

1.1 The Purpose of the Project

Internet avatars became prevalent several years ago in online forums. They were originally not a part of standard practice, but were added in order to personalize responses and provide a sense of identity.

The purpose of this project is to give users a program which can generate unique Avatars for them based on a string of their choosing, which will allow them to share it with their friends if they wish.

1.2 The Stakeholders

1.2.1 The Client

Since the Professor and TA's of the class want their students to be successful and for the class to do well, they can be considered stakeholders for the software.

Table 1: **Revision History**

Date	**Version**	**Notes**
2018-10-03	1.0	Created initial Draft, added F and NF requirements
2018-10-05	1.1	Filled in the remaining sections
2018-10-05	1.2	Finalized formatting
2018-12-03	1.3	Updated team name
2018-12-05	1.4	Removed old use cases 3, 5, 7, 8

1.2.2 The Customers

The customers will be internet users at large. Users of social media websites, forums, instant messaging clients to name a few will make good use of an implementation that addresses this problem.

1.2.3 Other Stakeholders

There are no further stakeholder apparent for this project.

1.3 Mandated Constraints

The main constraint we have is the time constraint - in that we need to have the entire project done with all accompanying documentation by December the 5th.

1.4 Naming Conventions and Terminology

Below is a glossary of terms used throughout the specification document

Table 2: Glossary of terms

Word/Phrase	Definition
Colour set	The colours selected to colour regions of the avatar
Item set	The templates of items that look like every day items (e.g. sword, apple)
Template	An image that shows what pixels will be filled where
Hash code	The output from a hash function (usually a string or hexadecimal)
Avatar	An icon used by an individual to identify themselves in a (typically) online community

1.5 Relevant Facts and Assumptions

Users of the program are most likely going to be people who are used to computers and play games etc. but when designing the program and UI we will assume that users have no prior knowledge of the terminology involved.

2 Functional Requirements

2.1 The Scope of the Work and the Product

2.1.1 The Context of the Work

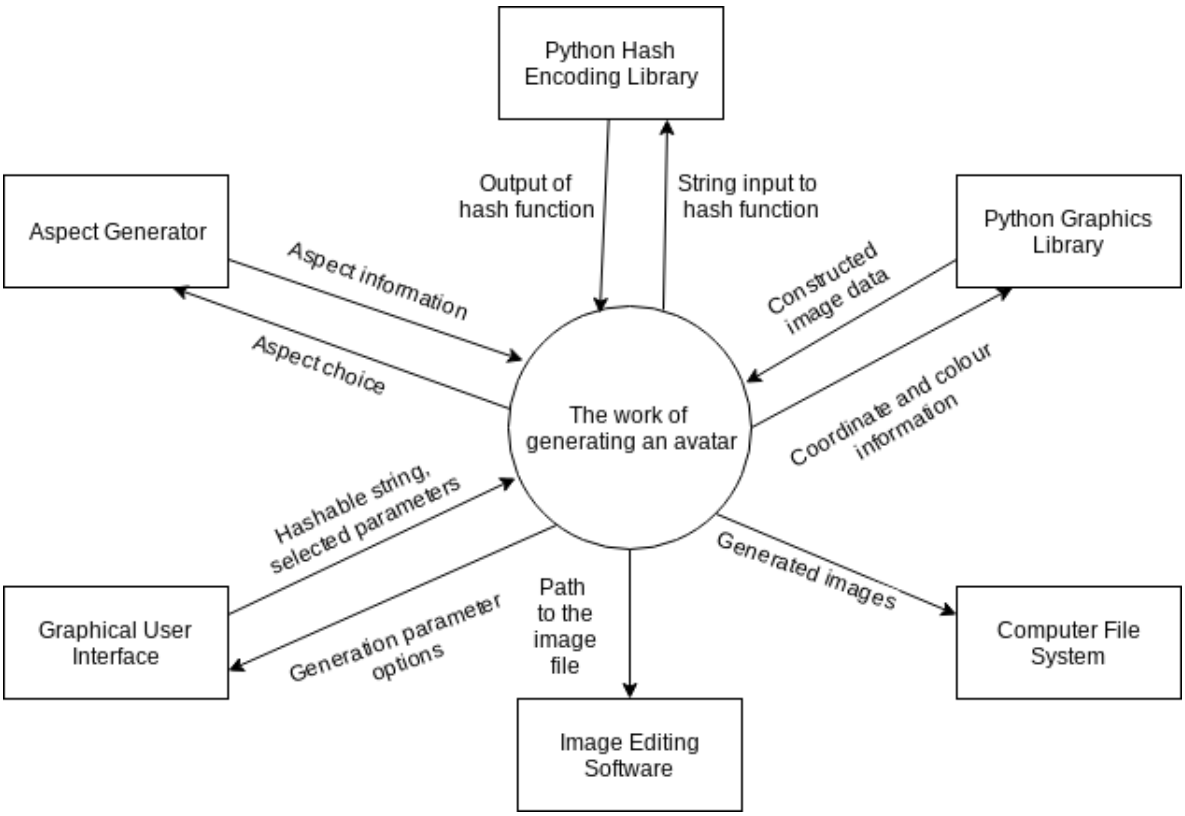

Figure 1: Context diagram

2.1.2 Work Partitioning

Table 3: Work Partitioning Table

Event Name	Input and Output	Summary
User enters seed string	input string (in)	Record the seed string in a variable within the program
User selects template	template key(in)	Set the value of the template variable to the template for the key specified
User presses generate button	start generation signal(in)	Software begins generation process by triggering the hash function event
Hash function event triggered	input string variable value (in), hash code (out)	Software passes input string to hash function and records output
Colour generation by hash code	hash code(in), hexadecimal colour code(out)	Software appends components of the hash code, then converts to hexadecimal to use for the colours
Weapon generation by hash code portion	hash code (in), weapon pattern (out)	Software uses the first two numeric values of the hash code to index an element in the weapons array
Picture assembly by property input	template(in), colour mapping(in), weapon key(in), image data(out)	Software passes the coordinate fill information from the template, colour values, and weapons to generate the image file
Program writes file to file system	image data(in), image file(out)	Software writes the image data to a file in the desired location in the file system

2.1.3 Individual Product Use Cases

Table 4: Use Case Table

Actor	Actor's Goal	Use Case Name
User	To generate an avatar with all parameters set at random	GenCompleteRand (UC-1)
User	To generate an avatar of a specific template with all other parameters set at random	GenRandLessTemp (UC-2)
User	To generate an avatar of a specific template and colour set with all other parameters set at random	GenRandLessTempColour (UC-3)
User	To generate an avatar of a specific colour set with all other parameters set at random	GenRandLessColour (UC-4)

1. **Product Use Case Name:** GenCompleteRand (UC-1)
    - **Trigger:** User presses generate button
    - **Preconditions:** N/A
    - **Interested stakeholders:** The user
    - **Actor:** User
    - **Outcome:** The avatar is generated with no parameters predetermined
2. **Product Use Case Name:** GenRandLessTemp (UC-2)
    - **Trigger:** User presses generate button $\wedge$ User selects template
    - **Preconditions:** N/A
    - **Interested stakeholders:** The user
    - **Actor:** User
    - **Outcome:** The avatar is generated with the selected template and no other parameters predetermined
3. **Product Use Case Name:** GenRandLessTempColour (UC-3)
    - **Trigger:** User presses generate button $\wedge$User selects property to fix $\wedge$ User selects template

- **Preconditions:** N/A
- **Interested stakeholders:** The user
- **Actor:** User
- **Outcome:** The avatar is generated with the selected template, selected colour set, and no other parameters predetermined

4. **Product Use Case Name:** GenRandLessColour (UC-4)
    - **Trigger:** User presses generate button $\wedge$ User selects property to fix
    - **Preconditions:** N/A
    - **Interested stakeholders:** The user
    - **Actor:** User
    - **Outcome:** The avatar is generated with the selected colour set, and no other parameters predetermined

2.2 Functional Requirements

- R1 : Software shall output an image of type .jpg
    - **Fit Criteria:** The resulting file is a lossy graphics file, the extention is .jpg, and it can be opened given the file extension
- R2 : Software shall use a hash function to randomize colours and items
    - **Fit Criteria:** The resulting output of the hash function is classified as a hash function of the encoding specified that has the following properties: Compression, strong collision resistance, and the one-way property
- R3 : Software shall allow user to select what qualities to fix (and by extention, what qualities to randomize)
    - **Fit Criteria:** The qualities set prior to generation are preserved in the result
- R4 : Software shall allow user to select an avatar template

  - **Fit Criteria:** The template set prior to generation is preserved in the result

- R5 : Software shall take to a string as input that is used to seed the hash function
  - **Fit Criteria:** The string seeded produces a consistent result with the given hash encoding

- R6 : Software shall allow the user to select what hash function that is being used to randomize the elements
  - **Fit Criteria:** The hash function selected, when given the input string, produces the same output as used in the program

- R7 : Software shall allow the user to specify the output path
  - **Fit Criteria:** The string representing the location of the file and the string the user entered to determine the location are equal

- R8 : Software shall save the image in the location that the user specifies
  - **Fit Criteria:** Indexing into that location in the file system allows the user to locate and open the output file

- R9 : Software shall provice access the help documentation
  - **Fit Criteria:** The help implementation can be opened and read in its entirety

- R10 : Software shall open the image instantly in an external image editor upon user's request
  - **Fit Criteria:** An image editor opens within 0.5 seconds of the request with the output file in it

- R11 : Software shall allow the user to modify the name of the output file
  - **Fit Criteria:** The name provided by the user and the name of the output file are equal

3 Non-functional Requirements

3.1 Look and Feel Requirements

3.1.1 Accessiblity

- R12 : The software shall implement accessible fonts. That would entail fonts that are sans-serif, simple, and highly legible
  - **Fit Criteria:** All guidelines in the Web Content Accessibility Guidelines are fulfilled to at least the A standard **?**
- R13 : The user interface elements (buttons, text) should have a high contrast with the background, to aid colour-blind users
  - **Fit Criteria:** Ensure the standards outlined in the World Wide Web Consortium for colour contrast and accessibility are met
- R14 : The user interface shall be compatible with a screen reader to aid those who are visually impaired
  - **Fit Criteria:** The running of a screen reader program produces a result that is clear and understandable
- R15 : The software shall use intuitive keystrokes throughout the user interface to the meaning of certain operations can be inferred.
  - **Fit Criteria:** A test of the system with a user who did not consult the help documentation yields success when the user is asked to fulfill all use cases of the system

3.2 Usability and Humanity Requirements

3.2.1 Ease of Use

- R16 : User interface will be simple, so people who are 8 or above will be able to use it.
  - **Fit Criteria:** 85% of users who attempt to use the software (out of a statistically random sample) can use the software with little difficulty

3.2.2 Ease of Learning

- R17 : Any user with basic knowledge of a computer's file system will be able to learn this software
  - **Fit Criteria:** 85% of people who read the help documentation and fulfil the above criterion can use the software with ease

3.3 Performance Requirements

3.3.1 Speed Requirements

- R18 : The algorithm used to generate should minimize wait times for the user
  - **Fit Criteria:** The trace of the algorithm should be proven through optimization to be the shortest possible route
- R19 : The program at most should take no longer than 5 seconds maximum. A study performed in 1995 says that 10 seconds is a tolerable wait time for a web response. Moreover, 1 second is the limit for the user's train of thought to flow continously. **?**
  - **Fit Criteria:** The difference in the moment the generator button is pressed and the result becomes avaliable in the file system should be no greater than 5 seconds

3.3.2 Precision Requirements

- R20 : Pixels displayed to the user shall exactly match the pixel placement outlined in the templates
  - **Fit Criteria:** $\forall a \in template \wedge b \in output \bullet a(x, y) = b(x, y)$

3.3.3 Reliability and Avaliability

- R21 : The software will be operable 24hrs a day, 7 days a week, 365 days a year, Due to the program being on a local machine with no dependency on any external resources other than a working file system and python compiler.

- **Fit Criteria:** An ongoing stress test of the system for the time specified above yields no blackouts or interuptions

3.3.4 Capacity Requirements

- R22 : The software shall have an ultimited amount of concurrent users on unique machines, Due to the program being on a local machine.
  - **Fit Criteria:** Have 10 users use the program concurrently and record the difference in time to only one person running it
- R23 : The software shall have a limit of one concurrent user per machine.
  - **Fit Criteria:** Attempts to have two users use the same program concurrently fail

3.3.5 Expected Technological Environment

- R24 : The expected technological environment will ultimately be a computer that is running an operating system that can support usage of a python3 compiler and interpreter, and has a traditional file system.
  - **Fit Criteria:** Download a selection of the most popular virtual machine environments and ensure the software runs on each

3.3.6 Partner Applications

- R25 : There is no partner applications that the software will need to interface with.
  - **Fit Criteria:** N/A

3.4 Maintainability and Support Requirements

3.4.1 Mainainability

- R26 : The software shall support the addition of new templates at a later date

  - **Fit Criteria:** An addition of a new template after software release fails to change program behaviour negatively

- R27 : The software shall support the addition of new hash functions at a later date
  - **Fit Criteria:** An addition of a new hash encoding after software release fails to change program behaviour negatively

- R28 : The software shall support the addition of new elements to the user interface at a later date
  - **Fit Criteria:** An addition of a new user interface element after software release fails to change program behaviour negatively

- R29 : The software shall be decomposed into modules with a level of low cohesion that allows them to be serviced without breaking the other modules
  - **Fit Criteria:** The modification of a module after release passes all integration tests

3.4.2 Portability

- R30 : As outlined in Section 3.4.5 the software shall run on any operating system that supports the usage of a python compiler, interpreter, and modifications to the files present on the file system
  - **Fit Criteria:** Same as in Section 3.4.5

3.5 Security Requirements

- R31 : The software shall dispose of all strings entered that hashed the functions
  - **Fit Criteria:** A query of the variables/data accessible by the user after run time yields no strings provided

- R32 : Randomness of the hash function shall ensure that the hash string cannot be infered from the template it generates

  - **Fit Criteria:** An attempt to reverse the hash function yields a result different from the one entered

- R33 : The software shall prevent data interceptions by running as a standalone application on a desktop computer

  - **Fit Criteria:** Attempts at a data interception via any method except unauthorized remote access is unsuccessful

3.6 Cultural Requirements

- R34 : The software shall not use any symbols or imagery for templates that could offend the country using it

  - **Fit Criteria:** Templates, items, and colour sets used do not resemble any content deemed to be sensitive

3.7 Legal Requirements

- R35 : The elements of the software shall support the requirements outlined in the licence underwhich the software is licenced.

  - **Fit Criteria:** Every regulation outlined in the General Public Licence 2.0 documentation is fulfilled **?**

- R36 : The software shall be compliant with all privacy regulations

  - **Fit Criteria:** Every regulation in the The Personal Information Protection and Electronic Documents Act (PIPEDA) is satisfied **?**

3.8 Health and Safety Requirements

Not applicable to this software.

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

None have been found so far.

4.2 Off-the-Shelf Solutions

There are already similar products available, such as https://robohash.org/, but string-hash based random avatar generation is a business field where getting off-the-shelf solutions from other developers for the purpose of using them in one's own product is a very difficult task.

4.3 New Problems

This software main problem that will exist after this software is in place is the restriction on resolution of avatar images. Users can of course still have higher resolution images, but in terms of generating random avatars that are unique to them, they will have to settle for a very low resolution.

4.4 Tasks

- plan for proof of concept demonstration
- create a test plan
- design and document revision 0
- prepare for revision 0 demonstration
- research into pixel art
- create the template for software design
- start work on the program

4.5 Migration to the New Product

Since the project's service is simple, discrete, and has data stored only on the user's side, there are no requirements for migration to the new product, nor does any data need to be modified or translated for the new system.

4.6 Risks

Risk	probability	countermeasure
Low productivity	high	implement agile method
Excessive schedule pressure	medium	reduce task load, implement agile method
Overabundance of features	low	remoe unnecessary features

4.7 Costs

As the resources being drawn upon for the creation of this project have no financial cost, the only cost to consider is the cost of time and effort. As outlined before, this project is well within the scope of a semester's worth of work split amongst three students, so the resource cost will be moderate.

4.8 User Documentation and Training

- User Documentation Requirements: an opeations manual describing how to use the product and what to expect from it should be provided alongside the product.
- Training Requirements: no training is needed to use the product.

4.9 Waiting Room

There are no requirements in the waiting room.

4.10 Ideas for Solutions

There are no ideas for solutions.

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.