Table 1: Revision History

Date	**Developer(s)**	**Change**
November 10, 2017	Thomas Mullen	Rev.0 of Document
November 10, 2017	Phillip Pavlich	Module M1 Section
...	...	...

3XA3 Module Interface Specification

Group 20 (2020Vision)
Mullen, Thomas - mullentc - 001406837
Pavlich, Phillip - pavlicpm - 001414960
Bauer, Ivan - bauerim - 001418765
L02

1 Module M1A (BrowserAction/UserInterface)

This section consists of the elements that are required for the user interface. It provides the user with a basic format to the display. Since it is a display, it does not have methods and state variables. It just displays the password from the Hasher module.

2 Module M1B (BrowserAction/Customization)

This section consists of css code to customize the user interface and improve the view for the user. It styles all the elements so that they are readable and gives an appealing view.

3 Module M1C (BrowserAction/EventListeners)

This section consists of the basic event listeners for the user interface. It has no state variable, only events that can be triggered by actions from the user. It has the following methods:

infoView: This method allows the user to click the info button to display more information about the application
tabChanged: This method detects if the user switches tabs to reflect the current domain
generate: This is a call to the hasher module to obtain the password and display it.
togglePass: This method toggles the view of the user key to make it either visible or hidden.

4 Module M2A (hasher/index)

$$generatePassword(master, domain) \tag{1}$$

$$result = applyConstraints(generateHash(master{+}domain{+}getSalt()), opts) \tag{2}$$

Name	hasher/index
Imported Indentifiers	*String* (String data type) *Object* (Object data type) *generateHash* (hasher/hash) *applyConstraints* (hasher/constraints) *getSalt* (hasher/salt)
Exported Access Routines	*init*, *generatePassword*
State Variables	None
State Invariant	$opts \in Object$
Assumptions	*init* called first

Table 2: Access Routine Semantics

Name	IO Relation	Domain
init()	$opts = \{\}$	None
(1)	(2)	$master \in String \wedge domain \in String$
setOptions(x)	$opts = x$	$x \in Object$

5 Module M2B (hasher/hash)

Name	hasher/hash
Imported Indentifiers	*String* (String data type) *Buffer* (Buffer data type) *hash* (function that is computationally infeasible to reverse)
Exported Access Routines	*generateHash*
State Variables	None
State Invariant	*true*
Assumptions	None

Table 3: Access Routine Semantics

Name	IO Relation	Domain
$generateHash(x)$	$result = hash(x) \wedge result \in String$	$x \in String \vee x \in Buffer$

6 Module M2C (hasher/unicode)

Name	hasher/unicode
Imported Indentifiers	$String$ (String data type) $CharacterType$ (Classes of Unicode character (special character, letter, number, alphanum, etc.))
Exported Access Routines	$fromHexCode$
State Variables	None
State Invariant	$true$
Assumptions	None

Table 4: Access Routine Semantics

Name	IO Relation	Domain
$fromHexCode(code, type)$	$result \in type \wedge result \in String$	$code \in String \wedge type \in CharacterType$

7 Module M2D (hasher/constraints)

Name	hasher/constraints
Imported Indentifiers	*String* (String data type) *Object* (Object data type) *CharacterType* (Classes of Unicode character (special character, letter, number, alphanum, etc.))
Exported Access Routines	*applyConstraints*
State Variables	None
State Invariant	*true*
Assumptions	None

Table 5: Access Routine Semantics

Name	IO Relation	Domain
$applyConstraints(x, opts)$	(3)	$x \in String \wedge opts \in Object$

$$result \in String \wedge ((\exists_n type \in result) \forall type \in keys(opts) \wedge n = opts[type]) \tag{3}$$

8 Module M2E (hasher/salt)

Name	hasher/salt
Imported Indentifiers	$String$ (String data type) $Buffer$ (String data type)
Exported Access Routines	$setSalt$, $getSalt$
State Variables	$salt$
State Invariant	$salt \in Buffer \vee salt = null$
Assumptions	None

Table 6: Access Routine Semantics

Name	IO Relation	Domain
$setSalt(x)$	$salt = x$	$x \in Buffer$
$getSalt()$	$result = toString(salt)$	$true$

9 Module M3A (options/constraints)

Name	options/constraints
Imported Indentifiers	N Natural numbers $(0, \infty]$
Exported Access Routines	$setLength$, $getLength$
State Variables	$length$
State Invariant	$length \in N$
Assumptions	None

Table 7: Access Routine Semantics

Name	IO Relation	Domain
$setLength(x)$	$length = x$	$x \in N$
$getLength()$	$result = length$	$true$

10 Module M3B (options/display)

Name	options/display
Imported Indentifiers	*Object* (Object data type)
Exported Access Routines	*setOptions*, *getOptions*
State Variables	*opts*
State Invariant	$opts \in Object$
Assumptions	None

Table 8: Access Routine Semantics

Name	IO Relation	Domain
$setOptions(x)$	$opts = x$	$x \in Object$
$getOptions()$	$result = opts$	$true$