Tasks:

1. **UML Diagram**
    - Ensure the UML diagram is created and up to date.
2. **Update README**
    - Add information about Sprint 1 planning, the Sequence Diagram, the ARTIFACTS folder, and the Google Slides presentation.
3. **Add Artifacts to the ARTIFACTS Folder**
    - Organize all relevant design documents, diagrams, and additional project documentation in the artifacts folder.

Links:

- Google Slides Presentation: [Sprint 1 Presentation]

Sequence Diagram Code Below

```
title WARP Diagram

actor user

user->Warp: main(new String[]{"gui", "ra"})
activate Warp

Warp->Warp:visualize(workLoad = warp, choice = SystemChoices.RELIABILITIES)
activate Warp

Warp->VisualizationFactory: viz = createVisualization(workLoad, outputDirectory =
outputSubDirectory, choice)
activate VisualizationFactory

VisualizationFactory->VisualizationImplementation: viz = new VisualizationImplementation(warp
= workLoad, outputDirectory, choice)
activate VisualizationImplementation

VisualizationImplementation->FileManager: fm = new FileManager()
activate FileManager
FileManager-->VisualizationImplementation: FileManager instance
deactivate FileManager

VisualizationImplementation->WarpInterface: workLoad = warp.toWorkLoad()
activate WarpInterface
```

```
note right of WarpInterface: TODO
WarpInterface-->VisualizationImplementation: WorkLoad instance
deactivate WarpInterface

VisualizationImplementation->WorkLoadDescription: inputFileName =
workLoad.getInputFileName()
activate WorkLoadDescription
WorkLoadDescription-->VisualizationImplementation: inputFileName
deactivate WorkLoadDescription

VisualizationImplementation->VisualizationImplementation: fileNameTemplate =
createFileNameTemplate(outputDirectory)
activate VisualizationImplementation

VisualizationImplementation->FileManager: workingDirectory = fm.getBaseDirectory()
activate FileManager
FileManager-->VisualizationImplementation: baseDirectory
deactivate FileManager

VisualizationImplementation->FileManager: newDirectory = fm.createDirectory(directory =
workingDirectory, subDirectory = outputDirectory)
activate FileManager

alt subDirectory.startsWith("/")
FileManager->FileManager: newDirectory = subDirectory
else
FileManager->FileManager: newDirectory = directory + File.separator + subDirectory
end

FileManager->FileManager: path = Paths.get(newDirectory)
FileManager->FileManager: Files.createDirectories(path)

opt verbose
FileManager->FileManager: System.out.println("Directory " + newDirectory + " is created!")
end

FileManager-->VisualizationImplementation: newDirectory
deactivate FileManager

alt inputFileName.contains("/")
  VisualizationImplementation->VisualizationImplementation: index =
inputFileName.lastIndexOf("/") + 1
  VisualizationImplementation->VisualizationImplementation: fileNameTemplate = newDirectory
+ File.separator + inputFileName.substring(index)
```

```
else
  VisualizationImplementation->VisualizationImplementation: fileNameTemplate = newDirectory
+ File.separator + inputFileName;
end

VisualizationImplementation-->VisualizationImplementation: fileNameTemplate
deactivate VisualizationImplementation

VisualizationImplementation->ReliabilityVisualization: obj = new ReliabilityVisualization(warp)
activate ReliabilityVisualization

ReliabilityVisualization->ReliabilityVisualization: this.warp = warp

ReliabilityVisualization->FileManager: fm = new FileManager()
activate FileManager
FileManager-->ReliabilityVisualization: FileManager instance
deactivate FileManager

ReliabilityVisualization->VisualizationObject: super(fm, warp, suffix = SOURCE_SUFFIX)
activate VisualizationObject

VisualizationObject->VisualizationObject: this.fm = fm

VisualizationObject->WorkLoad: m = workLoad.getMinPacketReceptionRate()
activate WorkLoad
WorkLoad-->VisualizationObject: minPacketReceptionRate
deactivate WorkLoad

VisualizationObject->WorkLoad: e2e = workLoad.getE2e()
activate WorkLoad
WorkLoad-->VisualizationObject: e2e
deactivate WorkLoad

VisualizationObject->VisualizationObject: nameExtension = String.format("-%sM-%sE2E",
String.valueOf(m), String.valueOf(e2e))

VisualizationObject->VisualizationObject: this.suffix = suffix
VisualizationObject->VisualizationObject: visualizationData = null

VisualizationObject-->ReliabilityVisualization: VisualizationObject instance
deactivate VisualizationObject

ReliabilityVisualization->WarpInterface: ra = warp.toReliabilityAnalysis()
activate WarpInterface
```

```
note left of WarpInterface: TODO
WarpInterface-->ReliabilityVisualization: ReliabilityAnalysis instance
deactivate WarpInterface

ReliabilityVisualization-->VisualizationImplementation: ReliabilityVisualization instance
deactivate ReliabilityVisualization

VisualizationImplementation->VisualizationImplementation: createVisualization(obj)
activate VisualizationImplementation

VisualizationImplementation->ReliabilityVisualization: visualization = obj.visualization()
activate ReliabilityVisualization

ReliabilityVisualization->VisualizationObject: visualization = super.visualization()
activate VisualizationObject

VisualizationObject->Description: content = new Description()
activate Description
Description->Description: super()
Description->VisualizationObject: Description instance
deactivate Description

VisualizationObject->ReliabilityVisualization: data = createVisualizationData()
activate ReliabilityVisualization
note right of ReliabilityVisualization: TODO
ReliabilityVisualization-->VisualizationObject: visualizationData
deactivate ReliabilityVisualization

VisualizationObject->ReliabilityVisualization: columnHeader = createColumnHeader()
activate ReliabilityVisualization
note right of ReliabilityVisualization: TODO
ReliabilityVisualization-->VisualizationObject: String[] instance
deactivate ReliabilityVisualization

VisualizationObject->VisualizationObject: nodeString = String.join("\t", columnHeader) + "\\n"
VisualizationObject->VisualizationObject: content.add(nodeString)

loop rowIndex = 0; rowIndex < data.length; rowIndex++
VisualizationObject->VisualizationObject: row = data[rowIndex]
VisualizationObject->VisualizationObject: rowString = String.join("\t", row) + "\\n"
VisualizationObject->VisualizationObject: content.add(rowString)
end

VisualizationObject-->ReliabilityVisualization: content
```

```
deactivate VisualizationObject

ReliabilityVisualization-->VisualizationImplementation: visualization
deactivate ReliabilityVisualization

VisualizationImplementation->ReliabilityVisualization: fileContent = obj.fileVisualization()
activate ReliabilityVisualization

ReliabilityVisualization->VisualizationObject: fileContent = super.fileVisualization()
activate VisualizationObject

VisualizationObject->ReliabilityVisualization: fileContent = createHeader()
activate ReliabilityVisualization
ReliabilityVisualization->Description: header = new Description()
activate Description
Description->Description: super()
Description->ReliabilityVisualization: Description instance
deactivate Description
note right of ReliabilityVisualization: TODO
ReliabilityVisualization-->VisualizationObject: header
deactivate ReliabilityVisualization

VisualizationObject->VisualizationObject: visualization = visualization()
activate VisualizationObject

VisualizationObject->Description: content = new Description()
activate Description
Description->Description: super()
Description->VisualizationObject: Description instance
deactivate Description

VisualizationObject->ReliabilityVisualization: data = createVisualizationData()
activate ReliabilityVisualization
note right of ReliabilityVisualization: TODO
ReliabilityVisualization-->VisualizationObject: visualizationData
deactivate ReliabilityVisualization

VisualizationObject->ReliabilityVisualization: columnHeader = createColumnHeader()
activate ReliabilityVisualization
note right of ReliabilityVisualization: TODO
ReliabilityVisualization-->VisualizationObject: String[] instance
deactivate ReliabilityVisualization

VisualizationObject->VisualizationObject: nodeString = String.join("\t", columnHeader) + "\\n"
```

```
VisualizationObject->VisualizationObject: content.add(nodeString)

loop rowIndex = 0; rowIndex < data.length; rowIndex++
VisualizationObject->VisualizationObject: row = data[rowIndex]
VisualizationObject->VisualizationObject: rowString = String.join("\t", row) + "\\n"
VisualizationObject->VisualizationObject: content.add(rowString)
end

VisualizationObject-->VisualizationObject: content
deactivate VisualizationObject

VisualizationObject->VisualizationObject: fileContent.addAll(visualization)

VisualizationObject->ReliabilityVisualization: footer = createFooter()
activate ReliabilityVisualization
ReliabilityVisualization->Description: footer = new Description()
activate Description
Description->Description: super()
Description->ReliabilityVisualization: Description instance
deactivate Description
note right of ReliabilityVisualization: TODO
ReliabilityVisualization-->VisualizationObject: footer
deactivate ReliabilityVisualization

VisualizationObject->VisualizationObject: fileContent.addAll(footer)

VisualizationObject-->ReliabilityVisualization: fileContent
deactivate VisualizationObject

ReliabilityVisualization-->VisualizationImplementation: fileContent
deactivate ReliabilityVisualization

VisualizationImplementation->ReliabilityVisualization: fileName =
obj.createFile(fileNameTemplate)
activate ReliabilityVisualization

ReliabilityVisualization->VisualizationObject: fileString = super.createFile(fileNameTemplate,
nameExtension, suffix)
activate VisualizationObject

VisualizationObject->FileManager: fileString = fm.createFile(file = fileNameTemplate,
nameExtension, suffix)
activate FileManager
```

```
FileManager->FileManager: suffixIndex = file.lastIndexOf('.')
FileManager->FileManager: fileString = file

opt suffixIndex > 0
FileManager->FileManager: fileString = file.substring(0, suffixIndex)
end

FileManager->FileManager: fileString = fileString + nameExtension + suffix

opt verbose
FileManager->FileManager: System.out.println("File " + fileString + " is created!")
end

FileManager-->VisualizationObject:
deactivate FileManager

VisualizationObject-->ReliabilityVisualization: fileString
deactivate VisualizationObject

ReliabilityVisualization-->VisualizationImplementation: fileString
deactivate ReliabilityVisualization

VisualizationImplementation->VisualizationImplementation: visualizationObject = obj
deactivate VisualizationImplementation

VisualizationImplementation-->VisualizationFactory: VisualizationImplementation instance
deactivate VisualizationImplementation

VisualizationFactory-->Warp: viz
deactivate VisualizationFactory

opt verbose
Warp->VisualizationImplementation: vizString = viz.toString()
activate VisualizationImplementation

VisualizationImplementation->Description: vizString = visualization.toString()
activate Description

Description->Description: sb = new StringBuffer()
loop row : this
Description->Description: sb.append(row)
end

Description-->VisualizationImplementation: sb.toString()
```

```
deactivate Description

VisualizationImplementation-->Warp: vizString
deactivate VisualizationImplementation

Warp->Warp: System.out.println(vizString)
end

Warp->VisualizationImplementation: viz.toFile()
activate VisualizationImplementation

VisualizationImplementation->Description: fileString = fileContent.toString()
activate Description

Description->Description: sb = new StringBuffer()
loop row : this
Description->Description: sb.append(row)
end

Description-->VisualizationImplementation: sb.toString()
deactivate Description

VisualizationImplementation->FileManager: fm.writeFile(file = fileName, fileContents = fileString)
activate FileManager
FileManager->FileManager: fileName = Path.of(file)
FileManager->FileManager: Files.writeString(fileName, fileContents)
deactivate FileManager

deactivate VisualizationImplementation

Warp->VisualizationImplementation: viz.toDisplay()
activate VisualizationImplementation

VisualizationImplementation->ReliabilityAnalysis: window =
visualizationObject.displayVisualization()
activate ReliabilityAnalysis
note right of ReliabilityAnalysis: TODO
ReliabilityAnalysis-->VisualizationImplementation: GuiVisualization instance
deactivate ReliabilityAnalysis

VisualizationImplementation->GuiVisualization: window.setVisible()
activate GuiVisualization
GuiVisualization->GuiVisualization: frame.setVisible(true)
deactivate GuiVisualization
```

deactivate VisualizationImplementation

deactivate Warp

deactivate Warp

README:
##CS 2820 Sprint1 -
##Jeff Bates
- responsible for writing and updating the README.md file, outlining your high-level plan and status.

##Cooper Fort
- Focus on creating the UML Sequence Diagram that shows the program flow starting from the Warp processing the 'ra' option
- Use sequencediagram.org to create the diagram.
- Ensure that the diagram clearly shows the sequence of interactions between program components.

##Colin Miller
- Worked closely with the sequence diagram designer to understand and implement the program logic for processing the 'ra' option in the Warp program.
- Assisted in writing pseudocode snippets for testing and implementing the 'ra' option.
- Document methods and flow in the README documents as needed.

##Yash Bandla
- Write how the project will be tested and documented.
- Make sure the plan is clear.

##Graham Besser
- Make a class diagram and a project plan that lists tasks in order
- Describes key methods for the Visualization class

##CS 2820 Sprint 2
-Updating ReliabilityVisualization code to be fully implemented
-Updating WARP Diagram to show changes to ReliabilityVisualization
-Adding JUnit Test to ensure stability of ReliabilityVisualization
-Adding JavaDoc comments to update documentation for functions and classes

##Jeff Bates
-Created JUnit Tests to test edge cases for all the methods in ReliabilityVisualization to ensure stability
- Tests for the following methods were created:

##Cooper Fort
-Updated sequencediagram.org with the new methods and added the diagram to ARTIFACTS folder
-Method calls that were updated: ReliabilityVisualization.getReliabilities, WorkLoad.getFlowNamesInPriorityOrder, WorkLoad.getNodesInFlow, createHeader, createColumnHeader, createVisualizationData, and saveVisualization

##Colin Miller
-

##Yash Bandla
-Responsible for updating the README.md file with high-level plans for updating ReliabilityVisualization with working code
-Clearly outlined everyone's tasks and explained all artifacts

##Graham Besser
-Created JavaDoc comments and updated documentation files to show each method's function and parameters in a high level manner

##ARTIFACTS
-Updated CS2820_Project ToDo with Sprint 2 tasks that need to be completed by every team member
-Created Sprint 2.pdf to show plans for staying organized, how we will complete every task, and modifications to UML Sequence/Model Diagrams and ReliabilityVisualization code
-Created a new WARP Sequence Diagram (titled WARP Diagram V2.jpg) that shows updated flow of all code (post ReliabilityVisualization code changes) in WARP codebase

Sprint 2 Sequence Diagram Code

```
title WARP Diagram

actor user

user->Warp: main(new String[]{"gui", "ra"})
activate Warp

Warp->Warp:visualize(workLoad = warp, choice = SystemChoices.RELIABILITIES)
activate Warp

Warp->VisualizationFactory: viz = createVisualization(workLoad, outputDirectory =
outputSubDirectory, choice)
activate VisualizationFactory

VisualizationFactory->VisualizationImplementation: viz = new VisualizationImplementation(warp
= workLoad, outputDirectory, choice)
activate VisualizationImplementation

VisualizationImplementation->FileManager: fm = new FileManager()
activate FileManager
FileManager-->VisualizationImplementation: FileManager instance
deactivate FileManager

VisualizationImplementation->WarpInterface: workLoad = warp.toWorkLoad()
activate WarpInterface
note right of WarpInterface: TODO
WarpInterface-->VisualizationImplementation: WorkLoad instance
deactivate WarpInterface

VisualizationImplementation->WorkLoadDescription: inputFileName =
workLoad.getInputFileName()
activate WorkLoadDescription
WorkLoadDescription-->VisualizationImplementation: inputFileName
deactivate WorkLoadDescription

VisualizationImplementation->VisualizationImplementation: fileNameTemplate =
createFileNameTemplate(outputDirectory)
activate VisualizationImplementation

VisualizationImplementation->FileManager: workingDirectory = fm.getBaseDirectory()
activate FileManager
FileManager-->VisualizationImplementation: baseDirectory
```

```
deactivate FileManager

VisualizationImplementation->FileManager: newDirectory = fm.createDirectory(directory =
workingDirectory, subDirectory = outputDirectory)
activate FileManager

alt subDirectory.startsWith("/")
FileManager->FileManager: newDirectory = subDirectory
else
FileManager->FileManager: newDirectory = directory + File.separator + subDirectory
end

FileManager->FileManager: path = Paths.get(newDirectory)
FileManager->FileManager: Files.createDirectories(path)

opt verbose
FileManager->FileManager: System.out.println("Directory " + newDirectory + " is created!")
end

FileManager-->VisualizationImplementation: newDirectory
deactivate FileManager

alt inputFileName.contains("/")
  VisualizationImplementation->VisualizationImplementation: index =
inputFileName.lastIndexOf("/") + 1
  VisualizationImplementation->VisualizationImplementation: fileNameTemplate = newDirectory
+ File.separator + inputFileName.substring(index)
else
  VisualizationImplementation->VisualizationImplementation: fileNameTemplate = newDirectory
+ File.separator + inputFileName;
end

VisualizationImplementation-->VisualizationImplementation: fileNameTemplate
deactivate VisualizationImplementation

VisualizationImplementation->ReliabilityVisualization: obj = new ReliabilityVisualization(warp)
activate ReliabilityVisualization

ReliabilityVisualization->ReliabilityAnalysis: reliabilityData = getReliabilities(warp.toProgram())
activate ReliabilityAnalysis
ReliabilityAnalysis-->ReliabilityVisualization: Reliability data
deactivate ReliabilityAnalysis

ReliabilityVisualization->WorkLoad: flows = getFlowNamesInPriorityOrder()
```

```
activate WorkLoad
WorkLoad-->ReliabilityVisualization: Flow names in priority order
deactivate WorkLoad

ReliabilityVisualization->WorkLoad: nodes = getNodesInFlow(flow)
activate WorkLoad
WorkLoad-->ReliabilityVisualization: Nodes in flow
deactivate WorkLoad

ReliabilityVisualization->FileManager: saveVisualization(fileContent)
activate FileManager
FileManager-->ReliabilityVisualization: File saved
deactivate FileManager

ReliabilityVisualization->GuiVisualization: displayVisualization()
activate GuiVisualization
GuiVisualization-->ReliabilityVisualization: GUI displayed
deactivate GuiVisualization

ReliabilityVisualization->FileManager: fileContent = createHeader()
activate FileManager
FileManager-->ReliabilityVisualization: Header created
deactivate FileManager

ReliabilityVisualization->FileManager: columnHeader = createColumnHeader()
activate FileManager
FileManager-->ReliabilityVisualization: Column header created
deactivate FileManager

ReliabilityVisualization->FileManager: visualizationData = createVisualizationData()
activate FileManager
FileManager-->ReliabilityVisualization: Visualization data created
deactivate FileManager

ReliabilityVisualization-->VisualizationImplementation: ReliabilityVisualization instance
deactivate ReliabilityVisualization

VisualizationImplementation->VisualizationImplementation: createVisualization(obj)
activate VisualizationImplementation

VisualizationImplementation->ReliabilityVisualization: visualization = obj.visualization()
activate ReliabilityVisualization

ReliabilityVisualization->VisualizationObject: visualization = super.visualization()
```

```
activate VisualizationObject

VisualizationObject->Description: content = new Description()
activate Description
Description->Description: super()
Description->VisualizationObject: Description instance
deactivate Description

VisualizationObject->ReliabilityVisualization: data = createVisualizationData()
activate ReliabilityVisualization
note right of ReliabilityVisualization: TODO
ReliabilityVisualization-->VisualizationObject: visualizationData
deactivate ReliabilityVisualization

VisualizationObject->ReliabilityVisualization: columnHeader = createColumnHeader()
activate ReliabilityVisualization
note right of ReliabilityVisualization: TODO
ReliabilityVisualization-->VisualizationObject: String[] instance
deactivate ReliabilityVisualization

VisualizationObject->VisualizationObject: nodeString = String.join("\t", columnHeader) + "\\n"
VisualizationObject->VisualizationObject: content.add(nodeString)

loop rowIndex = 0; rowIndex < data.length; rowIndex++
VisualizationObject->VisualizationObject: row = data[rowIndex]
VisualizationObject->VisualizationObject: rowString = String.join("\t", row) + "\\n"
VisualizationObject->VisualizationObject: content.add(rowString)
end

VisualizationObject-->ReliabilityVisualization: content
deactivate VisualizationObject

ReliabilityVisualization-->VisualizationImplementation: visualization
deactivate ReliabilityVisualization

VisualizationImplementation->ReliabilityVisualization: fileContent = obj.fileVisualization()
activate ReliabilityVisualization

ReliabilityVisualization->VisualizationObject: fileContent = super.fileVisualization()
activate VisualizationObject

VisualizationObject->ReliabilityVisualization: fileContent = createHeader()
activate ReliabilityVisualization
ReliabilityVisualization->Description: header = new Description()
```

```
activate Description
Description->Description: super()
Description->ReliabilityVisualization: Description instance
deactivate Description
note right of ReliabilityVisualization: TODO
ReliabilityVisualization-->VisualizationObject: header
deactivate ReliabilityVisualization

VisualizationObject->VisualizationObject: visualization = visualization()
activate VisualizationObject

VisualizationObject->Description: content = new Description()
activate Description
Description->Description: super()
Description->VisualizationObject: Description instance
deactivate Description

VisualizationObject->ReliabilityVisualization: data = createVisualizationData()
activate ReliabilityVisualization
note right of ReliabilityVisualization: TODO
ReliabilityVisualization-->VisualizationObject: visualizationData
deactivate ReliabilityVisualization

VisualizationObject->ReliabilityVisualization: columnHeader = createColumnHeader()
activate ReliabilityVisualization
note right of ReliabilityVisualization: TODO
ReliabilityVisualization-->VisualizationObject: String[] instance
deactivate ReliabilityVisualization

VisualizationObject->VisualizationObject: nodeString = String.join("\t", columnHeader) + "\\n"
VisualizationObject->VisualizationObject: content.add(nodeString)

loop rowIndex = 0; rowIndex < data.length; rowIndex++
VisualizationObject->VisualizationObject: row = data[rowIndex]
VisualizationObject->VisualizationObject: rowString = String.join("\t", row) + "\\n"
VisualizationObject->VisualizationObject: content.add(rowString)
end

VisualizationObject-->VisualizationObject: content
deactivate VisualizationObject

VisualizationObject->VisualizationObject: fileContent.addAll(visualization)

VisualizationObject->ReliabilityVisualization: footer = createFooter()
```

```
activate ReliabilityVisualization
ReliabilityVisualization->Description: footer = new Description()
activate Description
Description->Description: super()
Description->ReliabilityVisualization: Description instance
deactivate Description
note right of ReliabilityVisualization: TODO
ReliabilityVisualization-->VisualizationObject: footer
deactivate ReliabilityVisualization

VisualizationObject->VisualizationObject: fileContent.addAll(footer)

VisualizationObject-->ReliabilityVisualization: fileContent
deactivate VisualizationObject

ReliabilityVisualization-->VisualizationImplementation: fileContent
deactivate ReliabilityVisualization

VisualizationImplementation->ReliabilityVisualization: fileName =
obj.createFile(fileNameTemplate)
activate ReliabilityVisualization

ReliabilityVisualization->VisualizationObject: fileString = super.createFile(fileNameTemplate,
nameExtension, suffix)
activate VisualizationObject

VisualizationObject->FileManager: fileString = fm.createFile(file = fileNameTemplate,
nameExtension, suffix)
activate FileManager

FileManager->FileManager: suffixIndex = file.lastIndexOf('.')
FileManager->FileManager: fileString = file

opt suffixIndex > 0
FileManager->FileManager: fileString = file.substring(0, suffixIndex)
end

FileManager->FileManager: fileString = fileString + nameExtension + suffix

opt verbose
FileManager->FileManager: System.out.println("File " + fileString + " is created!")
end

FileManager-->VisualizationObject:
```

```
deactivate FileManager

VisualizationObject-->ReliabilityVisualization: fileString
deactivate VisualizationObject

ReliabilityVisualization-->VisualizationImplementation: fileString
deactivate ReliabilityVisualization

VisualizationImplementation->VisualizationImplementation: visualizationObject = obj
deactivate VisualizationImplementation

VisualizationImplementation-->VisualizationFactory: VisualizationImplementation instance
deactivate VisualizationImplementation

VisualizationFactory-->Warp: viz
deactivate VisualizationFactory

opt verbose
Warp->VisualizationImplementation: vizString = viz.toString()
activate VisualizationImplementation

VisualizationImplementation->Description: vizString = visualization.toString()
activate Description

Description->Description: sb = new StringBuffer()
loop row : this
Description->Description: sb.append(row)
end

Description-->VisualizationImplementation: sb.toString()
deactivate Description

VisualizationImplementation-->Warp: vizString
deactivate VisualizationImplementation

Warp->Warp: System.out.println(vizString)
end

Warp->VisualizationImplementation: viz.toFile()
activate VisualizationImplementation

VisualizationImplementation->Description: fileString = fileContent.toString()
activate Description
```

```
Description->Description: sb = new StringBuffer()
loop row : this
Description->Description: sb.append(row)
end

Description-->VisualizationImplementation: sb.toString()
deactivate Description

VisualizationImplementation->FileManager: fm.writeFile(file = fileName, fileContents = fileString)
activate FileManager
FileManager->FileManager: fileName = Path.of(file)
FileManager->FileManager: Files.writeString(fileName, fileContents)
deactivate FileManager

deactivate VisualizationImplementation

Warp->VisualizationImplementation: viz.toDisplay()
activate VisualizationImplementation

VisualizationImplementation->ReliabilityAnalysis: window =
visualizationObject.displayVisualization()
activate ReliabilityAnalysis
note right of ReliabilityAnalysis: TODO
ReliabilityAnalysis-->VisualizationImplementation: GuiVisualization instance
deactivate ReliabilityAnalysis

VisualizationImplementation->GuiVisualization: window.setVisible()
activate GuiVisualization
GuiVisualization->GuiVisualization: frame.setVisible(true)
deactivate GuiVisualization

deactivate VisualizationImplementation

deactivate Warp

deactivate Warp
```