RESUMASTER

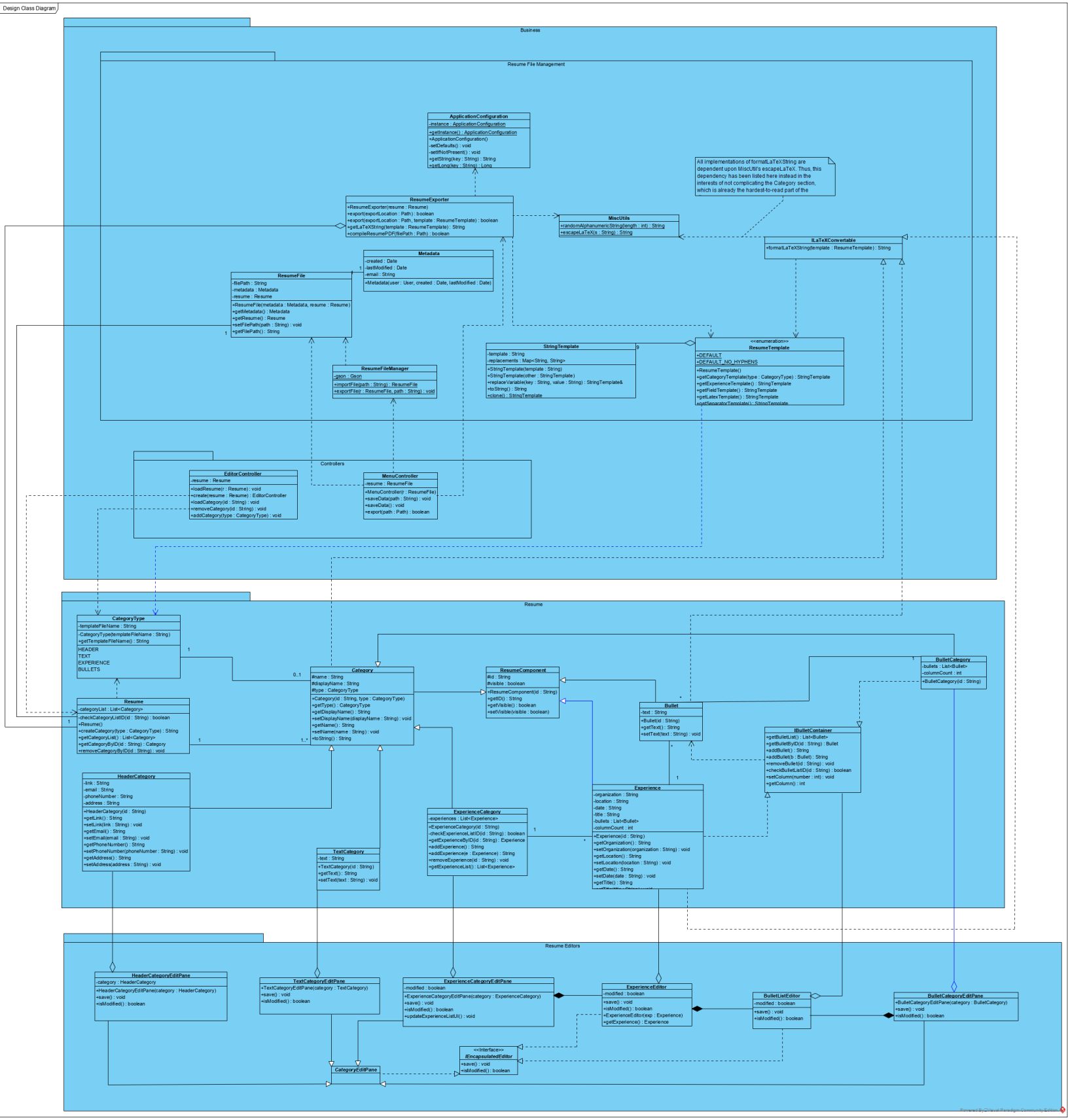
Iteration 2 Artifacts

Ashley Lu Couch
Brooklynn Stone
Jacob Curtis
Matthew McCaskill
Micah Schiewe

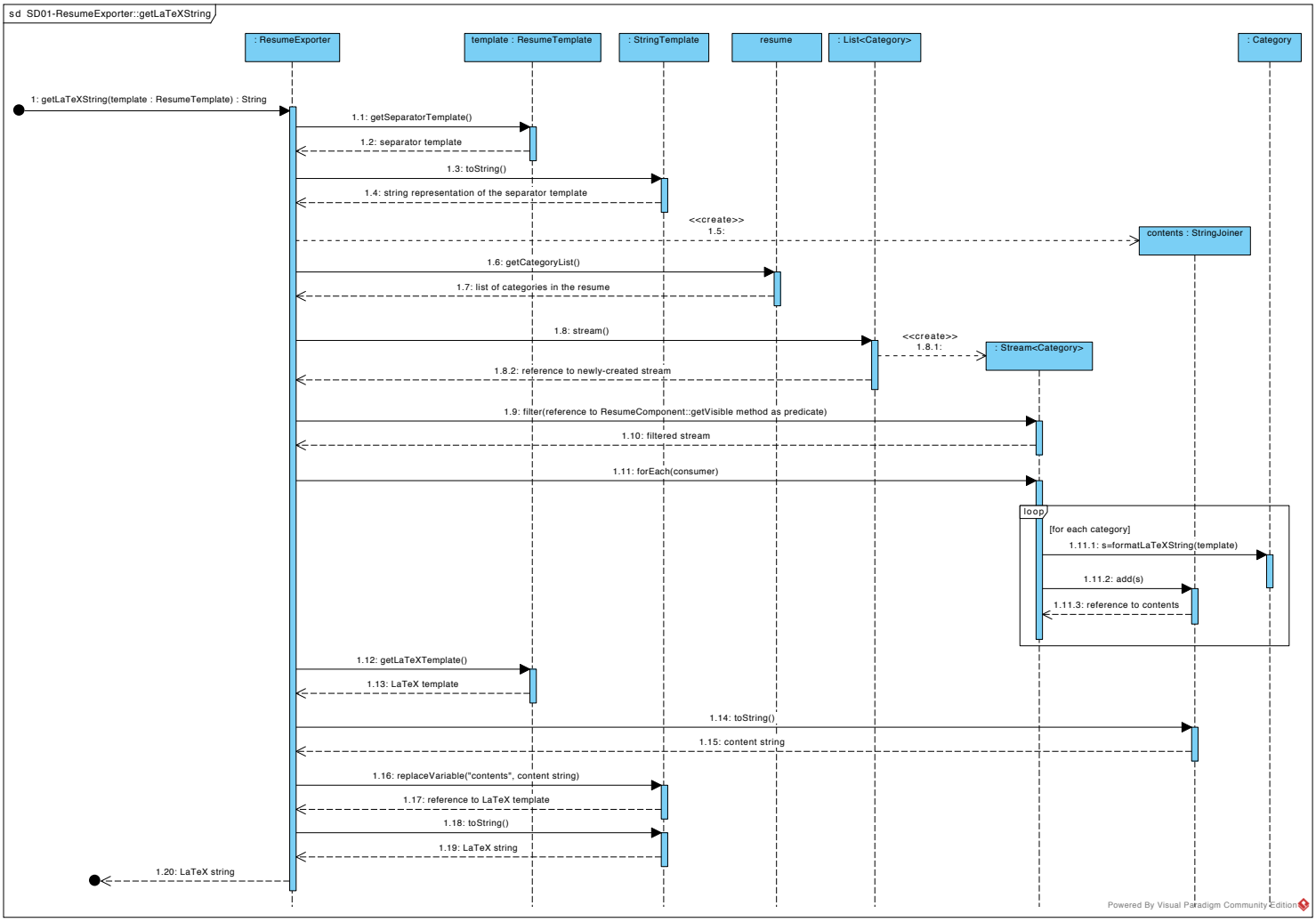
DEMO of User Interface

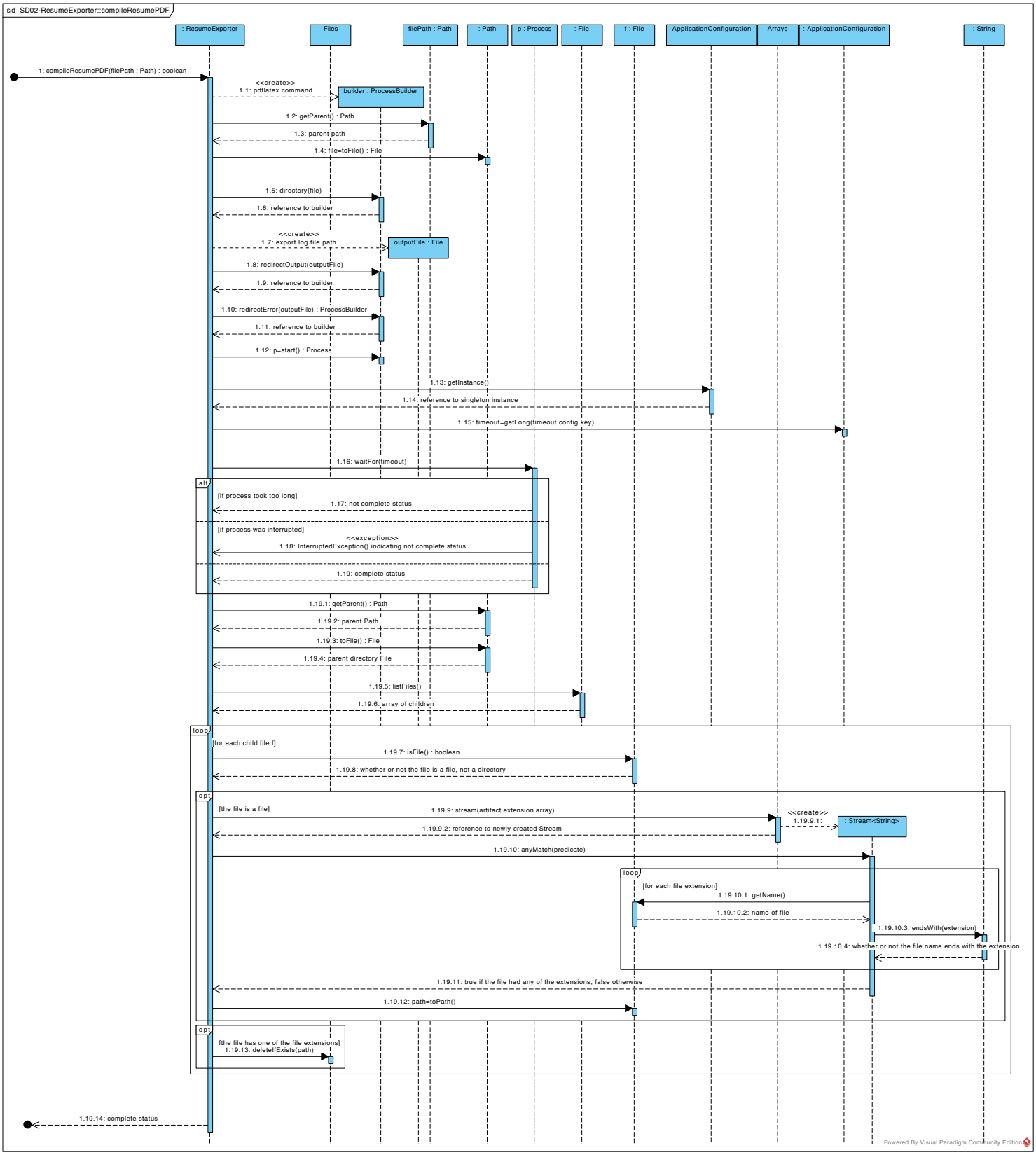
https://drive.google.com/file/d/1d_mtu4zzAzYmOVSgiRAbaECZWBY8fDp0/view?usp=sharing

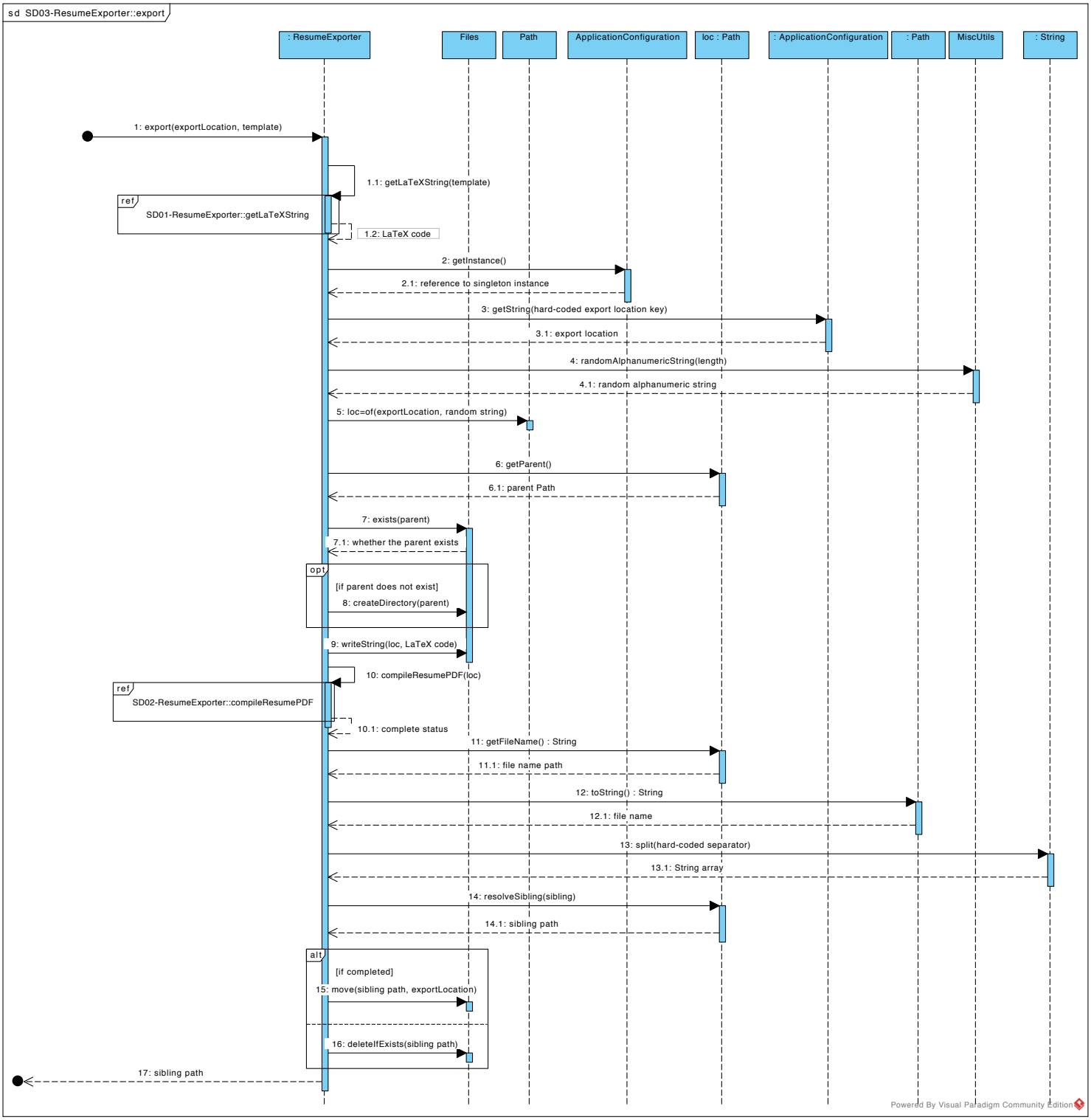
Design Class diagram

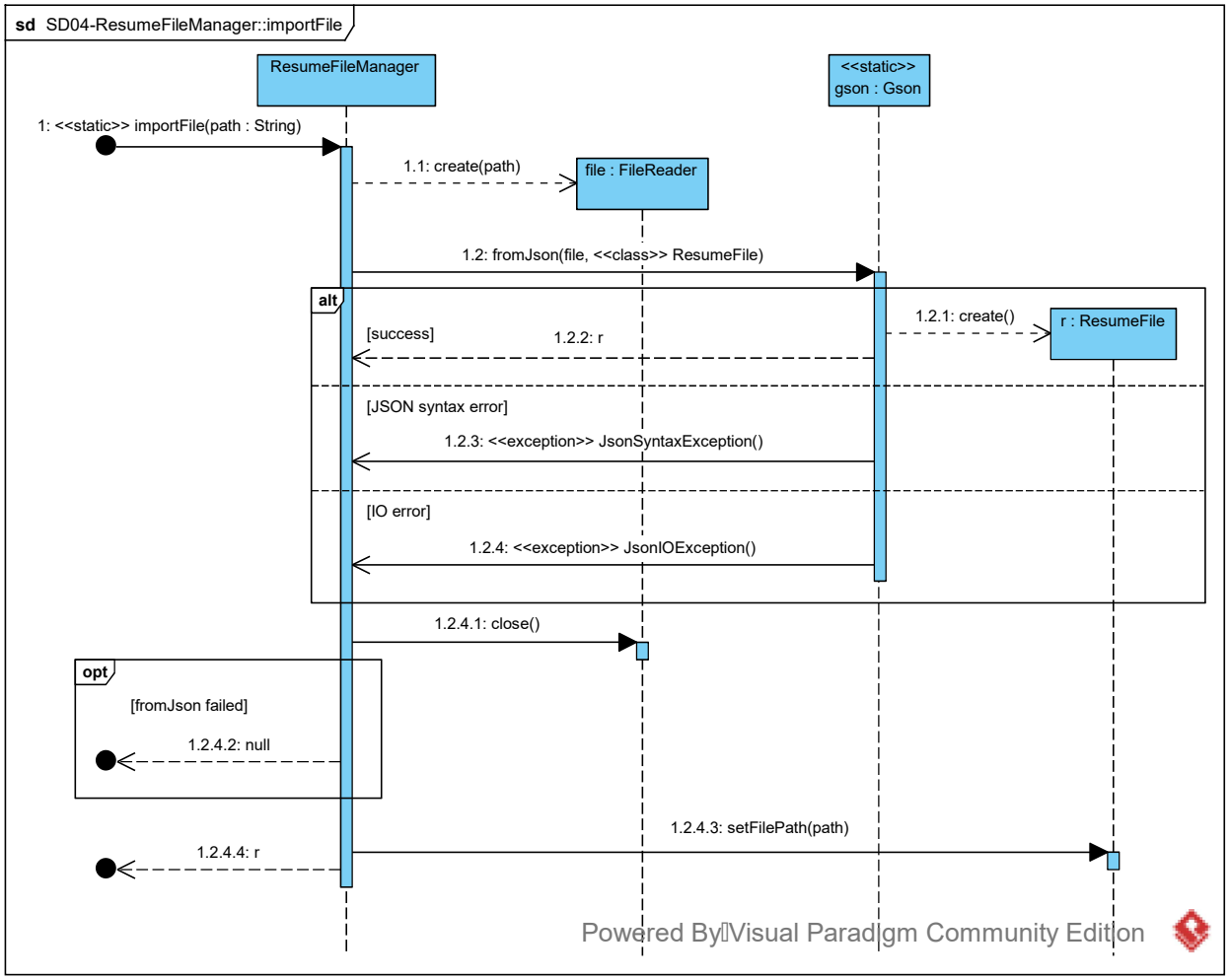

Design Class Diagram
Business
Resume File Management
ApplicationConfiguration
-instance : ApplicationConfiguration
+getInstance() : ApplicationConfiguration
+ApplicationConfiguration()
-setDefaults() : void
-setIfNotPresent() : void
+getString(key : String) : String
+getLong(key : String) : Long
All implementations of formatLaTeXString are dependent upon MiscUtil's escapeLaTeX. Thus, this dependency has been listed here instead in the interests of not complicating the Category section, which is already the hardest-to-read part of the
ResumeExporter
+ResumeExporter(resume : Resume)
+export(exportLocation : Path) : boolean
+export(exportLocation : Path, template : ResumeTemplate) : boolean
+getLaTeXString(template : ResumeTemplate) : String
+compileResumePDF(filePath : Path) : boolean
MiscUtils
+randomAlphanumericString(length : int) : String
+escapeLaTeX(s : String) : String
ILaTeXConvertable
+formatLaTeXString(template : ResumeTemplate) : String
Metadata
-created : Date
-lastModified : Date
-email : String
+Metadata(user : User, created : Date, lastModified : Date)
ResumeFile
-filePath : String
-metadata : Metadata
-resume : Resume
+ResumeFile(metadata : Metadata, resume : Resume)
+getMetadata() : Metadata
+getResume() : Resume
+setFilePath(path : String) : void
+getFilePath() : String
StringTemplate
-template : String
-replacements : Map<String, String>
+StringTemplate(template : String)
+StringTemplate(other : StringTemplate)
+replaceVariable(key : String, value : String) : StringTemplate&
+toString() : String
+clone() : StringTemplate
<<enumeration>>
ResumeTemplate
+DEFAULT
+DEFAULT_NO_HYPHENS
+ResumeTemplate()
+getCategoryTemplate(type : CategoryType) : StringTemplate
+getExperienceTemplate() : StringTemplate
+getFieldTemplate() : StringTemplate
+getLatexTemplate() : StringTemplate
+getSeparatorTemplate() : StringTemplate
ResumeFileManager
-gson : Gson
+importFile(path : String) : ResumeFile
+exportFile(r : ResumeFile, path : String) : void
Controllers
EditorController
-resume : Resume
+loadResume(r : Resume) : void
+create(resume : Resume) : EditorController
+loadCategory(id : String) : void
+removeCategory(id : String) : void
+addCategory(type : CategoryType) : void
MenuController
-resume : ResumeFile
+MenuController(r : ResumeFile)
+saveData(path : String) : void
+saveData() : void
+export(path : Path) : boolean
Resume
CategoryType
-templateFileName : String
-CategoryType(templateFileName : String)
+getTemplateFileName() : String
HEADER
TEXT
EXPERIENCE
BULLETS
Category
#name : String
#displayName : String
#type : CategoryType
+Category(id : String, type : CategoryType)
+getType() : CategoryType
+getDisplayName() : String
+setDisplayName(displayName : String) : void
+getName() : String
+setName(name : String) : void
+toString() : String
ResumeComponent
#id : String
#visible : boolean
+ResumeComponent(id : String)
+getID() : String
+getVisible() : boolean
+setVisible(visible : boolean)
BulletCategory
-bullets : List<Bullet>
-columnCount : int
+BulletCategory(id : String)
Resume
-categoryList : List<Category>
-checkCategoryListID(id : String) : boolean
+Resume()
+createCategory(type : CategoryType) : String
+getCategoryList() : List<Category>
+getCategoryByID(id : String) : Category
+removeCategoryByID(id : String) : void
Bullet
-text : String
+Bullet(id : String)
+getText() : String
+setText(text : String) : void
IBulletContainer
+getBulletList() : List<Bullet>
+getBulletByID(id : String) : Bullet
+addBullet() : String
+addBullet(b : Bullet) : String
+removeBullet(id : String) : void
+checkBulletListID(id : String) : boolean
+setColumn(number : int) : void
+getColumn() : int
HeaderCategory
-link : String
-email : String
-phoneNumber : String
-address : String
+HeaderCategory(id : String)
+getLink() : String
+setLink(link : String) : void
+getEmail() : String
+setEmail(email : String) : void
+getPhoneNumber() : String
+setPhoneNumber(phoneNumber : String) : void
+getAddress() : String
+setAddress(address : String) : void
ExperienceCategory
-experiences : List<Experience>
+ExperienceCategory(id : String)
-checkExperienceListID(id : String) : boolean
+getExperienceByID(id : String) : Experience
+addExperience() : String
+addExperience(e : Experience) : String
+removeExperience(id : String) : void
+getExperienceList() : List<Experience>
Experience
-organization : String
-location : String
-date : String
-title : String
-bullets : List<Bullet>
-columnCount : int
+Experience(id : String)
+getOrganization() : String
+setOrganization(organization : String) : void
+getLocation() : String
+setLocation(location : String) : void
+getDate() : String
+setDate(date : String) : void
+getTitle() : String
TextCategory
-text : String
+TextCategory(id : String)
+getText() : String
+setText(text : String) : void
Resume Editors
HeaderCategoryEditPane
-category : HeaderCategory
+HeaderCategoryEditPane(category : HeaderCategory)
+save() : void
+isModified() : boolean
TextCategoryEditPane
+TextCategoryEditPane(category : TextCategory)
+save() : void
+isModified() : boolean
ExperienceCategoryEditPane
-modified : boolean
+ExperienceCategoryEditPane(category : ExperienceCategory)
+save() : void
+isModified() : boolean
+updateExperienceListUI() : void
ExperienceEditor
-modified : boolean
+save() : void
+isModified() : boolean
+ExperienceEditor(exp : Experience)
+getExperience() : Experience
BulletListEditor
-modified : boolean
+save() : void
+isModified() : boolean
BulletCategoryEditPane
+BulletCategoryEditPane(category : BulletCategory)
+save() : void
+isModified() : boolean
<<Interface>>
IEncapsulatedEditor
+save() : void
+isModified() : boolean
CategoryEditPane
Powered By Visual Paradigm Community Edition

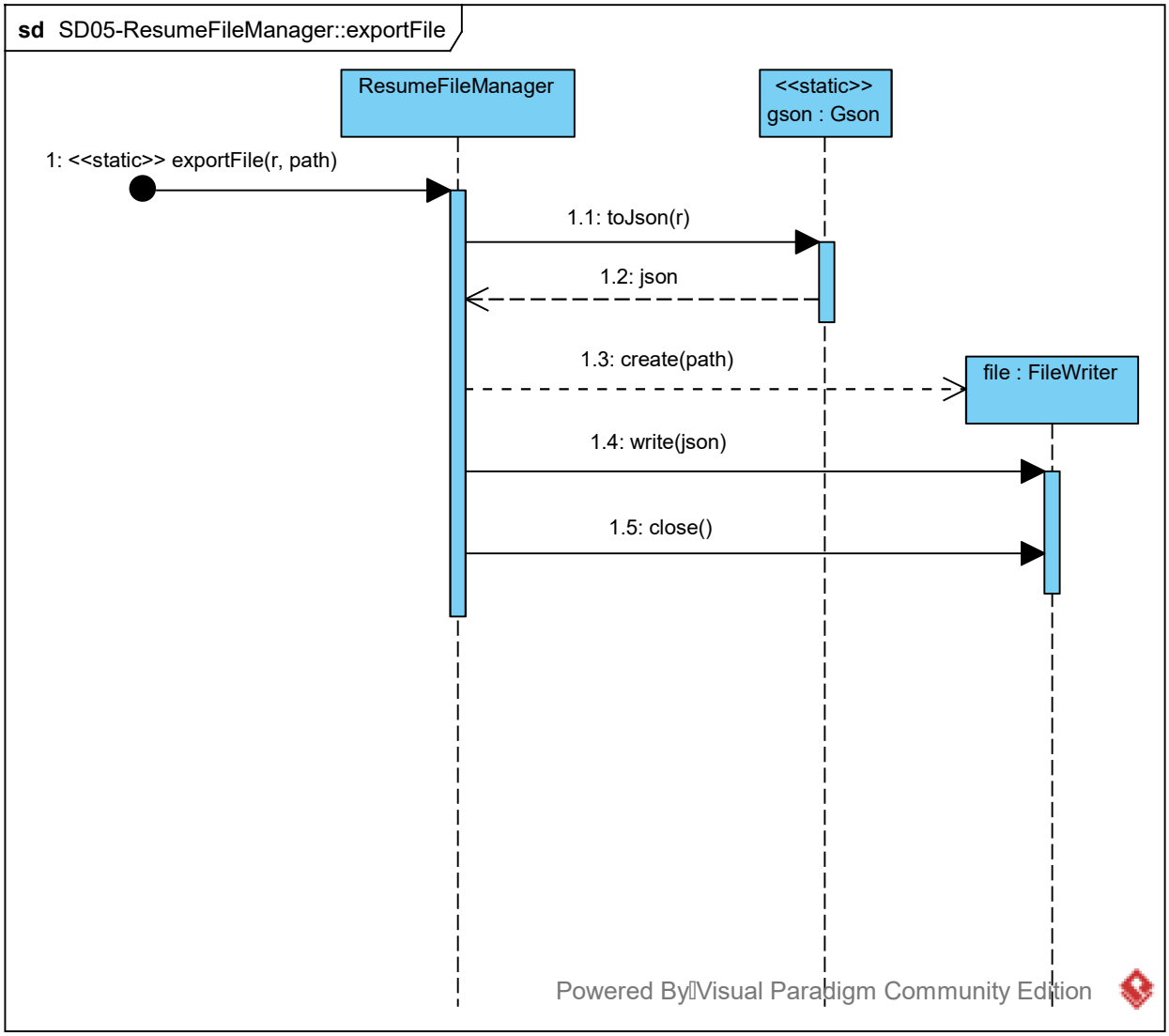
Sequence Diagrams

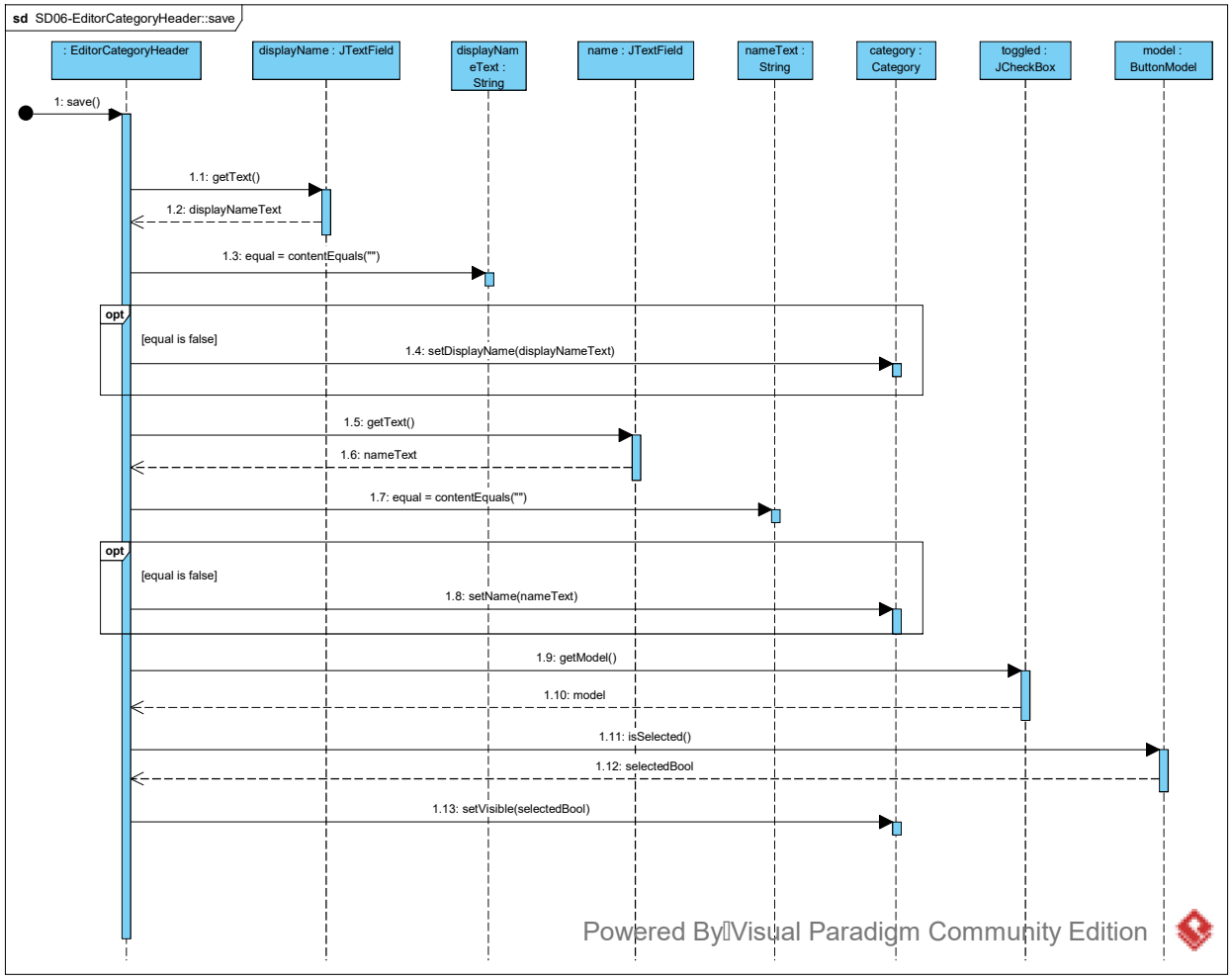

sd SD01-ResumeExporter::getLaTeXString
: ResumeExporter
template : ResumeTemplate
: StringTemplate
resume
: List<Category>
: Category
1: getLaTeXString(template : ResumeTemplate) : String
1.1: getSeparatorTemplate()
1.2: separator template
1.3: toString()
1.4: string representation of the separator template
<<create>>
1.5:
contents : StringJoiner
1.6: getCategoryList()
1.7: list of categories in the resume
1.8: stream()
<<create>>
1.8.1:
: Stream<Category>
1.8.2: reference to newly-created stream
1.9: filter(reference to ResumeComponent::getVisible method as predicate)
1.10: filtered stream
1.11: forEach(consumer)
loop
[for each category]
1.11.1: s=formatLaTeXString(template)
1.11.2: add(s)
1.11.3: reference to contents
1.12: getLaTeXTemplate()
1.13: LaTeX template
1.14: toString()
1.15: content string
1.16: replaceVariable("contents", content string)
1.17: reference to LaTeX template
1.18: toString()
1.19: LaTeX string
1.20: LaTeX string
Powered By Visual Paradigm Community Edition

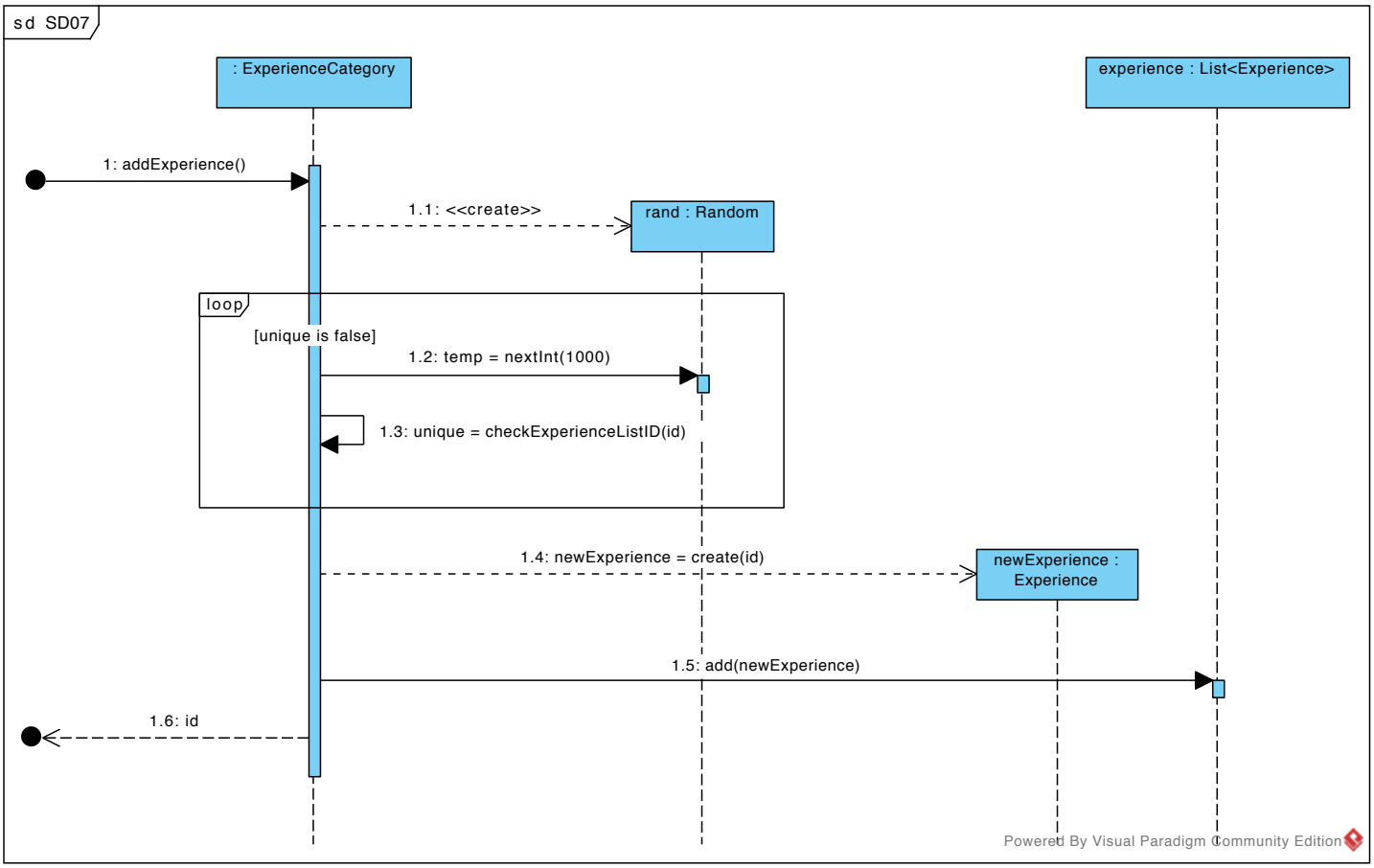

sd SD02-ResumeExporter::compileResumePDF
: ResumeExporter
Files
filePath : Path
: Path
p : Process
: File
f : File
ApplicationConfiguration
Arrays
: ApplicationConfiguration
: String
1: compileResumePDF(filePath : Path) : boolean
<<create>>
1.1: pdflatex command
builder : ProcessBuilder
1.2: getParent() : Path
1.3: parent path
1.4: file=toFile() : File
1.5: directory(file)
1.6: reference to builder
<<create>>
1.7: export log file path
outputFile : File
1.8: redirectOutput(outputFile)
1.9: reference to builder
1.10: redirectError(outputFile) : ProcessBuilder
1.11: reference to builder
1.12: p=start() : Process
1.13: getInstance()
1.14: reference to singleton instance
1.15: timeout=getLong(timeout config key)
1.16: waitFor(timeout)
alt
[if process took too long]
1.17: not complete status
[if process was interrupted]
<<exception>>
1.18: InterruptedException() indicating not complete status
1.19: complete status
1.19.1: getParent() : Path
1.19.2: parent Path
1.19.3: toFile() : File
1.19.4: parent directory File
1.19.5: listFiles()
1.19.6: array of children
loop
[for each child file f]
1.19.7: isFile() : boolean
1.19.8: whether or not the file is a file, not a directory
opt
[the file is a file]
1.19.9: stream(artifact extension array)
<<create>>
1.19.9.1:
: Stream<String>
1.19.9.2: reference to newly-created Stream
1.19.10: anyMatch(predicate)
loop
[for each file extension]
1.19.10.1: getName()
1.19.10.2: name of file
1.19.10.3: endsWith(extension)
1.19.10.4: whether or not the file name ends with the extension
1.19.11: true if the file had any of the extensions, false otherwise
1.19.12: path=toPath()
opt
[the file has one of the file extensions]
1.19.13: deleteIfExists(path)
1.19.14: complete status
Powered By Visual Paradigm Community Edition

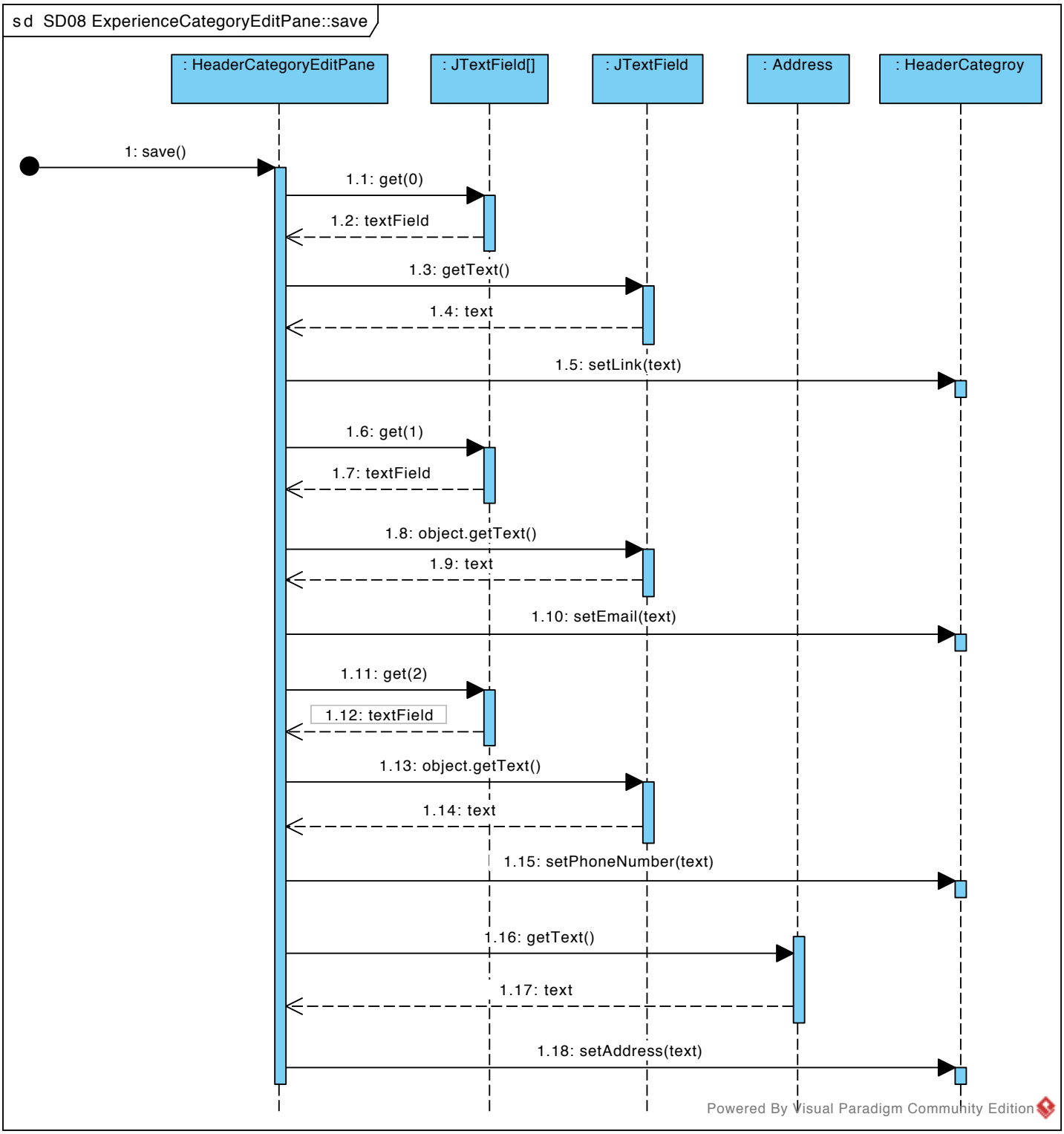

sd SD03-ResumeExporter::export
: ResumeExporter
Files
Path
ApplicationConfiguration
loc : Path
: ApplicationConfiguration
: Path
MiscUtils
: String
1: export(exportLocation, template)
1.1: getLaTeXString(template)
ref
SD01-ResumeExporter::getLaTeXString
1.2: LaTeX code
2: getInstance()
2.1: reference to singleton instance
3: getString(hard-coded export location key)
3.1: export location
4: randomAlphanumericString(length)
4.1: random alphanumeric string
5: loc=of(exportLocation, random string)
6: getParent()
6.1: parent Path
7: exists(parent)
7.1: whether the parent exists
opt
[if parent does not exist]
8: createDirectory(parent)
9: writeString(loc, LaTeX code)
10: compileResumePDF(loc)
ref
SD02-ResumeExporter::compileResumePDF
10.1: complete status
11: getFileName() : String
11.1: file name path
12: toString() : String
12.1: file name
13: split(hard-coded separator)
13.1: String array
14: resolveSibling(sibling)
14.1: sibling path
alt
[if completed]
15: move(sibling path, exportLocation)
16: deleteIfExists(sibling path)
17: sibling path
Powered By Visual Paradigm Community Edition

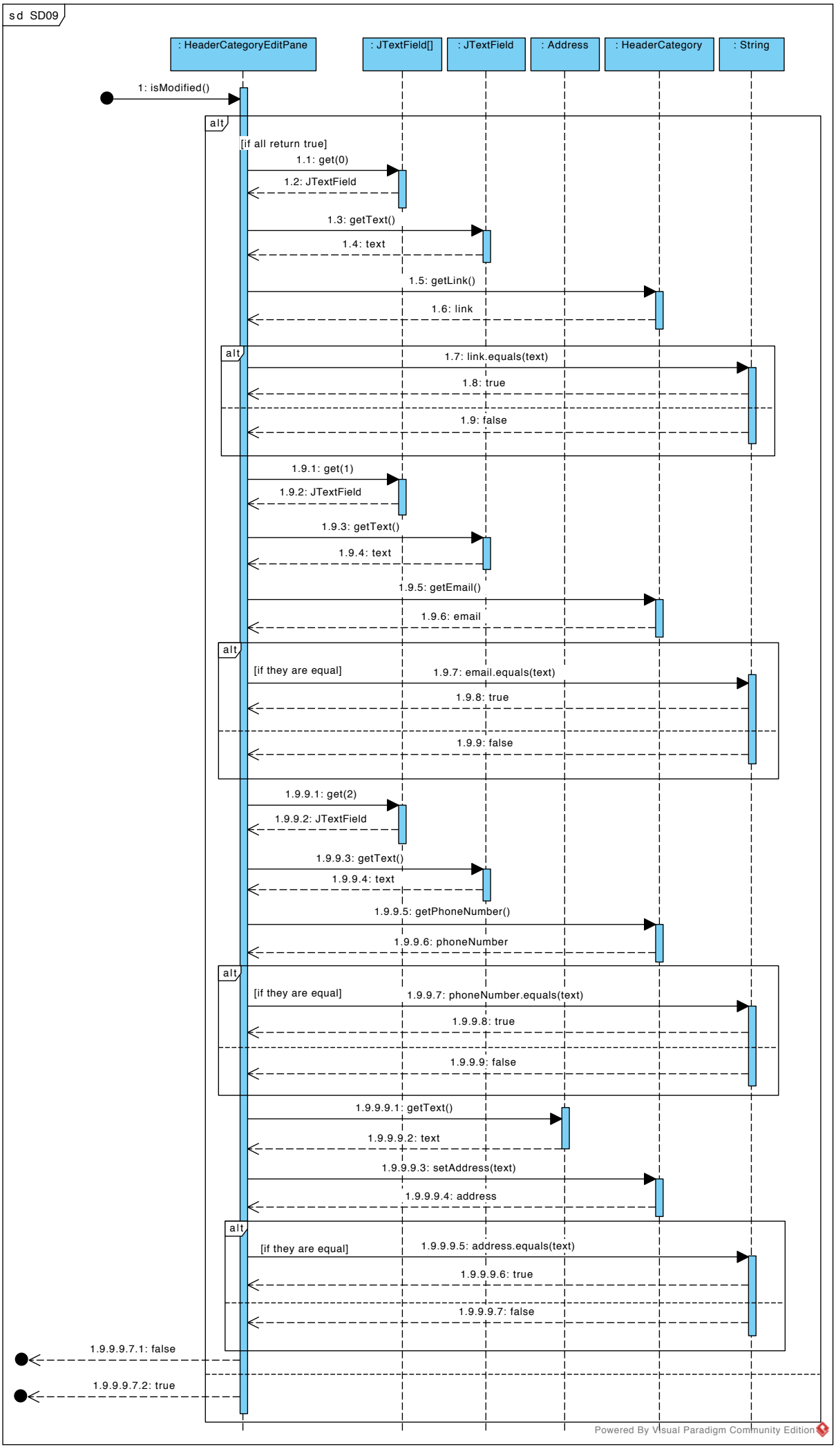

sd SD04-ResumeFileManager::importFile
ResumeFileManager
<<static>>
gson : Gson
1: <<static>> importFile(path : String)
1.1: create(path)
file : FileReader
1.2: fromJson(file, <<class>> ResumeFile)
alt
1.2.1: create()
r : ResumeFile
[success]
1.2.2: r
[JSON syntax error]
1.2.3: <<exception>> JsonSyntaxException()
[IO error]
1.2.4: <<exception>> JsonIOException()
1.2.4.1: close()
opt
[fromJson failed]
1.2.4.2: null
1.2.4.3: setFilePath(path)
1.2.4.4: r
Powered By Visual Paradigm Community Edition

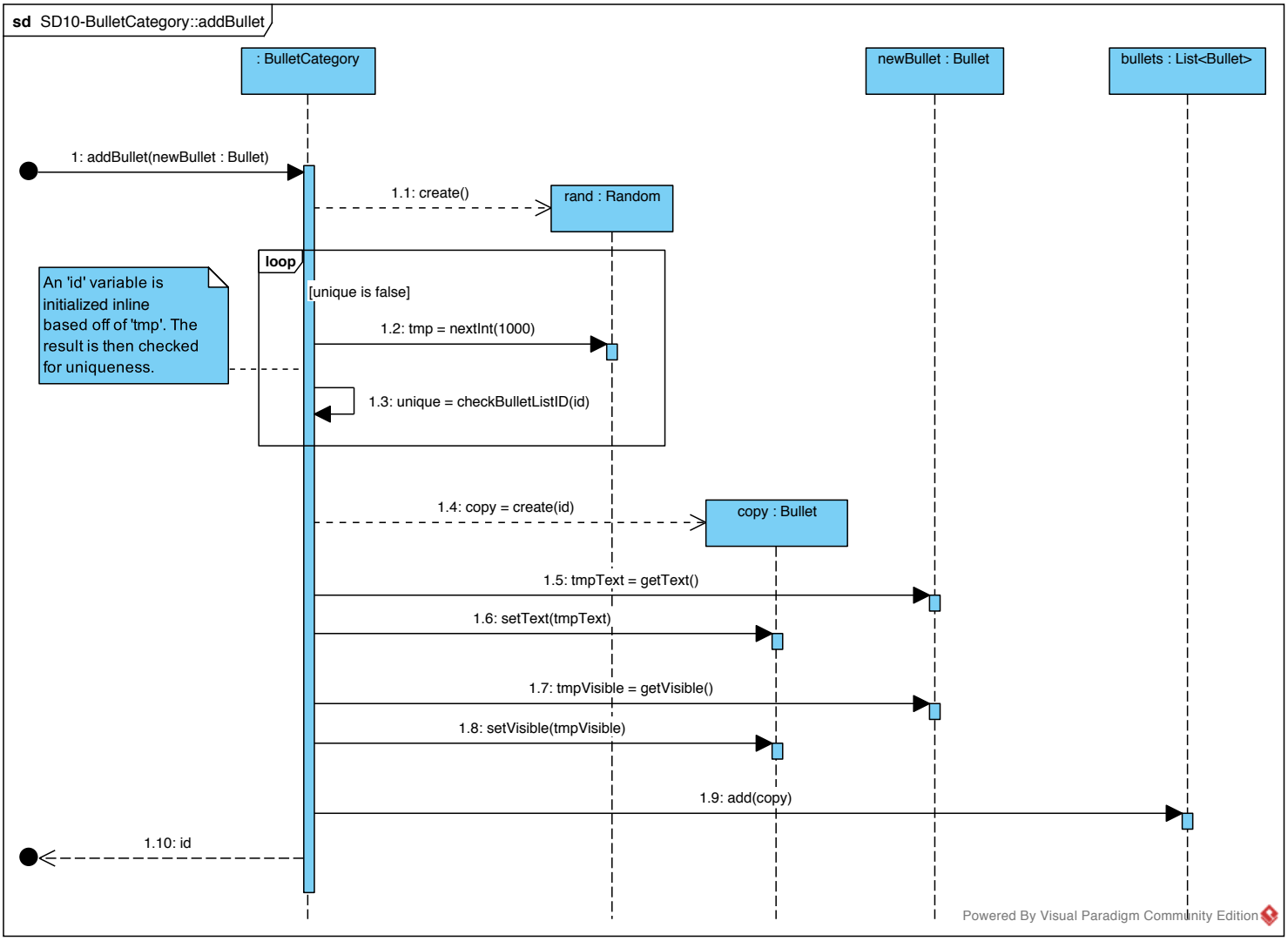

sd SD05-ResumeFileManager::exportFile
ResumeFileManager
<<static>>
gson : Gson
1: <<static>> exportFile(r, path)
1.1: toJson(r)
1.2: json
1.3: create(path)
file : FileWriter
1.4: write(json)
1.5: close()
Powered By Visual Paradigm Community Edition

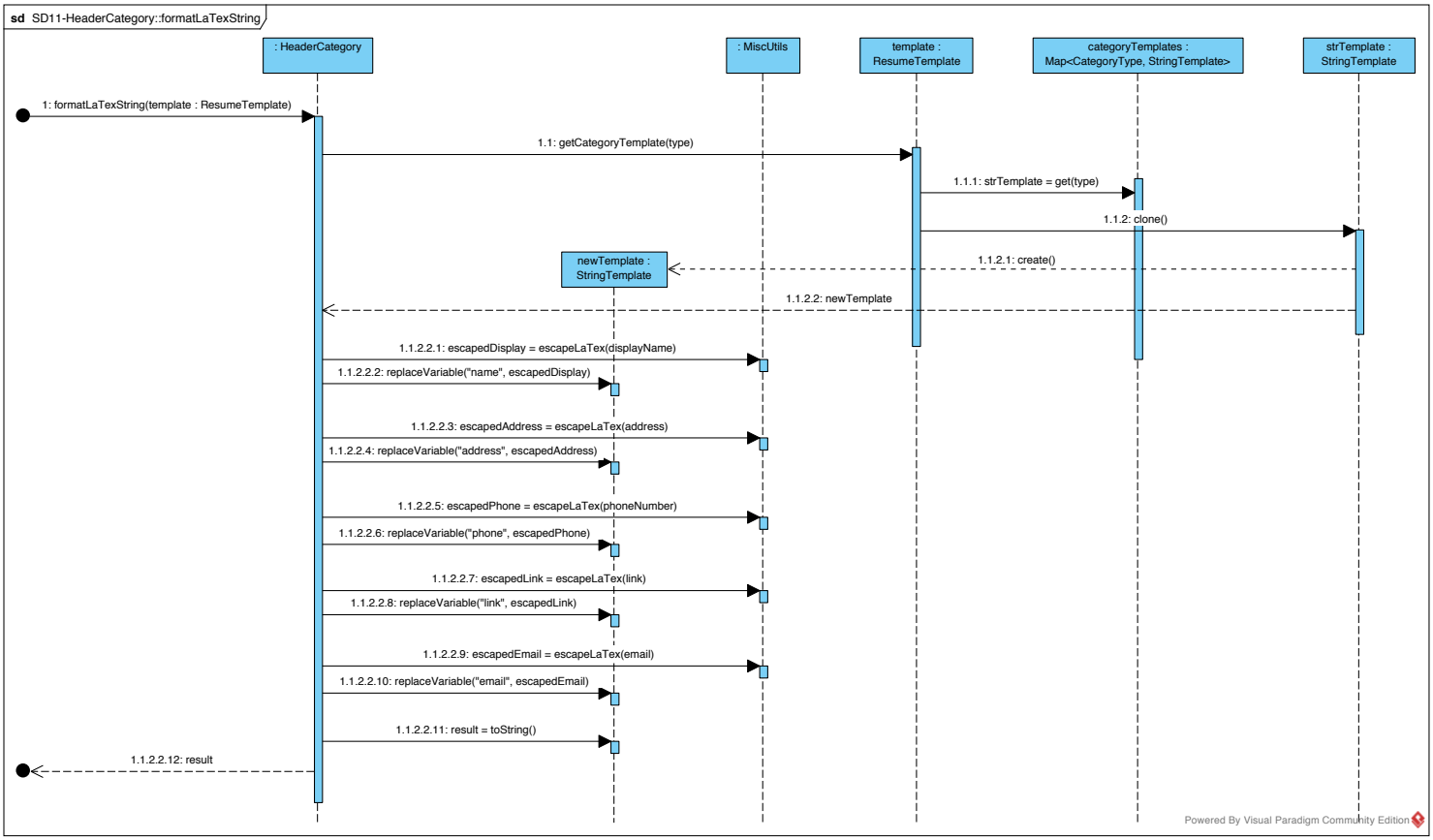

sd SD06-EditorCategoryHeader::save
: EditorCategoryHeader
displayName : JTextField
displayNameText : String
name : JTextField
nameText : String
category : Category
toggled : JCheckBox
model : ButtonModel
1: save()
1.1: getText()
1.2: displayNameText
1.3: equal = contentEquals("")
opt
[equal is false]
1.4: setDisplayName(displayNameText)
1.5: getText()
1.6: nameText
1.7: equal = contentEquals("")
opt
[equal is false]
1.8: setName(nameText)
1.9: getModel()
1.10: model
1.11: isSelected()
1.12: selectedBool
1.13: setVisible(selectedBool)
Powered By Visual Paradigm Community Edition

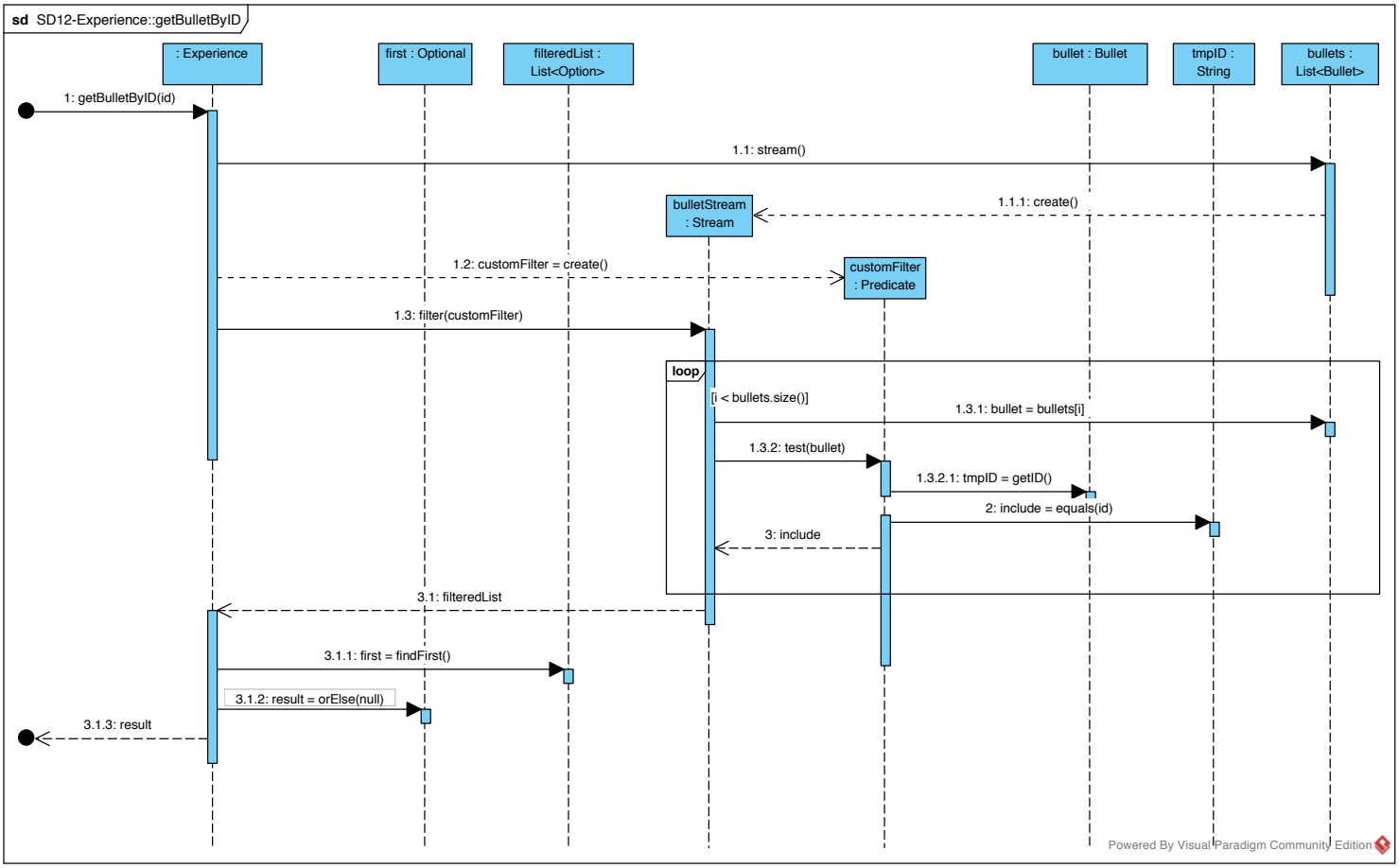

sd SD07
: ExperienceCategory
experience : List<Experience>
1: addExperience()
1.1: <<create>>
rand : Random
loop
[unique is false]
1.2: temp = nextInt(1000)
1.3: unique = checkExperienceListID(id)
1.4: newExperience = create(id)
newExperience : Experience
1.5: add(newExperience)
1.6: id
Powered By Visual Paradigm Community Edition

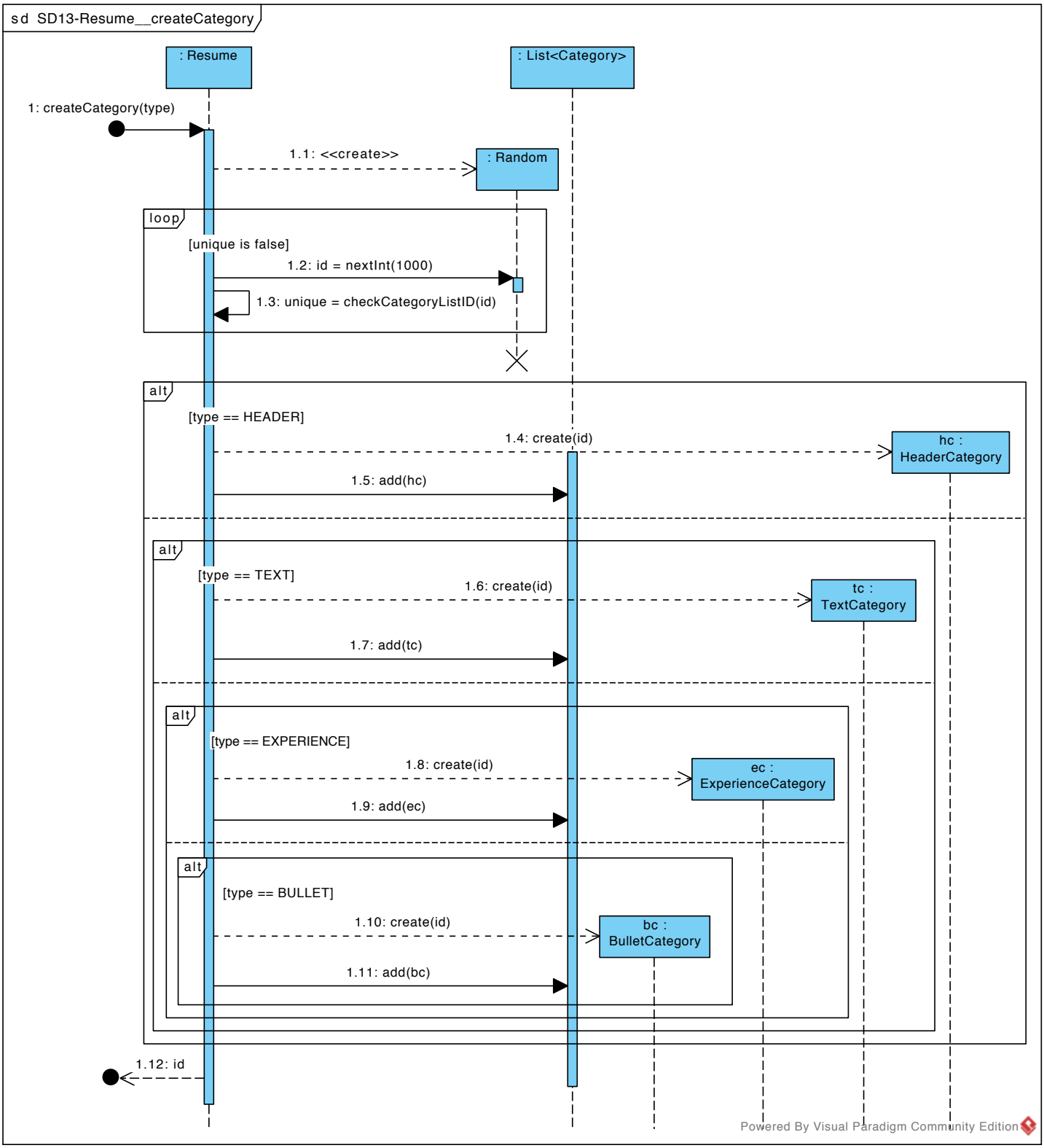

sd SD08 ExperienceCategoryEditPane::save
: HeaderCategoryEditPane
: JTextField[]
: JTextField
: Address
: HeaderCategroy
1: save()
1.1: get(0)
1.2: textField
1.3: getText()
1.4: text
1.5: setLink(text)
1.6: get(1)
1.7: textField
1.8: object.getText()
1.9: text
1.10: setEmail(text)
1.11: get(2)
1.12: textField
1.13: object.getText()
1.14: text
1.15: setPhoneNumber(text)
1.16: getText()
1.17: text
1.18: setAddress(text)
Powered By Visual Paradigm Community Edition

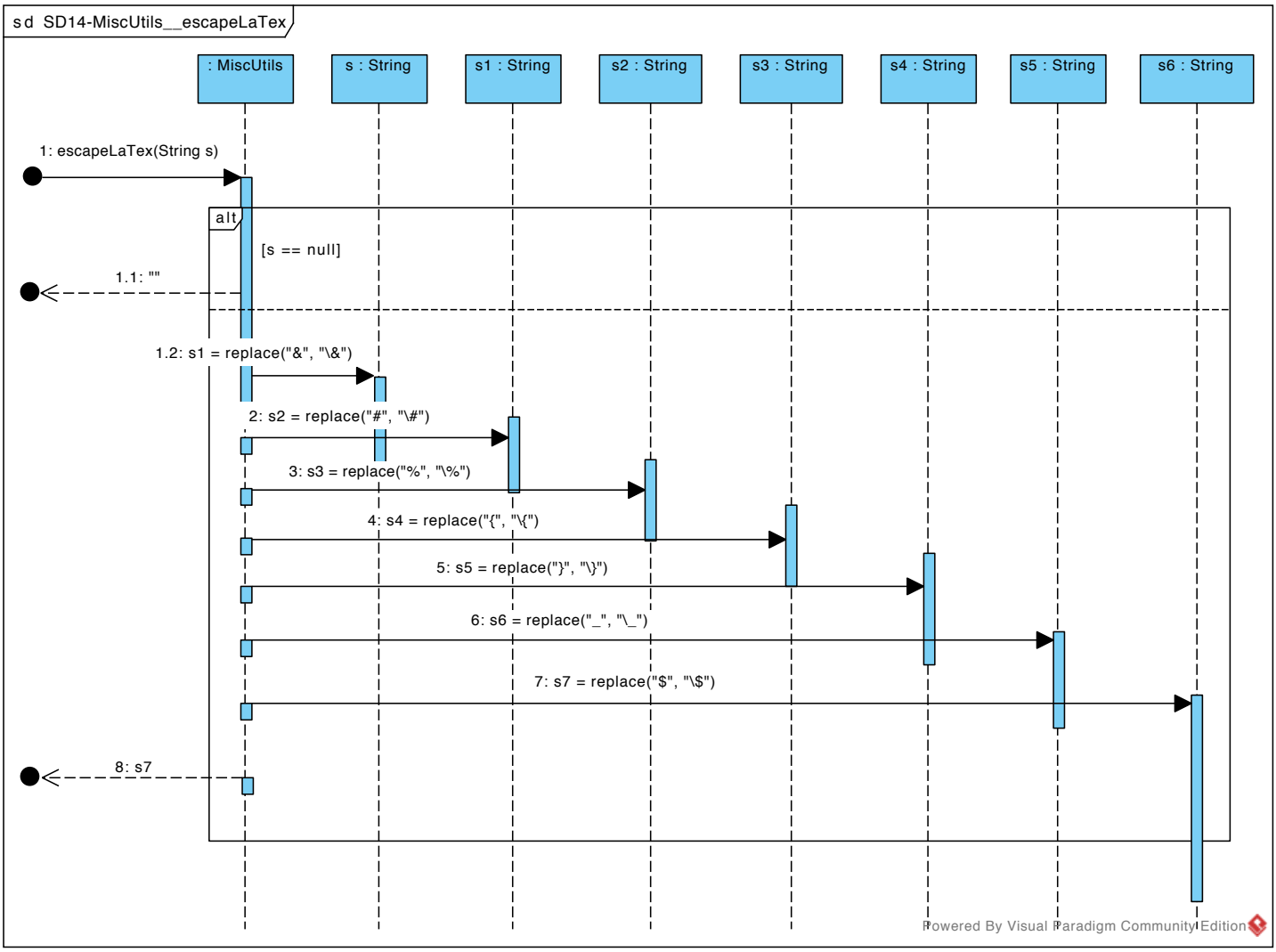

sd SD09
: HeaderCategoryEditPane
: JTextField[]
: JTextField
: Address
: HeaderCategory
: String
1: isModified()
alt
[if all return true]
1.1: get(0)
1.2: JTextField
1.3: getText()
1.4: text
1.5: getLink()
1.6: link
alt
1.7: link.equals(text)
1.8: true
1.9: false
1.9.1: get(1)
1.9.2: JTextField
1.9.3: getText()
1.9.4: text
1.9.5: getEmail()
1.9.6: email
alt
[if they are equal]
1.9.7: email.equals(text)
1.9.8: true
1.9.9: false
1.9.9.1: get(2)
1.9.9.2: JTextField
1.9.9.3: getText()
1.9.9.4: text
1.9.9.5: getPhoneNumber()
1.9.9.6: phoneNumber
alt
[if they are equal]
1.9.9.7: phoneNumber.equals(text)
1.9.9.8: true
1.9.9.9: false
1.9.9.9.1: getText()
1.9.9.9.2: text
1.9.9.9.3: setAddress(text)
1.9.9.9.4: address
alt
[if they are equal]
1.9.9.9.5: address.equals(text)
1.9.9.9.6: true
1.9.9.9.7: false
1.9.9.9.7.1: false
1.9.9.9.7.2: true
Powered By Visual Paradigm Community Edition

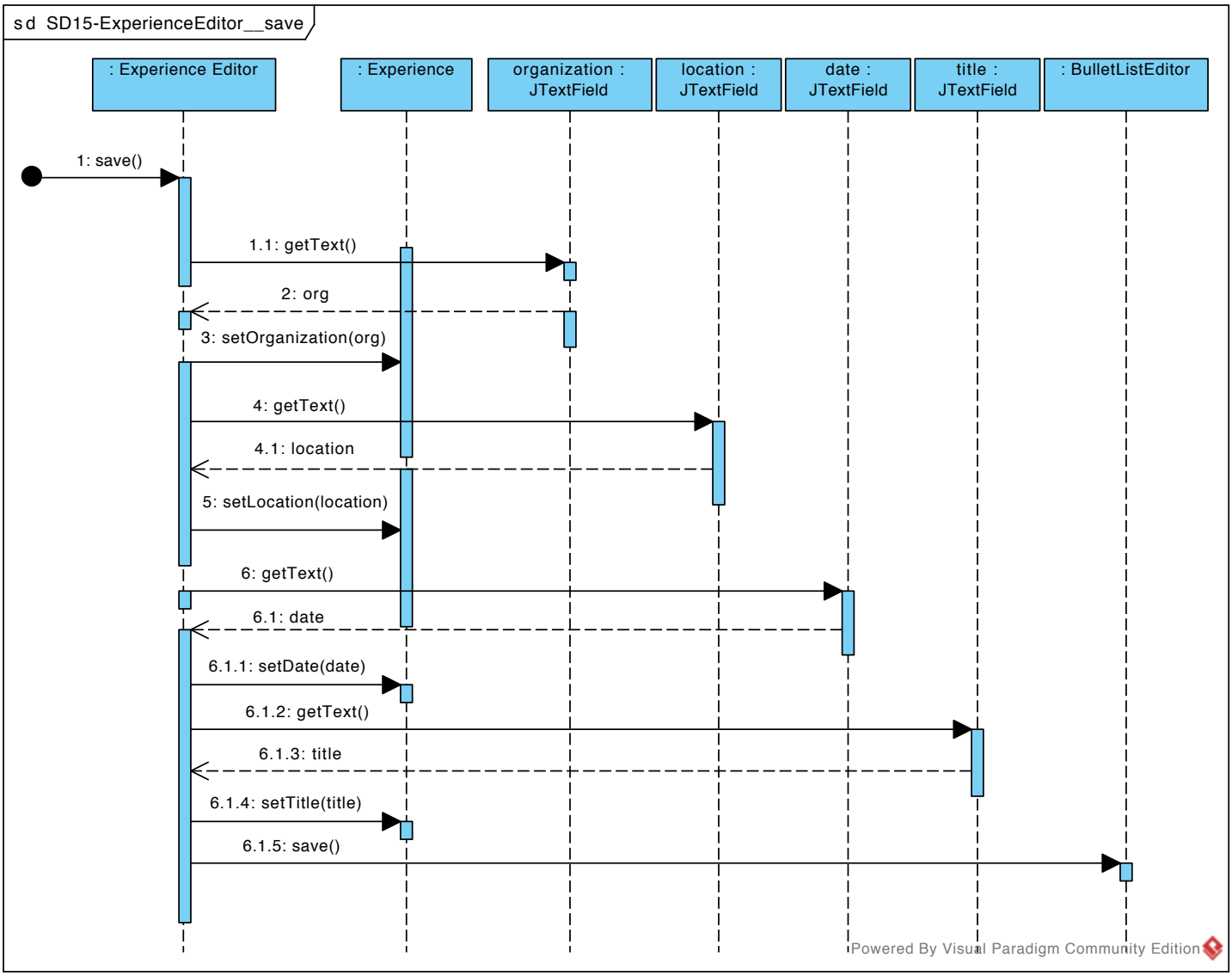

sd SD10-BulletCategory::addBullet
: BulletCategory
newBullet : Bullet
bullets : List<Bullet>
1: addBullet(newBullet : Bullet)
1.1: create()
rand : Random
loop
[unique is false]
An 'id' variable is initialized inline based off of 'tmp'. The result is then checked for uniqueness.
1.2: tmp = nextInt(1000)
1.3: unique = checkBulletListID(id)
1.4: copy = create(id)
copy : Bullet
1.5: tmpText = getText()
1.6: setText(tmpText)
1.7: tmpVisible = getVisible()
1.8: setVisible(tmpVisible)
1.9: add(copy)
1.10: id
Powered By Visual Paradigm Community Edition


sd SD11-HeaderCategory::formatLaTexString
: HeaderCategory
: MiscUtils
template : ResumeTemplate
categoryTemplates : Map<CategoryType, StringTemplate>
strTemplate : StringTemplate
newTemplate : StringTemplate
1: formatLaTexString(template : ResumeTemplate)
1.1: getCategoryTemplate(type)
1.1.1: strTemplate = get(type)
1.1.2: clone()
1.1.2.1: create()
1.1.2.2: newTemplate
1.1.2.2.1: escapedDisplay = escapeLaTex(displayName)
1.1.2.2.2: replaceVariable("name", escapedDisplay)
1.1.2.2.3: escapedAddress = escapeLaTex(address)
1.1.2.2.4: replaceVariable("address", escapedAddress)
1.1.2.2.5: escapedPhone = escapeLaTex(phoneNumber)
1.1.2.2.6: replaceVariable("phone", escapedPhone)
1.1.2.2.7: escapedLink = escapeLaTex(link)
1.1.2.2.8: replaceVariable("link", escapedLink)
1.1.2.2.9: escapedEmail = escapeLaTex(email)
1.1.2.2.10: replaceVariable("email", escapedEmail)
1.1.2.2.11: result = toString()
1.1.2.2.12: result
Powered By Visual Paradigm Community Edition

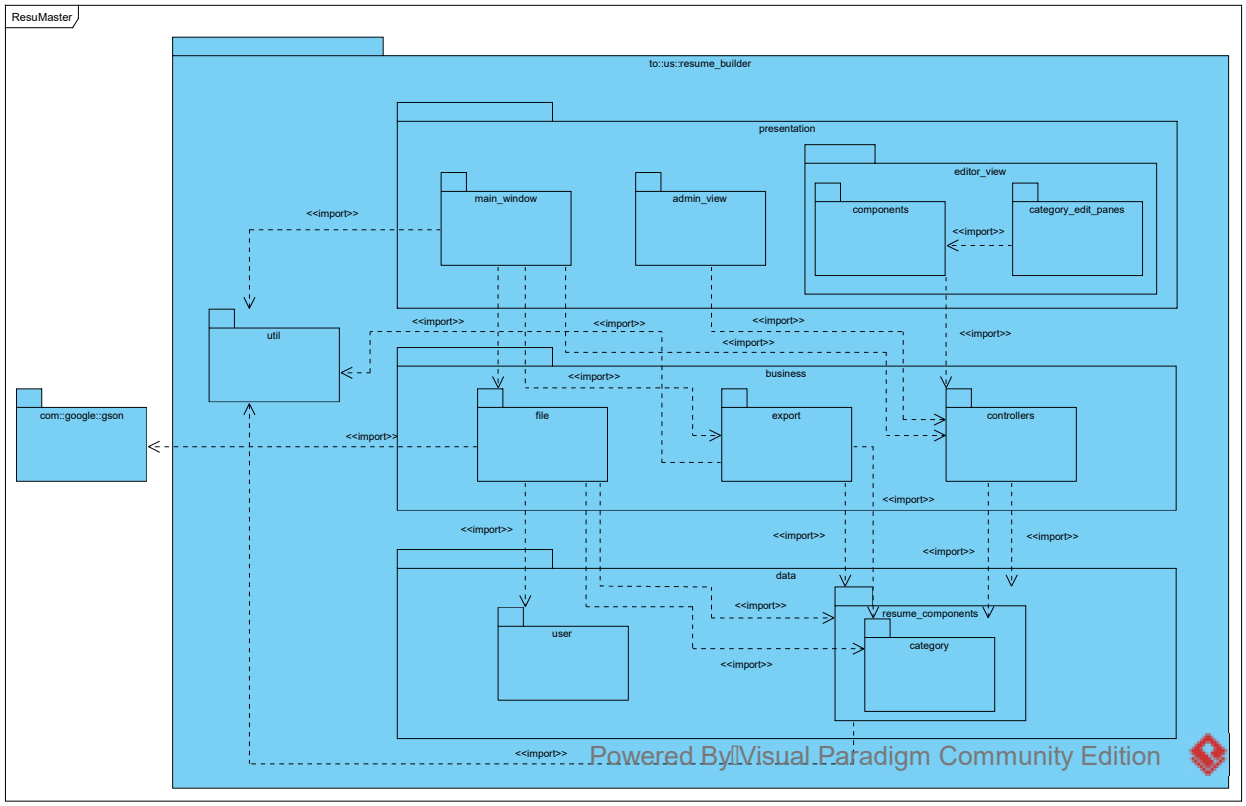

sd SD12-Experience::getBulletByID
: Experience
first : Optional
filteredList : List<Option>
bullet : Bullet
tmpID : String
bullets : List<Bullet>
1: getBulletByID(id)
1.1: stream()
bulletStream : Stream
1.1.1: create()
1.2: customFilter = create()
customFilter : Predicate
1.3: filter(customFilter)
loop
[i < bullets.size()]
1.3.1: bullet = bullets[i]
1.3.2: test(bullet)
1.3.2.1: tmpID = getID()
2: include = equals(id)
3: include
3.1: filteredList
3.1.1: first = findFirst()
3.1.2: result = orElse(null)
3.1.3: result
Powered By Visual Paradigm Community Edition


sd SD13-Resume__createCategory
: Resume
: List<Category>
1: createCategory(type)
1.1: <<create>>
: Random
loop
[unique is false]
1.2: id = nextInt(1000)
1.3: unique = checkCategoryListID(id)
alt
[type == HEADER]
1.4: create(id)
hc : HeaderCategory
1.5: add(hc)
alt
[type == TEXT]
1.6: create(id)
tc : TextCategory
1.7: add(tc)
alt
[type == EXPERIENCE]
1.8: create(id)
ec : ExperienceCategory
1.9: add(ec)
alt
[type == BULLET]
1.10: create(id)
bc : BulletCategory
1.11: add(bc)
1.12: id
Powered By Visual Paradigm Community Edition


sd SD14-MiscUtils__escapeLaTex
: MiscUtils
s : String
s1 : String
s2 : String
s3 : String
s4 : String
s5 : String
s6 : String
1: escapeLaTex(String s)
alt
[s == null]
1.1: ""
1.2: s1 = replace("&", "\&")
2: s2 = replace("#", "\#")
3: s3 = replace("%", "\%")
4: s4 = replace("{", "\{")
5: s5 = replace("}", "\}")
6: s6 = replace("_", "_")
7: s7 = replace("$", "\$")
8: s7
Powered By Visual Paradigm Community Edition


sd SD15-ExperienceEditor__save
: Experience Editor
: Experience
organization : JTextField
location : JTextField
date : JTextField
title : JTextField
: BulletListEditor
1: save()
1.1: getText()
2: org
3: setOrganization(org)
4: getText()
4.1: location
5: setLocation(location)
6: getText()
6.1: date
6.1.1: setDate(date)
6.1.2: getText()
6.1.3: title
6.1.4: setTitle(title)
6.1.5: save()
Powered By Visual Paradigm Community Edition

Package Diagram


ResuMaster
to::us::resume_builder
presentation
main_window
admin_view
editor_view
components
category_edit_panes
util
business
file
export
controllers
com::google::gson
data
user
resume_components
category
<<import>>
Powered By Visual Paradigm Community Edition

GRASP

Notes: Create Account, Request Review, Write Review Comments, Change User Role, Send Email With PDF, Modify User Details, and Check Review Comments have all been postponed until Iteration 3. Preview Resume was determined to not offer much value to the end user and was thus scrapped. Finally, the term ‘Item’ was renamed to ‘Experience’ and ‘Field’ renamed to ‘Bullet.’ Thus, old system operations referencing ‘Item/Field’ were re-named to ‘Experience/Bullet.’

Edit Category

- Operation Location: CategoryEditPane (Bullet,Text, Header, and Experience) and Component Editors (Bullet and Experience)
- Operation Rationale:
    - Polymorphism: Each Category and Component of the Resume behave differently by type, resulting in the multiple classes.
    - High Cohesion: Since each Category needs unique edit operations, splitting the operation across these multiple classes prevents any one from becoming incohesive.

Edit Experience/Bullet

- Operation Location: Bullet, Experience, IBulletContainer (implemented by BulletCategory and Experience), ExperienceCategoryEditPane, BulletCategoryEditPane, ExperienceEditor, BulletListEditor
- Operation Rationale:
    - Pure Fabrication: both ExperienceEditor and BulletCategoryEditPane use bullet editing and are Information Experts with regards to the bullet-possessing objects; however, duplicating the code to manage editing a list of bullets in both is a major violation of DRY. The BulletListEditor was the solution to this problem.
    - Polymorphism: The IBulletContainer creates a pluggable, re-usable interface for interacting with editing a set of bullets. As multiple different parts of a Resume need to do this, namely experiences and bullet categories, creating a common interface allows classes to be made which can act on either with the same interface.
    - Polymorphism: As noted above, all editors--like the ExperienceCategoryEditPane and the BulletCategoryEditPane--inherit from one central class, to allow them to act as a pluggable editing solution. This was equally important here, to allow Item/Field editing to be as seamless as possible--in spite of not all EditPanes offering Item editing.

Create/Remove Category

- Operation Location: EditorController
- Operation Rationale:
    - Controller: EditorController listens for events occuring in EditorAddCategoryButton and EditorCategorySelector.

    - Information Expert: The EditorController is connected directly to the Resume, as well as to the components that would know if the User wanted to add a Category. This makes it the expert when it comes to category edits.

Export Resume

- Operation Location: ResumeExporter
- Operation Rationale:
    - High Cohesion: This method is only concerned with exporting the Resume to a specified Path, using a given ResumeTemplate.
    - Low Coupling: Only coupled with Resume, Path, Files, and ResumeTemplate, and ApplicationConfiguration. These are all necessary in order to export the Resume.

Edit Header Links/Description

- Operation Location: HeaderCategoryEditPane
- Operation Rationale:
    - Information Expert: Based off of a decision early on, the Header was made its own unique Category to simplify the Resume's internals. The HeaderCategoryEditPane is the editor for this. It is thus tied to these fields already, making it an expert.
    - Low Coupling: Since the HeaderCategoryEditPane is already the only class directly aware of these attributes of the Header Category, placing this responsibility here lowers coupling of other parts of the design.

Add/Remove Bullet/Experience

- Operation Location: Bullet, Experience, IBulletContainer (implemented by BulletCategory and Experience), ExperienceCategoryEditPane, BulletCategoryEditPane, ExperienceEditor, BulletListEditor
- Operation Rationale:
    - Pure Fabrication: As noted earlier, the BulletListEditor is a pure fabrication to decouple bullet editing from the ExperienceEditor and BulletCategoryEditPane. This Pure Fabrication also offers bullet removal as reusable functionality, and preserves the cohesion of the other various editors.
    - Polymorphism: The IBulletContainer defines the interface for adding and removing Bullets and Experiences.
    - Pure Fabrication: As noted earlier, IBulletContainer is a pure fabrication to allow the interface of interacting with a list of bullets to be re-used between Experiences and BulletCategories.
    - Polymorphism: The EditPane components, as noted earlier, all inherit from one central class, to allow them to act as a pluggable editing solution. This was equally important here, to allow Experience and Bullets to be added and removed as seamlessly as possible--in spite of not all EditPanes handling Items.

Changing Visibility of Experience/Bullet

- Operation Location: Bullet, Experience, IBulletContainer (implemented by BulletCategory and Experience), ExperienceCategoryEditPane, BulletCategoryEditPane
- Operation Rationale:
    - Pure Fabrication: Changing visibility is another of the responsibilities placed in the BulletListEditor Pure Fabrication, as it is an operation needed by all of the various bullet-editing components; this offers bullet visibility toggling as a reusable functionality and preserves the cohesion of the other various editors.
    - Polymorphism: The IBulletContainer defines the interface for setting visibility of Bullets and Experiences.
    - Polymorphism: EditPanes are polymorphic to ease the design of the main editor.

Import Resume Data File

- Operation Location: ResumeFileManager
- Operation Rationale:
    - High Cohesion: Only concerned with importing a ResumeFile from the specified JSON file to return to the method caller. It delegates reading the file to a FileReader and deserializing JSON to a Gson object.
    - Low Coupling: The method is only dealing with the file containing the JSON, the Gson object to interpret the JSON, and ResumeFile to give the caller the requested ResumeFile.

Export Resume Data File

- Operation location: ResumeFileManager, EditorMenuBar, MenuController
- Operation rationale:
    - Controller: The EditorMenuBar receives the UI inputs from the users
    - Indirection: The MenuController disconnects the EditorMenuBar and the ResumeFileManager
    - High Cohesion: Only concerned with exporting a ResumeFile to the specified JSON file. It delegates writing the file to a FileWriter and serializing the ResumeFile to JSON to a Gson object
    - Low Coupling: The method is only concerned with converting the ResumeFile to JSON with a coupled Gson object, and writing the file with a FileWriter.

Toggle Category

- Operation Location: EditorCategoryHeader
- Operation Rationale:
    - Information Expert: The EditorCategoryHeader already has access to a Category, and specifically handles visualizing the visibility status and storing user suggestions regarding what this should be, making it the information expert in visibility.

- Low Coupling: the EditorCategoryHeader already handles displaying the visibility data, so making it read/write does not couple it to new data and removes the responsibility from other objects.

Test Coverage Plan

- Testing if saving a file and reloading it gives the same resume.
    - Testing each type of export and import that we offer, making sure data is stored and got correctly

- Testing each class and method in **editor_view**
    - *Testing each class and method in **category_edit_pane***
    - BulletCategoryEditPane
        - Construct Bullet Category Edit Pane
            - Make sure that all buttons do as should and that the pane loads with no thrown exceptions
        - Saving Bullet Category
            - Make sure that the data is saved in the correct place and correct information
        - Checking if Bullet Category isModified
            - Edit the text and/or visibility of the bullet and isModified should return true, else false
    - ExperienceCategoryEditPane
        - Construct Experience Category Edit Pane
            - Make sure that all buttons do as should and that the pane loads with no thrown exception
        - Saving ExperienceCategory
            - Make sure that the data is saved in the correct place and correct information
        - Checking if Experience Category isModified
            - Check changing each item and not changing and see if it returns true when changes happen and false when changes do not happen
        - Updating UI when an Experience is added or the Category changes
            - Check when adding an experience that the display changes
    - Header Category
        - Construct Header Category Edit Pane
            - Make sure that all buttons do as should and that the pane loads with no thrown exception
        - Saving Header Category
            - Make sure that the data is saved in the correct place and correct information
        - Checking if Header Category isModified
            - Check changing each item and not changing and see if it returns true when changes happen and false when changes do not happen
    - TextCategoryEditPane
        - Construct Text Category Edit Pane

            - Make sure that all buttons do as should and that the pane loads with no thrown exception
        - Saving Text Category
            - Make sure that the data is saved in the correct place and correct information
        - Checking if Text Category isModified
            - Check changing each item and not changing and see if it returns true when changes happen and false when changes do not happen
    - *Testing each class and method in **components***
    - BulletListEditor
        - Constructor
            - does the panel do what is wanted
        - Testing of each button
            - Test that each button does what is intended, for moving make sure that exceptions thrown
        - Testing of saved
            - Make sure that the data is saved in the correct place and correct information
        - Testing of isModified
            - Check changing each item and not changing and see if it returns true when changes happen and false when changes do not happen
    - BulletListEditorTableModel
        - Test addBullet
            - Make sure that adding a bullet the display is shown and that it has all the elements of a bullet
        - Test removeBullet
            - Make sure than the resume does not contain that bullet but ID
        - Test moveUp
            - Make sure that no exception are thrown and that when resume is reloaded that the bullets are in the correct place
        - Test moveDown
            - Make sure that no exception are thrown and that when resume is reloaded that the bullets are in the correct place
    - ExperienceEditor
        - Constructor
            - does the panel do what is wanted
        - Testing of each button
            - Test that each button does what is intended, for moving make sure that exceptions thrown
        - Testing of saved

                - Make sure that the data is saved in the correct place and correct information
            - Testing of isModified
                - Check changing each item and not changing and see if it returns true when changes happen and false when changes do not happen

- Testing each method and class in **export**:
    - Constructing a ResumeExporter
        - Test with null Resume.
            - If exception is thrown, pass
        - Test with a premade Resume and an empty resume.
            - If they match what was passed in, pass
    - Exporting a resume
        - Export a resume to PDF.
            - Test to see if a file exists, create the resume with the file name and check if it now exists.
    - Compiling a resume as a PDF
        - Empty resume
            - If return is true, pass.
        - Full resume
            - If return is true, pass.
        - Null path
            - If return is false, pass
    - Obtaining the correct LaTexString
        - Write the LaTexString to a file and compare with another file which contains the correct LaTexString.
            - If the files match, pass.
    - Constructing a resume template
        - Pass a templateName which does not exist
            - If an IOException is thrown, pass.
        - Pass a correct TemplateName
            - If the template is loaded accurately, pass.
    - Getters for Resume Template
        - Create a StringTemplate which is expected for Category template
            - If String from getter matches expected value, pass.
        - Create a StringTemplate which is expected for Experience Template
            - If String from getter matches expected value, pass.
        - Create a StringTemplate which is expected for Field template
            - If String from getter matches expected value, pass.
        - Create a StringTemplate which is expected for LaTex template
            - If String from getter matches expected value, pass.
        - Create a StringTemplate which is expected for Separator template

                - If String from getter matches expected value, pass.
            - **for each of these, test that changes to return value do not change internal values of Resume Template members**

- Testing each method and class in **file**:
    - Deserializing a Category
        - Make a Json file that is a resume with null and empty strings, import it and re-export it.
    - Serializing a list
    - Importing json file and Exporting to json file
        - For this one you would test by importing a file and taking the resume exporting it and reimporting it. If the first import and second import match then pass.
    - Testing all getters and setters
        - Pass a value to a setter and check that it matches what's returned from the corresponding getter.

- Testing each class and method in **main_window**
    - Add Category Button:
        - Make sure getType returns the correct selection
            - If the selected item is the Header type, pass.
        - Ensure setController and addCategory function correctly
            - If the user requests to add a Category within the Editor Frame and the selected Category is added, pass.
    - Editor Category Header:
        - Constructing a Editor Category Header
            - If no exceptions are thrown, pass.
            - If createLabelPanel and createActionPanel tests pass, pass.
        - Ensure changed display name and nickname save correctly
            - Use test data to set display name and nickname, then call save()
                - If display name and nickname match test data, pass.
        - Checking if an item has been modified
            - Add a letter to a field and call isModified
                - If return is true, pass.
            - Call isModified without modifying anything
                - If return is false, pass.
        - Constructing the Label panel
            - If no exceptions are thrown, pass.
        - Constructing an action panel
            - If the display name and nickname textfield texts match that of the category being displayed, pass.
            - If no exceptions are thrown, pass.
        - Make sure delete() removes the correct Category

- Delete the top Category
    - If second Category is the category member, pass
- Delete the bottom Category
    - If the category member doesn't change and the number of Categories in the resume is one less, pass.
    - If the selected Category is the new last Category, pass.

- Editor Category Selector
    - Constructing Editor Category Selector
        - If a blank resume results in an empty categories member, pass.
        - If a full resume results in the correct categories length, pass.
        - If each id corresponds to the correct Category in the resume, pass
    - Removing a category
        - If the Category previously below the removed category is loaded, pass.
        - If removing the last Category results in the above Category being loaded, pass.
        - If the last Category is unremovable, pass.
    - Adding a category
        - Add category that already exists
            - If no changes are made, pass.
        - Add category that doesn't exist
            - If the new Category is at the end of the list, pass.
            - If the new Category id matches the Category added, pass.
    - Moving categories up/down
        - If the category cannot be moved past the top, pass.
        - If the category cannot be moved past the bottom, pass.
        - Moving a category up
            - If the id of the category being moved up matches the id of the index before it after execution, pass.
        - Moving a category down
            - If the id of the category being moved down matches the id of the index after it after execution, pass.
    - Setting selected Category
        - If a null id results in no change, pass.
        - If no exception is thrown, pass.
        - Using a hardcoded id, if the selected Category id matches the id passed in, pass.
- Editor Controller
    - Constructing an Editor Controller (create)
        - Pass null arguments
            - If an exception is thrown, pass.
        - If arguments are non-null and no exception is thrown, pass.
    - Load resume → not fully implemented

        - Load category
            - Pass a valid id
                - If the id is selected in EditorCategorySelector, pass.
            - Pass an invalid id
                - If no changes are made, pass.
        - Remove/add Category
            - If no exceptions are thrown, pass.
            - Remove category:
                - If the editorCategorySelector's remove tests pass, pass.
            - Add Category:
                - If the new category is selected in editorCategorySelector, pass.
                - If the resume has another Category of the specified type, pass.
                - If the number of Categories has increased by one, pass.
        - Registering side list
            - If no exceptions, pass.
            - If the controller for the side list equals the object it was called on, pass.
        - Registering the editor stage
            - If no exceptions, pass.
            - If the controller for the stage equals the object it was called on, pass.
        - Repainting the side list when a Category gets selected
            - Add Category:
                - If the new category is still the last item in the list, pass.
            - Remove Category:
                - If the categories get shifted up/down correctly in the repainted version of the left panel and the correct Category is loaded, pass.
                    - Test this with removing first, middle, and bottom category
    - Editor Frame
        - Constructing an editor frame
            - If no exceptions are thrown, pass.
            - If r is null and an exception is thrown, pass.
    - Editor Menu Bar
        - Constructing an editor menu bar
            - Export Data File Menu
                - Exporting to a new file
                    - If a new file is created and matches desired structure, pass.
                - Exporting to existing file

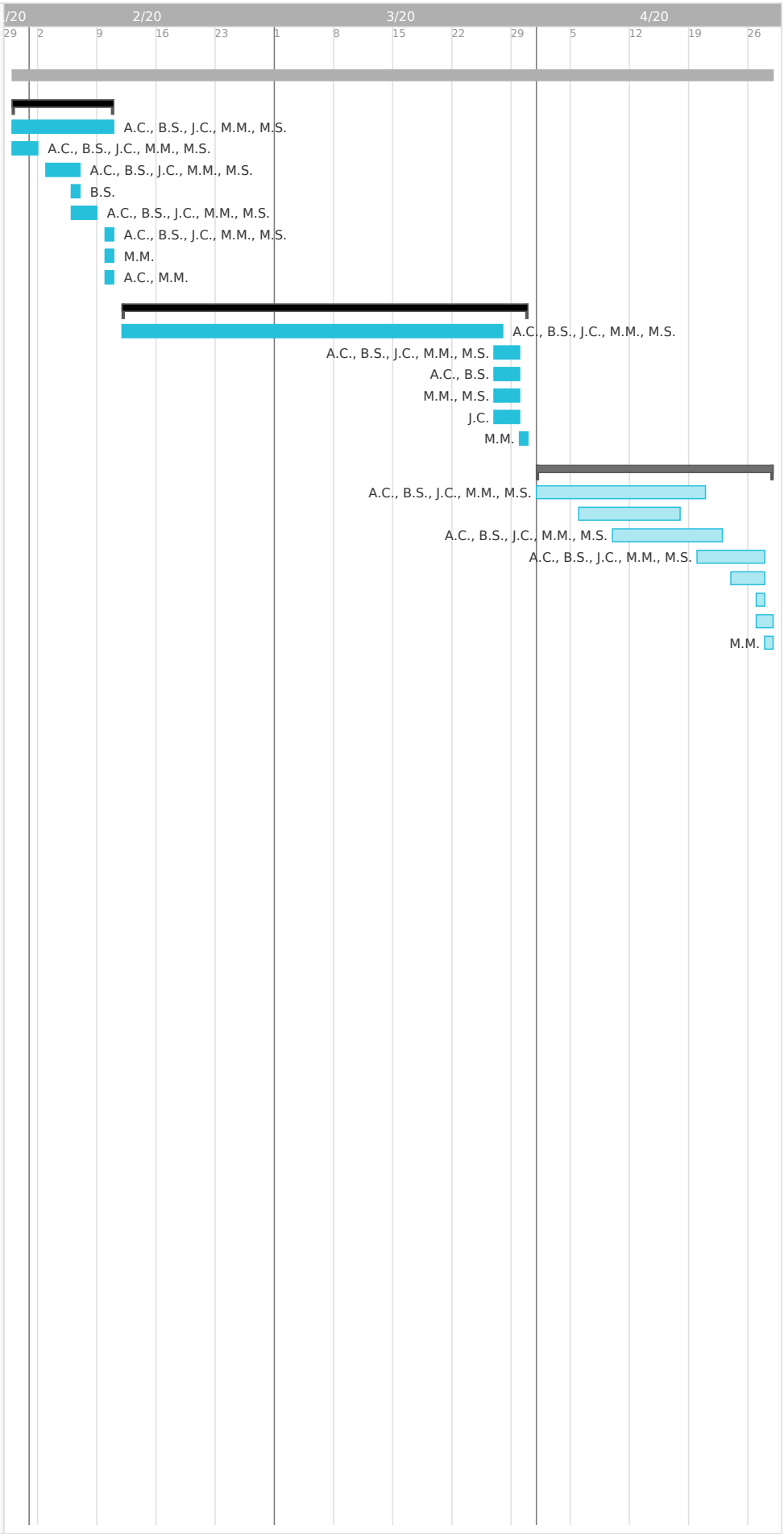
                        - If the existing file is overwritten with the correct structure, pass.
                    - ***When I say correct structure, this means the same order of categories and bullets that the user has requested***
                    - If the controller doesn't throw any exceptions, pass.
                    - If file extension is json, pass.
                - Export Resume
                    - Exporting to a new file
                        - If a new file is created and matches the desired structure, pass.
                    - Exporting to an existing file
                        - If the existing file is overwritten with the desired structure, pass.
                    - If the controller doesn't throw any exceptions, pass.
                    - If the extension is pdf, pass/
            - Setting controller
                - If this controller is a MenuController with the correct resume, pass.
                    - Check if the resume exported on its own matches the file exported through the EditorMenuBar
    - Editor Stage
        - Constructing an Editor Stage
            - If the controller type for the Category matches the Category passed in, pass.
            - If no exceptions are thrown, pass.
            - If the Category selected in the EditorCategory matches the Category passed to the constructor, pass.
        - Displaying category in the Editor Stage
            - If the id of the category in the EditorCategoryHeader matches the id of the Category passed, pass.
            - If the type of the CategoryEditPane matches the type of the Category passed, pass. (test with different Category types)
        - Set controller
            - If the controller is set to what is passed to it, pass.
            - If the controller has access to this EditorStage, pass.
                - Test different methods of the controller on the EditorStage and make sure changes were made.
        - Getter for Category id
            - If the id matches that of the startingCategory before any changes have been made, pass.
            - After showInEditor(Category toEdit)
                - If the toEdit id matches the id delivered by this method, pass.

            - Make sure saves are only initialized and finalized by the user and make sure save actually saved
                - Make changes to resume
                    - If the user elects to change and those changes are confirmed, pass.
                    - If the user elects not to save those changes and the resume remains identical before and after, pass.
    - Menu Controller
        - Construct menu controller
            - If the resume matches what is passed, pass.
        - Save as JSON
            - If no exception is thrown and none is expected, pass.
            - If passing a null or incorrect path results in an exception, pass.
        - Overwrite JSON
            - If no exception is thrown, pass.
        - Export as PDF
            - If given a valid path and no exceptions are thrown, pass.
            - If given an invalid path and an exception is thrown, pass.

- Testing each function in **resume_components**.
    - Making and removing category
        - Test adding by adding a category to a resume and seeing if the resume contains the category, if it contains then pass.
        - Test removing by removing the category the checking contains, if contains returns false then pass.
    - Adding and removing bullets
        - Test adding by adding a bullet to a category and seeing if the category contains the bullet, if it contains then pass.
        - Test removing by removing the bullet the checking contains, if contains returns false then pass.
    - Check contains for experience list
        - By hardcoding an ID and checking if it contains
    - Adding and removing experience
        - Add an experience, keep the returned ID, and see if the list contains that ID
        - Call removeExperience by passing an ID, see if the list contains the ID if it does then fail. If it does not then pass
    - Sets and gets for all
        - For each of these we will set them to a value and then get the value, if they are the same then it passes.
    - formatLaTeXString function
        - If the function returns a string that matches what is expected then pass.

- Testing each class and method in **util**
    - Getting file from file chooser
        - If no exceptions are thrown, pass.
            - (no exceptions should get thrown)
    - Using escape characters in Resume
        - If no escape characters have been used in the resume and the resume is the same before and after exporting, pass.
        - If escape characters are used, no exceptions should be thrown → pass.
        - If escape characters are used and the resume is reloaded with those characters correctly depicted, pass.
    - String formatting class?

Updated Project Plan (Gantt)



Resume Builder	**58%**
Iteration 1	**100%**
Domain Model	100%
Requirements	100%
Use Cases	100%
Wireframes	100%
System Sequence Diagrams	100%
System Operation Contracts	100%
Website	100%
Traceability Matrix	100%
Iteration 2	**100%**
Implementation	100%
Sequence Diagrams	100%
Test Coverage Plan	100%
Design Class Diagram	100%
Package Diagram	100%
Update Website	100%
Iteration 3	**0%**
Implementation	0%
Design Patterns	0%
Testing	0%
Use Case Videos	0%
Static Code Analysis	0%
Git Analysis	0%
User Guide	0%
Update Website	0%

Linked Issue Tracking System and Git

https://github.com/ZekNikZ/CSI3471-Resume-Builder

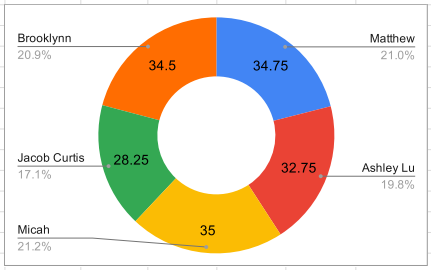
Time Sheet

Time Sheet

Team Member	Total	Misc. Total	Meeting 5	Misc. Total	Meeting 6	Misc. Total	Meeting 7	Misc. Total	Meeting 8	Misc. Total	Meeting 9	Misc. Total	Meeting 10	Misc. Total	Meeting 11	Misc. Total	Meeting 12
Matthew McCaskill	34.75	22.5	1		1.5		1		1		1.5		1		1.75		3.5
Ashley Lu Couch	32.75	1	1	4	1.5	2.5	1	1.5	1	2.5	1.5	2	1	2.5	1.75	4.5	3.5
Micah Schiewe	35		1	1.5	1.5	3	1	4	1		1.5	2	1	2	1.75	10.25	3.5
Jacob Curtis	28.25		1	2.5	1.5	1	1	4.5	1	2	1.5	1	1	0.5	1.75	4.5	3.5
Brooklynn Stone	34.5		1	1.5	1.5	2	1	7.5	1	1.5	1.5	3.75	1	1	1.75	5	3.5

Miscellaneous Entries

Team Member	Hours	Description
Matthew McCaskill	0.5	Data File Java Example
Jacob Curtis	2.5	Implementing ResumeFileManager
Micah Schiewe	1.5	Implementing basic User classes
Ashley Lu Couch	5.5	Finish Resume Components and Doc Comments
Micah Schiewe	3	Starting work on ResumeEditor
Jacob Curtis	1	Starting work on ResumeEditor
Micah Schiewe	4	Finish first draft of ResumeEditor
Jacob Curtis	4	Finish first draft of ResumeEditor and fixing ResumeFileManage
Brooklynn Stone	1.5	Method calls for UI actions (we didn't end up using)
Brooklynn Stone	2	Worked with Ashley on planning out EditPane
Brooklynn Stone	1.5	Started Experience component (now Matthew's)
Ashley Lu Couch	6.5	Worked on bulletComponent and bulletCategory
Brooklynn Stone	6	Header Category Edit Pane and Text Category Edit Pane
Micah Schiewe	3.5	Fixing JTextArea in Text Category Edit Pane (to resize properly
Brooklynn Stone	1.5	isModified method and GUI fixes in Header and Text Categories
Brooklynn Stone	0.5	Changed Header Category Address to accept multi-line input
Brooklynn Stone	1	Added JavaDoc comments to all editor-view classes
Brooklynn Stone	0.75	Worked on a dropdown option for adding new Categories
Micah Schiewe	2	Add category button
Micah Schiewe	2	Fixing header and adding Javadoc comments
Brooklynn Stone	1	Added labels for display name and nickname
Ashley Lu Couch	2.5	Changes to visuals on UI
Brooklynn Stone	4	Meeting with Ashley to discuss test plan and writing my portion
Brooklynn Stone	1	Sequence Diagrams
Matthew McCaskill	8	Resume Exporter
Matthew McCaskill	9	Experience component editor
Matthew McCaskill	3	Misc. features and look-and-feel upgrades
Matthew McCaskill	2	BulletComponent and other UI features
Micah Schiewe	10.25	Design Class Diagram, SDs
Ashley Lu Couch	4.5	SD's and Testing List
Ashley Lu Couch	1	Compile all PDFs



Point Redistribution

All points distributed evenly.