RESUMÉ REVIEW SOFTWARE

Iteration 1 Artifacts

Ashley Lu Couch Brooklynn Stone Jacob Curtis Matthew McCaskill Micah Schiewe

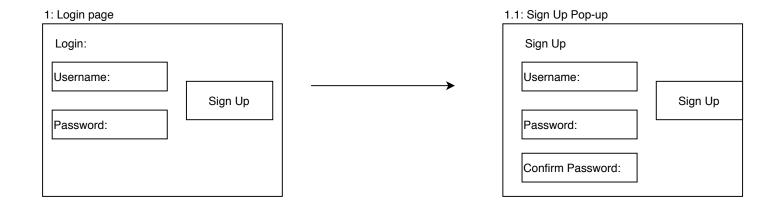
PAGE	ARTIFACT
3	Vision
5	Wireframes
8	Domain Model
10	Requirements
13	Traceability Matrix
15	Use Cases
42	System Sequence Diagrams
58	Operation Contracts
64	Gantt Diagram
70	Time Sheet
72	Point Redistribution
74	Presentation

Vision

Resumé Review Software: Vision

Creating a custom resumé for each company is a pain. We plan to deliver a unified resumé software. This software allows users to store data they would spread across separate resumés. It allows the user to vary their resumés depending on the company they would be applying to. Resumé writers can export resumés containing specific data segments in a formatted PDF file. Users can request to have their resumé reviewed in part or whole without having to make an appointment. Reviewers will leave constructive criticism in the form of comments. Users can use these fields to leave notes for themselves. The user will then see these comments and can make edits to their resumé. The user will also be able to import and export their resumé data to ease writing their resumé across computers. This will benefit our users by reducing stress around creating a resumé and help them to apply to positions in a more efficient manner.

Wireframes



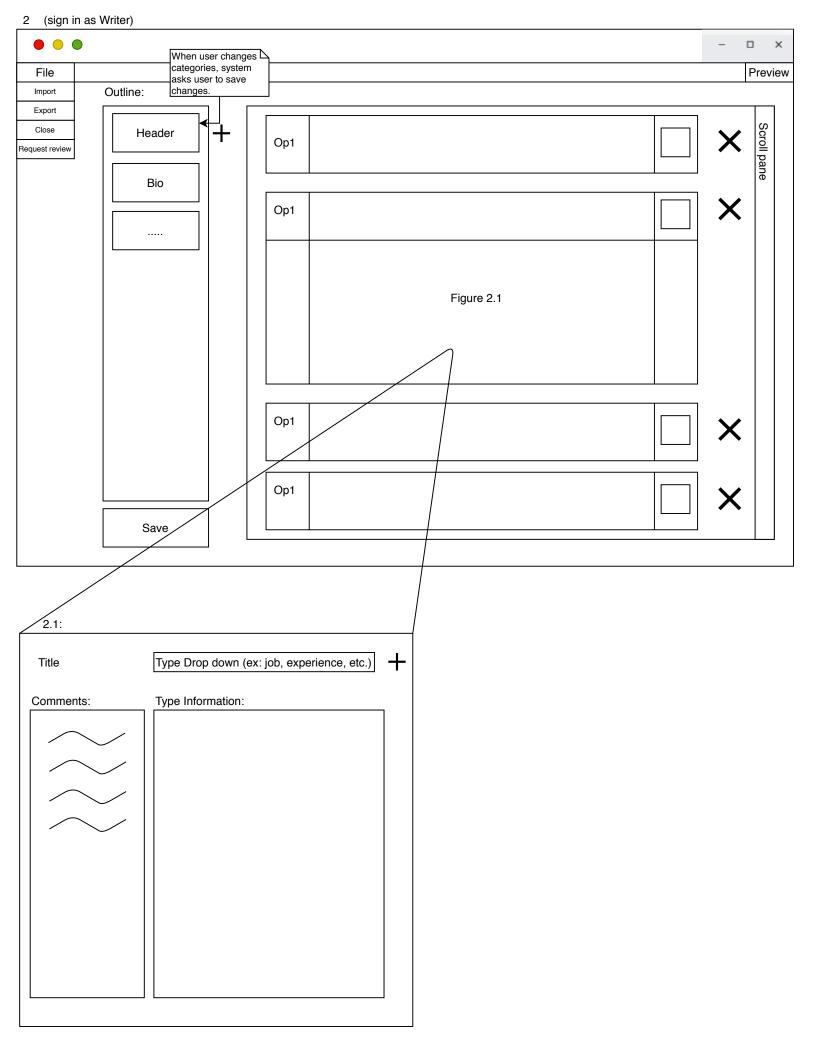
Search by Name

Resume - User

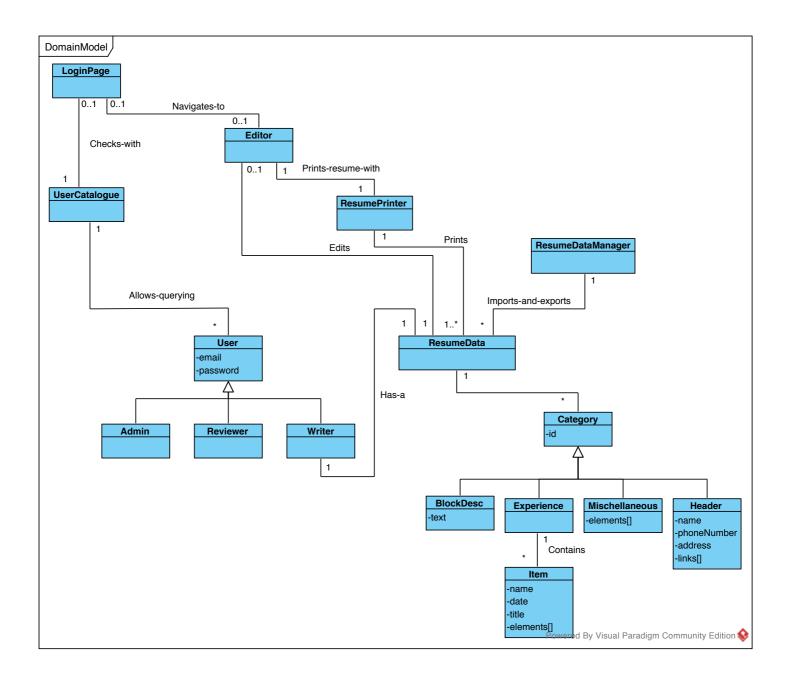
Resume - User

Resume - User

Resume - User



Domain Model



Requirements

Requirements:

R01 Modify Resumé Fields:

Description: A writer must be able to add, remove, and edit fields that are contained in categories within their Resumé. Here they will fill out their relevant information for the category they are writing in.

R02 Have Custom Categories:

Description: As a Resumé Writer, the user should be able to create custom Categories which they can add custom Items and Fields to.

R03 Modify User Details:

Description: A user (Reviewer, Writer, Admin) must be able to make changes to their account, or in the case of the Admin, to others'. This includes role changes, password resets, password changes, and adding users, as applicable.

R04 User/Reviewer Role Distinction:

Description: In the application, user types must be mutually exclusive. This implies that Admins may not review Writer Resumés, and Resumé Writers are not Admins.

R05 Export Resumé:

Description: Resumé Writers must be able to export their Resumé in the form of a formatted PDF, and also as a data file to be used on another machine with the application installed - allowing users to work across multiple machines.

R06 Send Review Request:

Description: As a Writer, the user must be able to request a Reviewer to view their Resumé and leave feedback. This will be in the form of an email sent from the System/Writer to a Reviewer with a request to Review the desired Categories.

R07 Have Comments:

Description: In order for a Review to be useful, the Reviewer must be able to leave comments on the Resumé on specific Categories and fields so that the Writer can get constructive feedback and be able to make changes to their Resumé

R08 Toggle Resumé Parts:

Description: For the purpose of customizing Resumés for various companies and requesting reviews, Resumé writers should have the ability to toggle which categories should be included in their exported Resumés.

R09 User Login:

Description: Users (Reviewers, Writers, Admins) should be able to enter in their email and password in order to gain access to the application. If a User does not have an account, they must first create one in order to log in.

Traceability Matrix

	AND THE POST OF DOS DOS DOS DOS DOS DOS								
UC01 - Create Account									X
UC02 - Request Review				Х		Х			X
UC03 - Write Review Comments				Х			Х		X
UC04 - Edit Items/Fields				Χ					X
UC05 - Create/Remove Category		Х		Х				Х	X
UC06 - Export Resume					Х				X
UC07 - Edit Header Links/Description			X	Χ				Х	X
UC08 - Check Review Comments				Х			Χ		X
UC09 - Add/Remove Items/Fields				Χ				X	X
UC10 - Edit Category				Χ					X
UC11 - Toggle Fields				Χ				X	X
UC12 - Change User Role			Χ	Χ					X
UC13 - Send Email With PDF				Х	Χ	Х			X
UC14 - Preview Resume					Х				X
UC15 - Import Resume Data File						Х			X
UC16 - Export Desume Data File					Х	Х			X
UC17 - Toggle Category		Х						Х	X
UC18 - Modify User Details			Χ	Χ					

Use Cases

Name: UC01 - Create an account Scope: Resumé Review Software

Level: User goal

Primary Actor: Writer

Stakeholders and Interests:

- Writer: Wishes to create a new account that will let them log in and be able to make a resume.

- Admin: Wishes to allow users to have an account that is not a duplicate.

Preconditions: none

Postcondition: An account has been created for that user and they can now login normally

Main Success Scenario:

- 1. The user opens the application.
- 2. The user indicates that they do not have an account.
- 3. The system prompts the user to create an account.
- 4. The user inputs their information needed for the account. (Email, password, name)
- 5. The System checks the user input for validity and duplicates.
- 6. The user is logged in.
- 7. The account information is saved.

- 4a. If the information inputted is already an account
 - 1. System will display that an account is made with that information.
 - 2. User will redirect to log in and change password.

Name: UC02 - Request Review Scope: Resumé Review Software

Level: User goal

Primary Actor: Writer

Stakeholders and Interests:

- Writer: Wishes to reliably notify a specific reviewer that his resumé is available to review.
- Reviewer: Wishes to be quickly notified of a request to review a resumé.
- Admin: Wishes to ensure non-Reviewers are not contacted regarding a resumé review.

Preconditions: The Writer is signed into the system.

Postcondition: The Reviewer has received an email notification that the Writer has a resumé to review. Alternatively, the Writer has cancelled the operation.

Main Success Scenario:

- 1. The Writer alerts the System that they wish to have their current Resumé reviewed.
- 2. The System notes the current Resumé the Writer has written (i.e., which fields are active) and creates a Resumé Review file linking to these fields in the Review Repository.
- 3. The System then requests which Reviewers the Writer wishes to alert.
- 4. The Writer inputs all desired Reviewers.
- 5. The System validates the requested Reviewers
- 6. The System generates an email to each of the requested Reviewers alerting them that the Writer wishes to have their Resumé reviewed

Alternate Flow:

*a: The Writer decides to cancel requesting a Review:

- 1. The System deletes the Resumé Review files generated in Step 2
- 2a. The Writer has no currently active fields in their Resumé:
 - 1. The System alerts the Writer that a review cannot be generated for an empty Resumé
- 5a. The Writer requests an invalid reviewer (e.g., someone with no account, someone who is not a Reviewer, et. cetera):
 - 1. The System alerts the Writer of the requested invalid Reviewer.
- 6a. The server cannot be reached to send an email:
 - The System alerts the Writer that the Reviewers cannot be contacted at the moment.
 - a. The Writer chooses to post their Resumé anyway.
 - b. The Writer decides to not post the Resumé given the inability to alert Reviewers:

1.	1. The System deletes the Resumé Review file created in Step 2.						

Name: UC03 - Write Review

Scope: Resumé Review Software

Level: User goal

Primary Actor: Reviewer
Stakeholders and Interests:

- Reviewer: wants to leave a review on a resumé for the Writer
- Admin: wants the Reviewer to successfully leave feedback on a resumé for the Writer to see these comments when they view the review
- Writer: wants to have their resumé reviewed by a Reviewer

Preconditions: The Reviewer is signed into the System and has either received a request to review a resumé or is accessing a resumé to review

Postcondition: The Reviewer has left their review on the resumé and the Writer has been notified by the System

Main Success Scenario:

- 1. The Reviewer chooses a Resumé they would like to review
- 2. The System opens the Resumé which the Reviewer would like to review
- 3. The Reviewer specifies which Category they would like to leave a comment on
- 4. The Reviewer types their comment to the Writer
- 5. The System records this comment from the Reviewer
- 6. The Reviewer submits the comment
- 7. The System saves this comment on the Resumé

Repeat steps 2-6 as needed

- 8. The Reviewer submits their comments
- 9. The System requests a final comment from the Reviewer
- 10. The Reviewer submits a final comment which is saved by the System
- 11. The Reviewer submits their review to the Writer
- 12. The System saves all comments from the Reviewer
- 13. The System alerts the Writer that their review is ready

- 2a. The Reviewer decides to come back and review later
 - 1. The System closes the resumé and the Reviewer can return to step 1 at any time
- 3a. The Reviewer cancels request to leave a comment on the Category

- 1. The System does not save the comment text and sends the Reviewer back to step 2
- 3b. The Reviewer does not want to leave any comments
 - 1. The System sends the Reviewer to step 7
- 4a. The Reviewer no longer wishes to leave a review on the specified Category
 - 1. The System does not save the comments and sends the Reviewer to step 2
- 8a. The Reviewer has decided to go back and add more comments
 - 1. The System sends the Reviewer back to step 2
- 8b. The Reviewer decides they no longer want to leave any of the comments they have made
 - 1. The System cancels their submit request
 - 2. The System cancels their review
 - 3. The comments are not saved by the System and the Reviewer may return to step 1 whenever they choose

Identifier: UC04i

Name: Edit Item(s)

Scope: Resumé Review Software

Level: user-goal

Primary Actor: Writer

Stakeholders and Interests:

• Writer: Wishes to edit a specific item and be able to save the change.

• Reviewer: Wishes to be able to see the modified value of the field.

Preconditions: The Writer knows which item(s) they would like to edit. Application is open to login page.

Postcondition: The item(s) that the Writer wanted to update have been changed and saved.

Main Success Scenario:

- 1. Writer enters login username and password and logs into the System.
- 2. System processes login information to determine validity.
- 3. System opens Writer's Resumé.
- 4. Writer chooses the Category to edit.
- 5. Writer chooses the item to edit.
- 6. Writer edits Item.

Writer repeats steps 4-6 as applicable until they are done editing items.

- 7. System saves changes.
- 8. Writer logs out.

- 1-2a. Login information is incorrect.
 - 1. System notifies Writer that their credentials are incorrect.
 - 2. Writer returns to step 1.
- 3-*a. Writer decides to cancel editing items.
 - 1. Writer logs out of the System.
 - 2. System returns to default state.
- 4a. Category does not exist: refer to UC05a.
- 5a. Item does not exist: refer to UC09ci.

Identifier: UC04f
Name: Edit Field(s)

Scope: Resumé Review Software

Level: user-goal

Primary Actor: Writer

Stakeholders and Interests:

• Writer: Wishes to edit a specific field and be able to save the change.

Reviewer: Wishes to be able to see the modified value of the field.

Preconditions: The Writer knows which field(s) they would like to edit. Application is open to login page.

Postcondition: The field(s) that the Writer wanted to update have been changed and saved.

Main Success Scenario:

- 1. Writer enters login username and password and logs into the System.
- 2. System processes login information to determine validity.
- 3. System opens Writer's resume.
- 4. Writer chooses category to edit.
- 5. Writer chooses the item to edit.
- 6. Writer chooses the field to edit.
- 7. Writer edits field.

Writer repeats steps 4-7 as applicable until they are done editing fields.

- 8. System saves changes.
- 9. Writer logs out.

- 1-2a. Login information is incorrect.
 - 1. System notifies Writer that their credentials are incorrect.
 - 2. Writer returns to step 1.
- 3-*a. Writer decides to cancel editing fields.
 - 1. Writer logs out of the System.
 - 2. System returns to default state.
- 4a. Category does not exist: refer to UC05a.
- 5a. Item does not exist: refer to UC09ci.

6a. Field does not exist: refer to UC09cf.

Name: UC05a - Create category Scope: Resumé Review Software

Level: User goal

Primary Actor: Writer

Stakeholders and Interests:

- Writer: Wishes to create a new category that they can add resume elements to.

Preconditions: The Writer is signed into the system.

Postcondition: A category has been added or removed from the list of resume Categories.

Main Success Scenario:

1. The Writer indicates that they would like to add a new category

- 2. The Writer chooses the type of category they would like to add from the choices provided by the System
- 3. The System will ask the user to provide the information about the Category being added
- 4. The Writer will give the Category a Title
- 5. The Writer will enter the information about each item they want to include in the new Category
- 6. The Writer indicates to the System that they would like to save their new Category.
- 7. The System will save changes
- 8. The System will send the Writer back to step 1.

- a. The Writer does not save changes at any point in the process
 - 1. Changes will not be saved by the System

Name: UC05b - Remove category Scope: Resumé Review Software

Level: User goal

Primary Actor: Writer

Stakeholders and Interests:

- Writer: Wishes to remove a category that they no longer want to be part of a resume. **Preconditions:** The Writer is signed into the system and has an available category to remove.

Postcondition: A category has been removed from the list of resume categories.

Main Success Scenario:

1. The Writer chooses the Category they want to remove.

- 2. The Writer indicates to the System that they want to remove that Category.
- 3. The System confirms that the Writer wants to remove the Category and all the data associated with the Category
- 4. The Writer confirms the removal
- 5. The System removes all data associated with the Category and the Category itself
- 6. The System updates the Resumé and saves changes.

Alternate Flow:

4a. If the Writer decides not to remove the category

1. The System does not remove any information.

Name: UC06 - Export Resumé Scope: Resumé Review Software

Level: User goal

Primary Actor: Writer

Stakeholders and Interests:

- Writer: Wishes to generate his resumé.

- Admin: Wishes to avoid cluttering the filesystem with many exported Writer resumés.

Preconditions: The Writer is signed into the system. **Postcondition:** The User has received the final resumé.

Main Success Scenario:

- 1. The Writer alerts the system that they wish to export their current Resumé.
- 2. The System notes the current Resumé the Writer has written (i.e., which fields are active) and generates a preview of the Resumé.
- 3. The Writer confirms that this Resumé is as desired.
- 4. The System prompts the user for how the Resumé should be delivered to them.
- 5. The Writer inputs how they wish to receive the Resumé.
- 6. The System generates a PDF of the final Resumé.
- 7. The System transmits the final PDF to the user in the manner they specified.

- *a: The Writer decides to cancel requesting export:
 - 1. The System exits the scenario.
- 2a. The Writer has no currently active fields in their resumé:
 - 1. The System alerts the Writer that a resumé cannot be generated with no fields.
- 3a. The Writer does not like the current resumé
 - 1. The Writer cancels exporting the resumé
 - 2. The System discards the preview and cancels the operation
- 6a. The export to PDF fails
 - 1. The Writer is alerted that exporting is unavailable
 - 2. The System notes the error to the system log
- 7a. The Writer requests their resumé by email:
 - 1. The System generates and sends an email with the PDF of the Resumé attached.

- 1a. The server cannot be reached to send an email:
 - 1. The System alerts the Writer that the resumé cannot be sent by email at the moment.
 - 2. The Writer selects a different option to export.
- 7b. The Writer requests their resumé saved to a file:
 - 1. The User selects the file to save their resumé to.
 - 2. The System saves the final PDF to the specified location.

Name: UC08 - Check Review Comments

Scope: Resumé Review Software

Level: User goal

Primary Actor: Writer

Stakeholders and Interests:

Writer: wants to read a Review on their resumé made by a Reviewer

- Admin: wants the Writer to successfully view feedback from a Reviewer
- Reviewer: wants the Writer to receive their feedback and be able to make constructive changes

Preconditions: The Writer is signed into the System and has received a notification that they have a Review to read or is checking for any past or new Reviews

Postcondition: The Writer has successfully checked for reviews or has read a review from a Reviewer

Main Success Scenario:

- 1. The Writer chooses a Review to open
- 2. The System loads the comments made by a Reviewer so that they are visible to the Writer
- 3. The Writer selects a Category and Item to check for comments
- 4. The System shows the Writer the selected Item and its corresponding comments
- 5. The Writer may make changes to items if they choose

See UC010 and UC009

- 6. The Writer may save any changes and the System records these changes
- 7. The Writer may choose to resolve comments
- 8. The Writer may choose to respond to a comment made by the Reviewer by emailing them

Repeat steps 3 through 8 as long as there are unresolved comments

9. The System saves any changes to comments

- 1a. The System has no reviews for the Writer
 - 1. The Writer does not have any comments from a new Review to look at
- 2a. The Reviewer did not leave any comments
 - 1. The System does not display any comments
- 3a. The Writer may choose to complete steps in UC010 and UC009 for this use case

- 6a. The Writer does not save changes
 - 1. The System does not record changes
 - 2. The changes made by the Writer are not saved
- 6b. The Writer makes no changes
 - 1. The System has no changes to records
- 7a. The Writer resolves no comments
 - 1. The System will display the comments as long as they are unresolved
 - 2. The Writer can come back to step 7 whenever they choose
- 8a. The Writer does not respond to the Reviewer
- 9a. See 6a

Identifier: UC09ci

Name: Create Item(s)

Scope: Resumé Review Software

Level: user-goal

Primary Actor: Writer

Stakeholders and Interests:

• Writer: Wishes to create specific item(s) and to be able to save the change(s).

• Reviewer: Wishes to be able to see the modified version of the resume.

Preconditions: The Writer knows which item(s) they would like to create. Application is open to login page.

Postcondition: The items(s) that the Writer wanted to create have been changed and saved.

Main Success Scenario:

- 1. Writer enters login username and password and logs into the System.
- 2. System processes login information to determine validity.
- 3. System opens Writer's resume.
- 4. Writer chooses Category to edit.
- 5. Writer indicates that they would like to add an Item.
- 6. Writer edits Item description and other applicable fields.

Writer repeats steps 4-6 as applicable until they are done creating items.

- 7. System saves any changes made.
- 8. Writer logs out of the System.

- 1-2a. Login information is incorrect.
 - 1. System displays an error message to Writer.
 - 2. Writer returns to step 1.
- 3-*a. Writer decides to cancel editing items.
 - 1. Writer logs out of the System.
 - 2. System returns to default state.
- 4a. Category does not exist: refer to UC05a.

Identifier: UC09ri

Name: Remove Item(s)

Scope: Resumé Review Software

Level: user-goal

Primary Actor: Writer

Stakeholders and Interests:

• Writer: Wishes to remove specific item(s) and to be able to save the change(s).

• Reviewer: Wishes to be able to see the modified version of the resume.

Preconditions: The Writer knows which item(s) they would like to remove. Application is open to login page.

Postcondition: The items(s) that the Writer wanted to remove have been changed and saved.

Main Success Scenario:

- 1. Writer enters login username and password and logs into the System.
- 2. System processes login information to determine validity.
- 3. System opens Writer's resume.
- 4. Writer chooses category to edit.
- 5. Writer chooses the item to remove.
- 6. Writer indicates to the System that they would like to remove an Item.

Writer repeats steps 4-6 as applicable until they are done removing items.

- 7. System saves any changes.
- 8. Writer logs out of the System.

- 1-2a. Login information is incorrect.
 - 3. System displays an error message to Writer.
 - 4. Writer returns to step 1.
- 3-*a. Writer decides to cancel editing items.
 - 3. Writer logs out of the System.
 - 4. System returns to default state.
- 4a. Category does not exist: refer to UC05a.

Identifier: UC09cf

Name: Create Field(s)

Scope: Resumé Review Software

Level: user-goal

Primary Actor: Writer

Stakeholders and Interests:

• Writer: Wishes to create specific fields(s) and to be able to save the change(s).

• Reviewer: Wishes to be able to see the modified version of the resume.

Preconditions: The Writer knows which fields(s) they would like to create. Application is open to login page.

Postcondition: The items(s) that the Writer wanted to create have been changed and saved.

Main Success Scenario:

- 1. Writer enters login username and password and logs into the System.
- 2. System processes login information to determine validity.
- 3. System opens Writer's resume.
- 4. Writer chooses category to edit.
- 5. Writer chooses item to edit.
- 6. Writer indicates to the System that they would like to add an Item.
- 7. Writer edits field text.

Writer repeats steps 4-7 as applicable until they are done creating items.

- 8. System saves any changes.
- 9. Writer logs out of the System.

- 1-2a. Login information is incorrect.
 - 1. System displays an error message to Writer.
 - 2. Writer returns to step 1.
- 3-*a. Writer decides to cancel editing items.
 - 1. Writer logs out of the System.
 - 2. System returns to default state.
- 4a. Category does not exist: refer to UC05a.
- 5a. Item does not exist: refer to UC09ci.

Identifier: UC09rf

Name: Remove Field(s)

Scope: Resumé Review Software

Level: user-goal

Primary Actor: Writer

Stakeholders and Interests:

• Writer: Wishes to remove specific field(s) and to be able to save the change(s).

• Reviewer: Wishes to be able to see the modified version of the resume.

Preconditions: The Writer knows which field(s) they would like to remove. Application is open to login page.

Postcondition: The field(s) that the Writer wanted to remove have been changed and saved.

Main Success Scenario:

- 1. Writer enters login username and password and logs into the System.
- 2. System processes login information to determine validity.
- 3. System opens Writer's resume.
- 4. Writer chooses category to edit.
- 5. Writer chooses the item to edit.
- 6. Writer indicates to the System that they would like to remove a Field.

Writer repeats steps 4-6 as applicable until they are done removing items.

- 7. System saves any changes.
- 8. Writer logs out of the System.

- 1-2a. Login information is incorrect.
 - 1. System displays an error message to Writer.
 - 2. Writer returns to step 1.
- 3-*a. Writer decides to cancel editing items.
 - 1. Writer logs out of the System.
 - 2. System returns to default state.
- 4a. Category does not exist: refer to UC05a.

Name: UC010 - Edit category Scope: Resumé Review Software

Level: User goal

Primary Actor: Writer

Stakeholders and Interests:

- Writer: Wishes to edit a category, name, type, or add an element

Preconditions: The Writer is signed into the system and a category already exists.

Postcondition: The information is saved to be referenced later.

Main Success Scenario:

- 1. The Writer picks the category they wish to edit by indicating to the System that they would like to edit a Category.
- 2. In the Writer navigates to the field that they want to edit.
- 3. The Writer edits the category in the correct way.
- 4. The System saves the changes.
- 5. The System keeps the changes updating the Resumé Data.

- 3a. If the Writer is wanting to change the name or edit an element:
 - 1. The Writer can enter any information to change the text or delete the text.
- 3b. If the Writer is wanting to change the **category type**:
 - 1. The Writer will navigate to the Category type and select a new type.
 - 2. Some specialized information from the previous Category type will be removed by the System
- 3c. If the Writer is wanting to **remove an element**:
 - 1. The Writer will remove the element
 - 2. The System will remove any element data
- 3d. If the user is wanting to add an element (list item):
 - 1. The Writer will add an element
 - 2. The System will create a space for them to enter the text for the element
- 4a. If this step does not happen changes will not be saved, and the Resumé will revert back to the previous state.

Identifier: UC11i

Name: Toggle Item(s)

Scope: Resumé Review Software

Level: user-goal

Primary Actor: Writer

Stakeholders and Interests:

• Writer: Wishes to toggle the visibility of specific item(s) and be able to save the change(s).

• Reviewer: Wishes to be able to see the modified toggles.

Preconditions: The Writer knows which item(s) they would like to toggle. Application is open to login page.

Postcondition: The item(s) that the Writer wanted to toggle have been toggled and saved.

Main Success Scenario:

- 1. Writer enters login username and password and logs into the System.
- 2. System processes login information to determine validity.
- 3. System opens Writer's Resumé.
- 4. Writer chooses Category to edit.
- 5. Writer indicates that they would like to toggle the specified Category.

Writer repeats steps 4-5 as applicable until they are done toggling items.

- 6. System saves changes.
- 7. Writer logs out of the System.

- 1-2a. Login information is incorrect.
 - 1. System displays an error message to Writer.
 - 2. Writer returns to step 1.
- 3-*a. Writer decides to cancel editing items.
 - 1. Writer logs out of the System.
 - 2. System returns to default state.
- 4a. Category does not exist: refer to UC05a.

Identifier: UC11f

Name: Toggle Fields(s)

Scope: Resumé Review Software

Level: user-goal

Primary Actor: Writer

Stakeholders and Interests:

• Writer: Wishes to toggle the visibility of specific field(s) and be able to save the change(s).

• Reviewer: Wishes to be able to see the modified toggles.

Preconditions: The Writer knows which field(s) they would like to toggle. Application is open to login page.

Postcondition: The field(s) that the Writer wanted to toggle have been toggled and saved.

Main Success Scenario:

- 1. Writer enters login username and password and logs into the System.
- 2. System processes login information to determine validity.
- 3. System opens Writer's resume.
- 4. Writer chooses Category to edit.
- 5. Writer chooses Item to edit.
- 6. Writer indicates that they would like to toggle the specified Category.

Writer repeats steps 4-6 as applicable until they are done toggling fields.

- 7. System saves changes.
- 8. Writer logs out of the System.

- 1-2a. Login information is incorrect.
 - 1. System displays an error message to Writer.
 - 2. Writer returns to step 1.
- 3-*a. Writer decides to cancel editing items.
 - 1. Writer logs out of the System.
 - 2. System returns to default state.
- 4a. Category does not exist: refer to UC05a.
- 5a. Item does not exist: refer to UC09ci.

Name: UC12 - Change User Roles Scope: Resumé Review Software

Level: User goal

Primary Actor: Admin

Stakeholders and Interests:

- Admin: Wishes to change the role of a user, either at their request or for some other reason. Wishes to retain control of who holds what role in the System.

- User: Wishes to be alerted should their role be changed.

Preconditions: The Admin is signed into the system and has found the User they wish to change the role of.

Postcondition: The Admin has changed a specified User's role. The User is aware of the change to their role.

Main Success Scenario:

- 1. The Admin alerts the System the wish to change the User's role
- 2. The System provides the Admin the possible roles for the User
- 3. The Admin selects the role they wish to change the User to
- 4. The System changes the User's role to the specified role
- 5. The System requests a message to send the User
- 6. The Admin enters a message to alert the User of their role change
- 7. The System sends an email with the specified message to the User

- 1a. The Admin is attempting to change their own role:
 - 1. The System alerts the Admin that they cannot change their own role
 - 2. The System cancels the operation
- 4a. The User whose role was changed was a Writer
 - 1. The System attaches the affected User's Resumé Data to an email
 - 2. The System sends the email to the affected User
 - 3. The System deletes the affected User's Resumé Data
- 4b. The User whose role was changed was not a Writer:
 - 1. The System creates the new Writer's Resumé Data file
- 5a. The Admin declines sending an email:
 - 1. The System skips steps 7-8

Name: UC13 - Send email with PDF Scope: Resumé Review Software

Level: User goal

Primary Actor: Writer

Stakeholders and Interests:

• Writer: Wishes to have their resume reviewed..

• Reviewer: Wishes to receive the Writer's Resume requesting review.

• Admin: Wishes for the Writer to have their Resume reviewed.

Preconditions: The Writer is signed into the system.

Postcondition: The User's Resume PDF is sent via email to a Reviewer

Main Success Scenario:

- 1. The Writer requests for their resume to be sent to a Reviewer
- 2. The System exports the Writer's resume to PDF
- 3. The System opens the User's default email program.
- 4. The System attaches the exported PDF to a blank email
- 5. The Writer fills out the email and sends it

- 2a. The export to PDF fails.
 - 1. The System cancels the operation and alerts the user.
- 5a. The Writer decides to not send the email.
 - 1. The Writer closes out the email.

Name: UC14 - Preview Resumé Scope: Resumé Review Software

Level: User goal

Primary Actor: Writer

Stakeholders and Interests:

- Writer: wants to view their resumé as a pdf before they export the resumé
- Admin: wants the Writer to preview only their resumé as a pdf

Preconditions:

 Writer is logged into the System and has entered their resumé information into different categories

Postcondition:

• The Writer successfully viewed their resumé as a pdf

Main Success Scenario:

- 1. The System loads all the Writer's information
- 2. The Writer may choose to make edits to their resumé

See UC009 and UC010

- 3. The System saves any changes made to the resumé
- 4. The Writer decides to preview their resumé as a pdf
- 5. The System loads the selected Categories into a resumé pdf format
- 6. The Writer views their resumé as a pdf

Repeat steps 2 through 6 as needed

- 1a. The Writer has not entered any information into the System
 - 1. The System does not load any information
- 2a. The Writer has no changes to make
 - 1. The System proceeds to step 4
- 5a. The Writer has not selected any Categories to include
 - 1. The System renders a blank pdf

Name: UC15 - Import Resume Data File

Scope: Resumé Review Software

Level: User goal

Primary Actor: Writer

Stakeholders and Interests:

• Writer: Wishes to use the Resume Data File they created on another computer.

• Admin: The input file does not contain corrupted data.

Preconditions: The Writer is signed into the system.

Postcondition: The Resume Data File's data is imported and is open in the editor

Main Success Scenario:

- 1. The Writer requests for the System to import a Resume Data File
- 2. The System prompts for the user to select the file
- 3. The Writer selects the file
- 4. The System validates the file extension of the file
- 5. The System parses the provided Resume Data File
- 6. The System opens the editor with the data read from the file

- 2a. The user cancels selecting a Resume Data File.
 - 1. The System is alerted and cancels the operation.
- 4a. The file contains an invalid file extension.
 - 1. The System alerts the user the provided file is invalid
 - 2. The System cancels the operation.
- 5a. The user is not associated with the provided Resume Data File.
 - 1. The System alerts the Writer they cannot import the Resume Data File.
 - 2. The System cancels the operation.
- 6a. The file does not follow the expected format.
 - 1. The System alerts the Writer the file could not be parsed.
 - 2. The System cancels the operation.

Name: UC16 - Export Resume Data File

Scope: Resumé Review Software

Level: User goal

Primary Actor: Writer

Stakeholders and Interests:

• Writer: Wishes to create a Resume Data File for use on another computer.

• Admin: Wishes for the Writer to receive their Resume Data File.

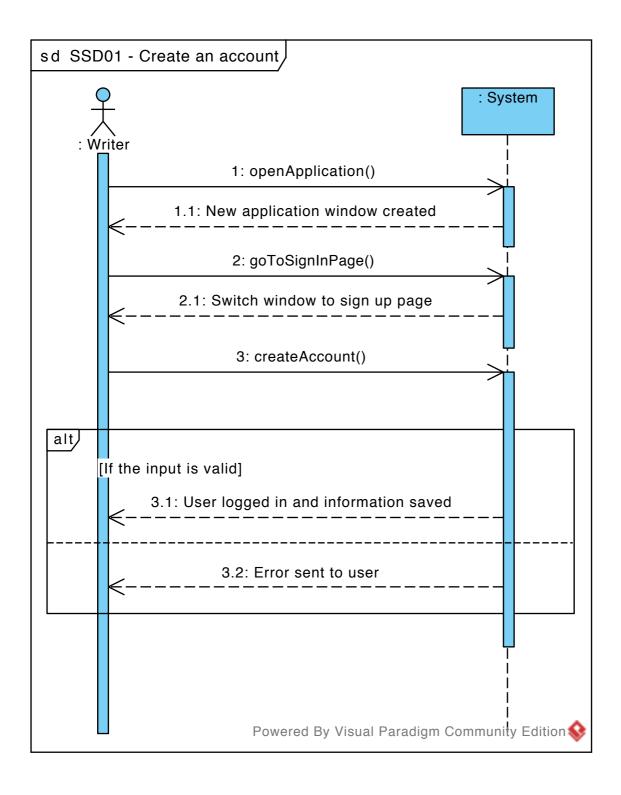
Preconditions: The Writer is signed into the system. **Postcondition:** The Resume Data File is exported.

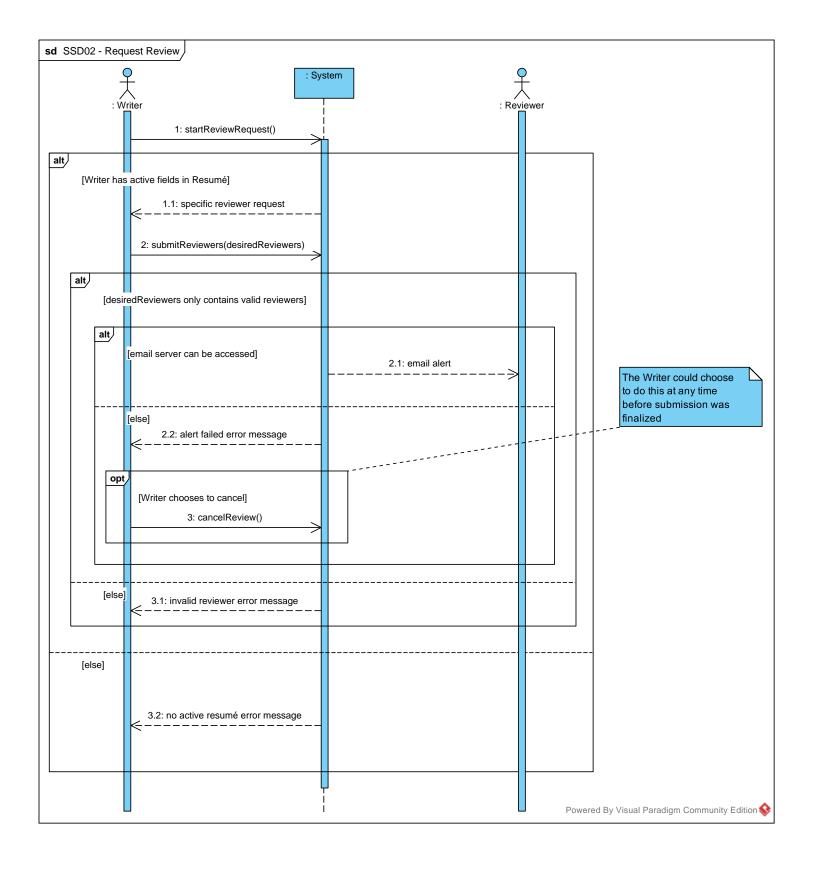
Main Success Scenario:

- 1. The Writer requests for their Resume Data File
- 2. The System requests the Writer to specify the folder to save the file in
- 3. The System compiles all of the Writer's Resume Data and the currently toggled fields
- 4. The System creates the Resume Data File
- 5. The System writes the resume data to the created file

- *a. The user requests to cancel the operation.
 - 1. The System cancels the operation.
- 4a. The Resume Data File failed to be created.
 - 1. The System cancels the operation and alerts the user.
- 5a. There was an error writing to the file.
 - 1. The System deletes the created file.
 - 2. The System cancels the operation and alerts the user.

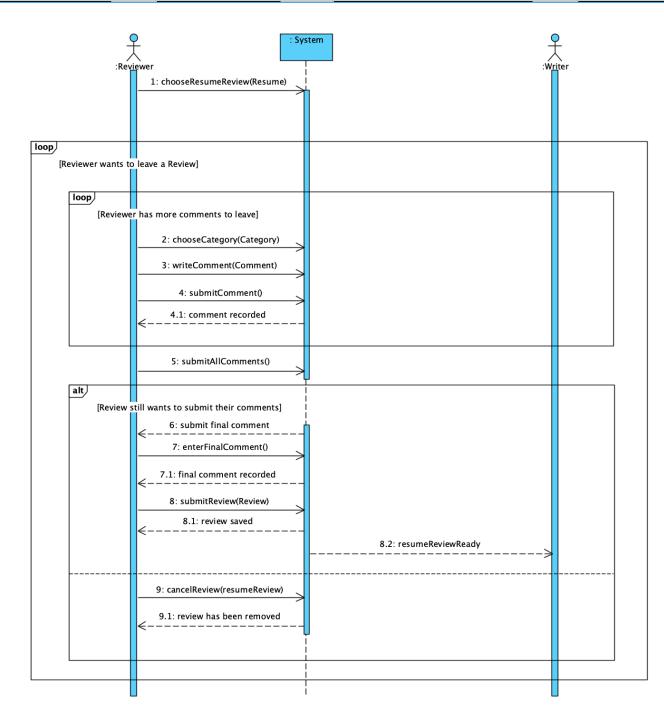
System Sequence Diagrams

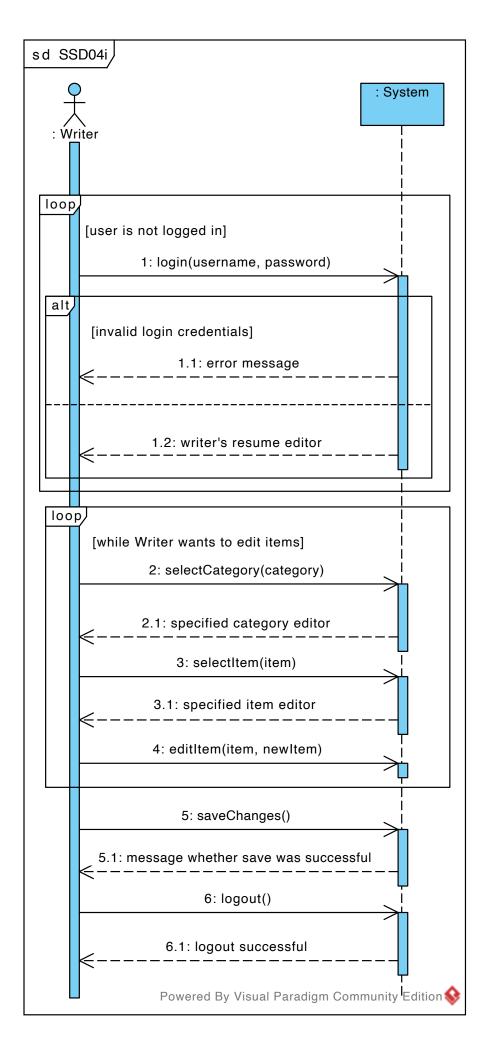


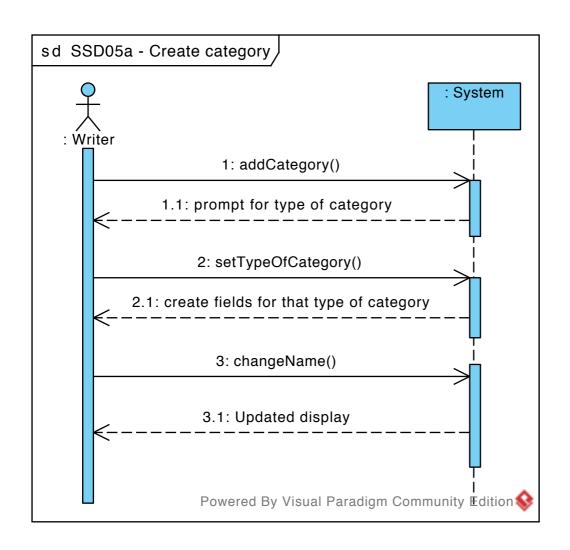


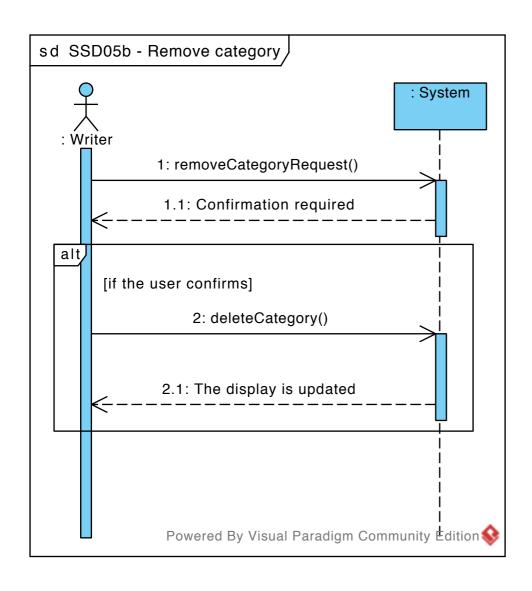
:Revie... : System :Writer

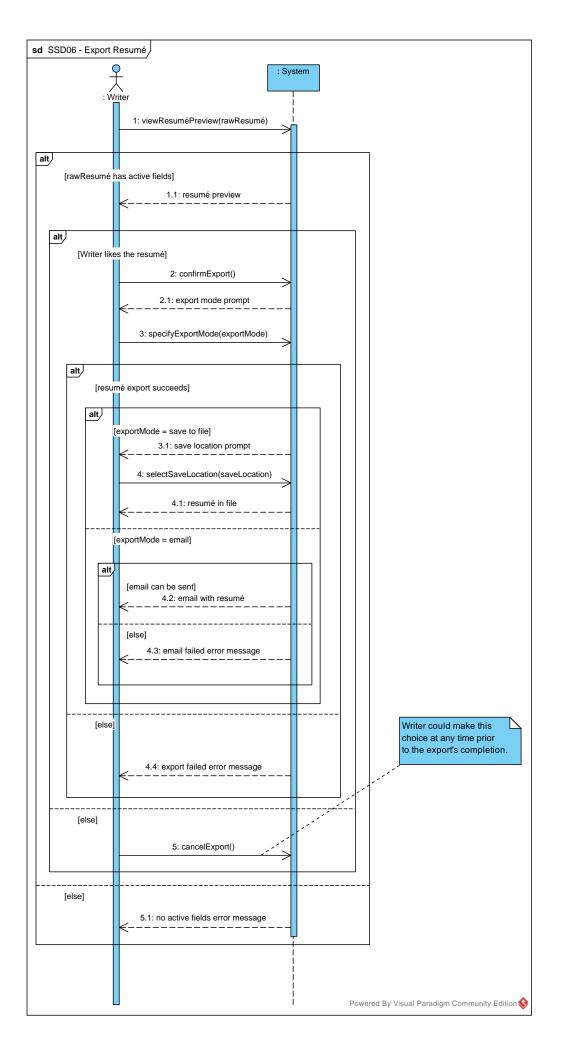
sd SSD03

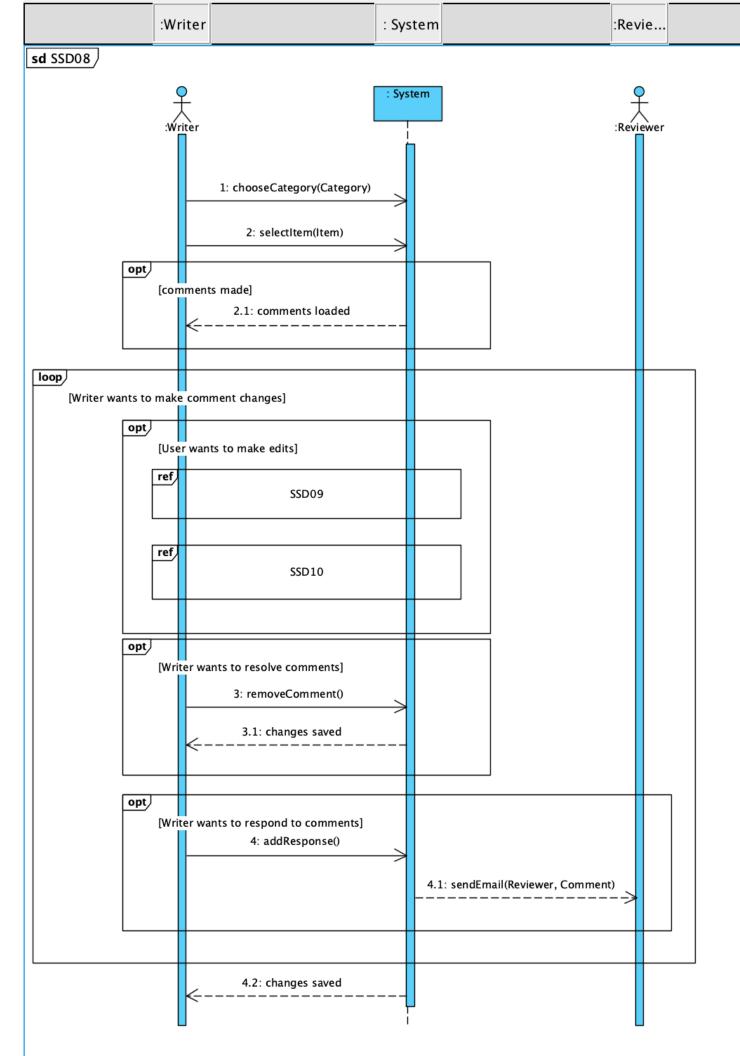


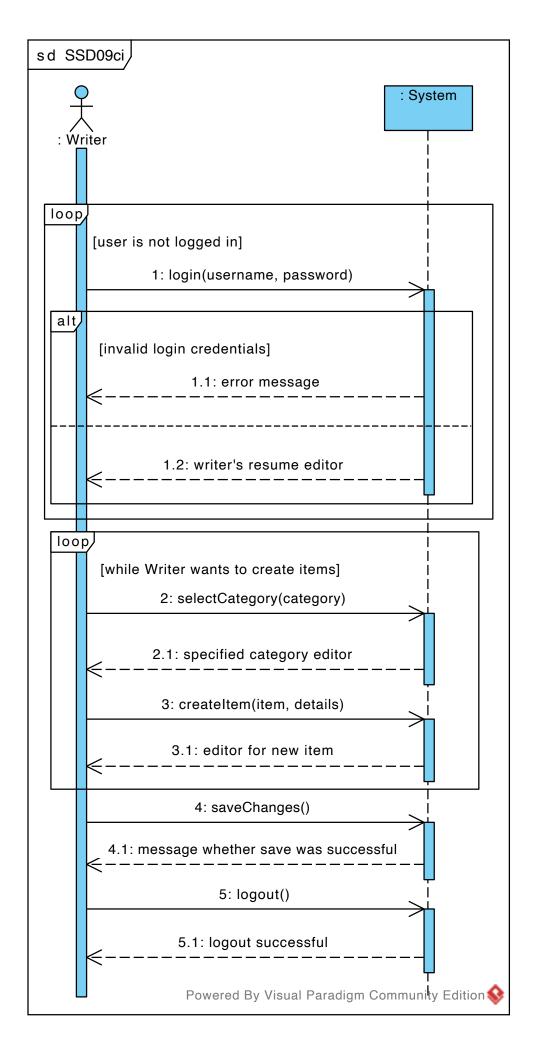


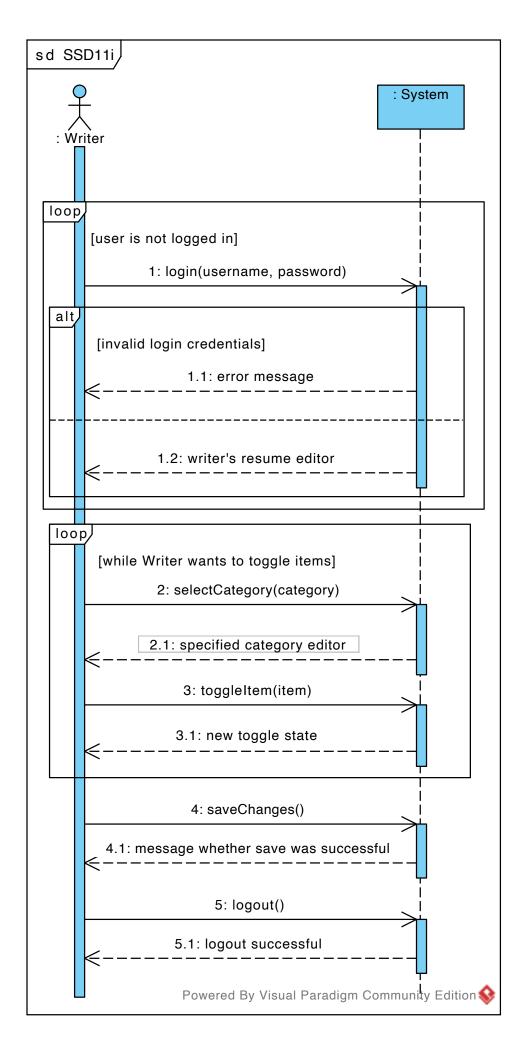


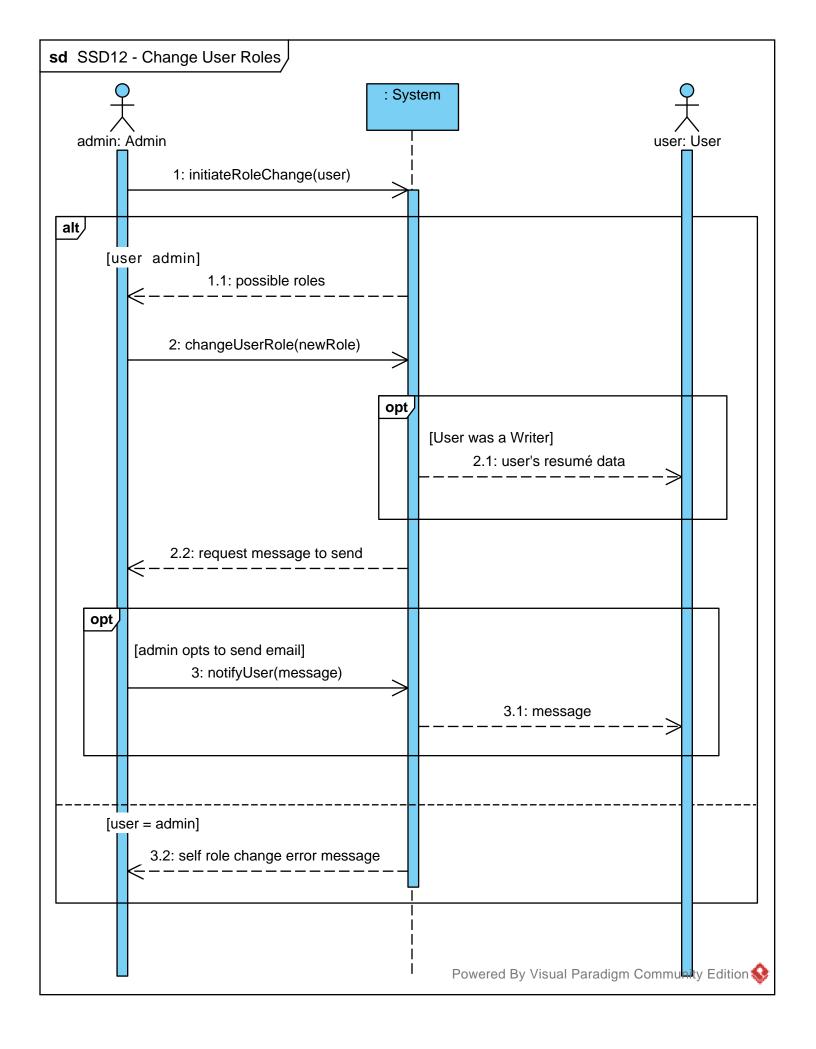


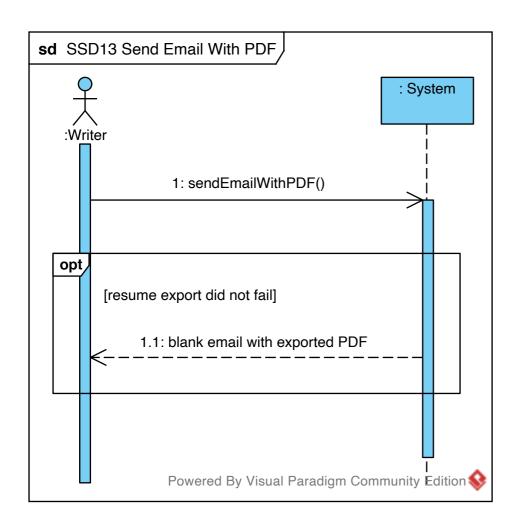






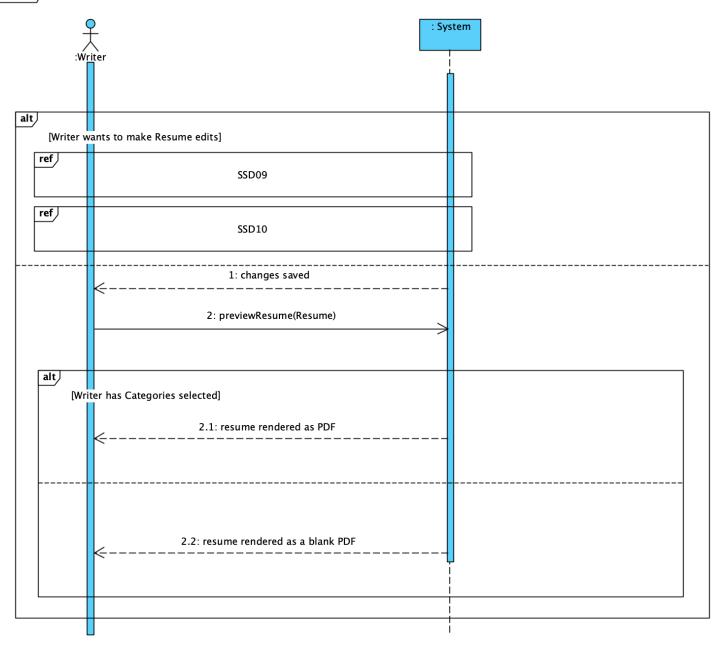


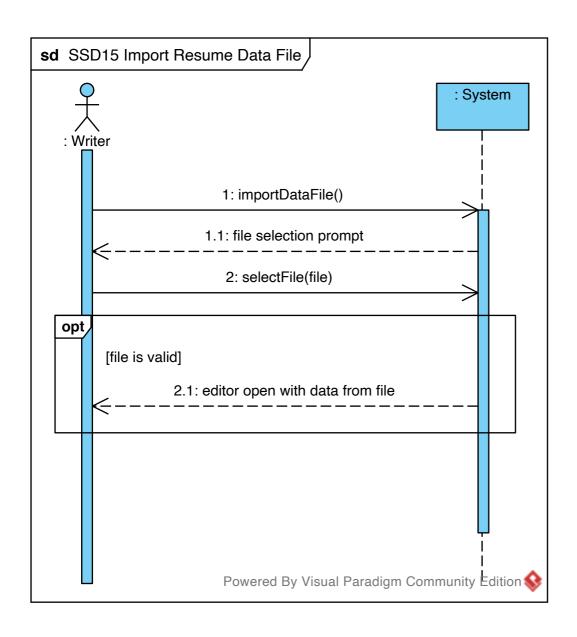


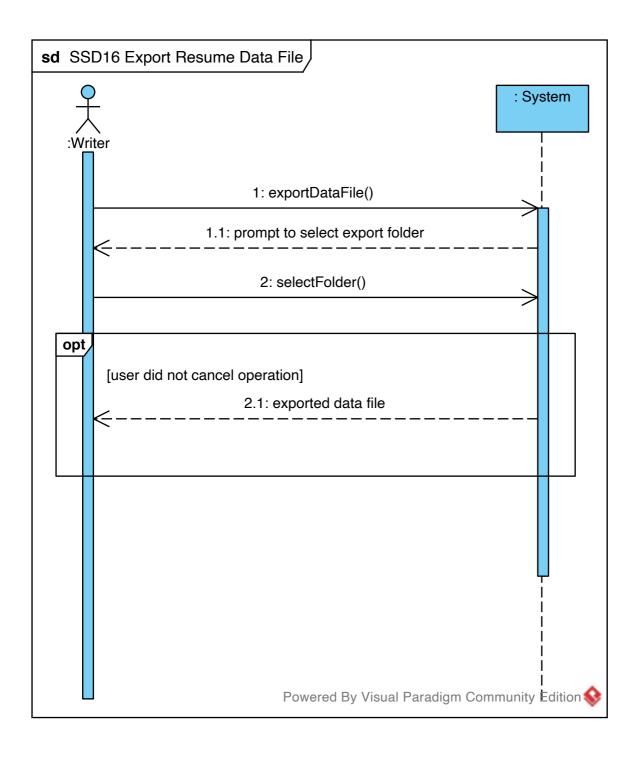


:Writer : System

sd SSD14







Operation Contracts

Contract OC01: createAccount

Operation: createAccount(email: String, password: String)

Cross References:

Use Cases: UC01 (Create Account)

Pre-Conditions:

- Application is opened.
- Text fields are filled.

Post-Conditions:

- The input was checked for validity.
- The new user account was created with the specified *email* and *password*.

Contract OC02: createResumeReview

Operation: createResumeReview()

Cross References:

• Use Cases: UC02 (Request Review)

Pre-Conditions:

 Writer has a Resume file with some entered categories/fields/items and has prompted System for a review.

Post-Conditions:

- A ReviewResume rr has been created.
- rr has been populated with links to the active categories/fields/items from the Writer's resume data file.
- rr has been written to the filesystem.

Contract OC03: submitReview

Operation: submitReview(review : Review)

Cross References:

• Use Cases: UC03 (Write Review Comments)

Pre-Conditions:

 A resumé exists to be reviewed by a Reviewer and the Reviewer is logged into their account on the System

- A new Resume Review File has been created.
- An Association between the Resume and the Review has been created to ensure that the comments will load in when the Writer logs into their account.

Contract OC04: login

Operation: login(username : String, password : String)

Cross References:

 Use Cases: UC04 (Edit Items/Fields), UC09 (Add/Remove Items/Fields), UC11 (Toggle Items/Fields)

Pre-Conditions:

Application is open to sign-in window.

Post-Conditions:

- Sign-in window was closed.
- Editor, User Manager, or Reviewer Tool was opened, depending on the role of the logging in user.

Contract OC05: changeCategoryName

Operation: changeCategoryName(category : Category, name: String)

Cross References:

• Use Cases: UC05 (Create/Remove Category)

Pre-Conditions:

- The user is logged in.
- category exists and is open in the editor.

Post-Conditions:

- category's name was changed to name.
- The Editor view was updated to reflect the change.

Contract OC06: cancelExport

Operation: cancelExport()

Cross References:

• Use Cases: UC06 (Export Resume)

Pre-Conditions:

Writer has a Resume export in progress.

- Exported resume file, if created, has been deleted.
- Lingering preview resume, if created, has been deleted.
- Resume export mode has been cleared.
- System has exited resume export mode.

Contract OC08: loadComments

Operation: loadComments()

Cross References:

• Use Cases: UC08 (Check Review Comments)

Pre-Conditions:

 An association exists between the Resume being opened and a Resume Review File.

Post-Conditions:

• Comment content has been added to various Items on the Writer's Resumé so that when the Item is opened, the content will load with the Item.

Contract OC09: createItem

Operation: createItem(category: Category)

Cross References:

Use Cases: UC09 (Add/Remove Items/Fields)

Pre-Conditions:

- User has logged in successfully.
- category exists in the Writer's resume and is currently selected.

Post-Conditions:

- The Editor window was updated to display the new item.
- The new Item has been created.

Contract OC10: setCategoryType

Operation: setCategoryType(type : CategoryType)

Cross References:

Use Cases: UC05 (Create/Remove Category)

Pre-Conditions:

- The user is logged in.
- *type* is a recognized type.

- The category type of the current category was changed to *type*.
- All current information was deleted from the category.

Contract OC11: editItem

Operation: editItem(item: Item, changes: ItemChanges)

Cross References:

• Use Cases: UC04 (Edit Items/Fields), UC09 (Add/Remove Items/Fields)

Pre-Conditions:

- *item* is selected and open in the Editor.
- item exists.
- Editor is in edit mode.

Post-Conditions:

- Editor recorded the specified *changes* to the *item*.
- Editor prepared changes to be stored.
- Editor was updated to reflect changes.

Contract OC12: changeUserRole

Operation: changeUserRole(newRole : UserRole)

Cross References:

• Use Cases: UC12 (Change User Role)

Pre-Conditions:

 Admin is signed in and attempting to change the role of a known user (called 'user').

Post-Conditions:

- user's role was set to newRole.
- If user was a Writer, the writer's Resume Data file was sent to them by email and the Resume Data file was deleted.
- If the User was made a Writer, a new empty Resume Data file was created for the user.

Contract OC13: sendEmailWithPDF

Operation: sendEmailWithPDF()

Cross References:

Use Cases: UC13 (Send Email With PDF)

Pre-Conditions:

• The Writer is signed into the system.

- The Writer's Resume PDF was created.
- The Resume PDF was attached to a blank email.

Contract OC14: previewResume

Operation: previewResume(resume : Resume)

Cross References:

• Use Cases: UC14 (Preview Resume)

Pre-Conditions:

 The Resume exists and the Writer has requested to preview their Resume as a PDF.

Post-Conditions:

 A Resume PDF File was created from the extracted data information from resume.

Contract OC15: importDataFile

Operation: importDataFile(file : File)

Cross References:

Use Cases: UC15 (Import Resume Data File)

Pre-Conditions:

• The Writer is signed into the system.

Post-Conditions:

- The Writer's Resume Data was collected from file.
- ResumeData instance r was created.
- r was filled with collected data.
- Editor instance e was created.
- e was associated with r.
- r is opened in e.

Contract OC16: exportDataFile

Operation: exportDataFile(folder : File)

Cross References:

• Use Cases: UC16 (Export Resume Data File)

Pre-Conditions:

- The Writer is signed into the system.
- ResumeData instance r exists.

- The Writer's Resume Data is collected from r.
- The Writer's collected Resume Data is written to a new file in the specified folder.

Gantt Diagram

Resume Builder

http://resume-builder.us.to/

Project managerAshley Couch (Team Leader)Project datesJan 30, 2020 - Apr 30, 2020

Completion13%Tasks31Resources5

2

Tasks

Iteration 1	
Requirements 1/30/20 2/1/20 Use Cases 2/3/20 2/6/20 Wireframes 2/6/20 2/6/20 System Sequence Diagrams 2/6/20 2/8/20 System Operation Contracts 2/10/20 2/10/20 Website 2/10/20 2/10/20 Traceability Matrix 2/10/20 2/10/20 Iteration 2 2/12/20 3/23/20 Diagrams 1 2/12/20 2/14/20 Implementation 1 2/14/20 2/22/20 Testing 1 2/22/20 2/23/20 Diagrams 2 2/24/20 2/26/20 Implementation 2 2/27/20 3/6/20 Testing 2 3/7/20 3/9/20 Diagrams 3 3/10/20 3/11/20 Implementation 3 3/12/20 3/22/20 Update Website 3/23/20 3/23/20 Iteration 3 3/25/20 4/29/20 Diagrams 1 3/25/20 3/27/20 Implementation 1 3/27/20 4/4/20 Implementation 2 4/6/20 4/8/20 Implementation 2 4/6/20	
Use Cases 2/3/20 2/6/20 Wireframes 2/6/20 2/6/20 System Sequence Diagrams 2/6/20 2/8/20 System Operation Contracts 2/10/20 2/10/20 Website 2/10/20 2/10/20 Traceability Matrix 2/10/20 2/10/20 Iteration 2 2/12/20 3/23/20 Diagrams 1 2/12/20 2/14/20 Implementation 1 2/14/20 2/22/20 Testing 1 2/22/20 2/23/20 Implementation 2 2/24/20 2/26/20 Implementation 2 2/27/20 3/6/20 Testing 2 3/7/20 3/9/20 Diagrams 3 3/10/20 3/11/20 3/20/20 Testing 3 3/21/20 3/22/20 Update Website 3/23/20 3/23/20 Iteration 3 3/25/20 3/27/20 Implementation 1 3/25/20 3/27/20 Implementation 1 3/27/20 4/4/20 Testing 1 4/4/20 4/5/20 Implementation 2 4/8/20 4/8/20	
Wireframes 2/6/20 2/6/20 System Sequence Diagrams 2/6/20 2/8/20 System Operation Contracts 2/10/20 2/10/20 Website 2/10/20 2/10/20 Traceability Matrix 2/10/20 2/10/20 Iteration 2 2/12/20 3/23/20 Diagrams 1 2/12/20 2/14/20 Implementation 1 2/14/20 2/22/20 Testing 1 2/22/20 2/23/20 Diagrams 2 2/24/20 2/26/20 Implementation 2 2/27/20 3/6/20 Testing 2 3/7/20 3/9/20 Diagrams 3 3/10/20 3/11/20 Implementation 3 3/12/20 3/20/20 Testing 3 3/21/20 3/22/20 Update Website 3/23/20 3/23/20 Iteration 3 3/25/20 4/29/20 Diagrams 1 3/25/20 3/27/20 Implementation 1 3/27/20 4/4/20 Testing 1 4/4/20 4/5/20 Diagrams 2 4/6/20 4/8/20 Implementation 2 4/8/20	
System Sequence Diagrams 2/6/20 2/8/20 System Operation Contracts 2/10/20 2/10/20 Website 2/10/20 2/10/20 Traceability Matrix 2/10/20 2/10/20 Iteration 2 2/12/20 3/23/20 Diagrams 1 2/12/20 2/14/20 Implementation 1 2/14/20 2/22/20 Testing 1 2/22/20 2/23/20 Diagrams 2 2/24/20 2/26/20 Implementation 2 2/27/20 3/6/20 Testing 2 3/7/20 3/9/20 Diagrams 3 3/10/20 3/11/20 Implementation 3 3/12/20 3/20/20 Testing 3 3/21/20 3/22/20 Update Website 3/23/20 3/23/20 Iteration 3 3/25/20 4/29/20 Diagrams 1 3/25/20 3/27/20 Implementation 1 3/27/20 4/4/20 Testing 1 4/4/20 4/5/20 Diagrams 2 4/6/20 4/8/20 Implementation 2 4/8/20 4/16/20	
System Operation Contracts 2/10/20 2/10/20 Website 2/10/20 2/10/20 Traceability Matrix 2/10/20 2/10/20 Iteration 2 2/12/20 3/23/20 Diagrams 1 2/12/20 2/14/20 Implementation 1 2/14/20 2/22/20 Testing 1 2/22/20 2/23/20 Diagrams 2 2/24/20 2/26/20 Implementation 2 2/27/20 3/6/20 Testing 2 3/7/20 3/9/20 Diagrams 3 3/10/20 3/11/20 Implementation 3 3/12/20 3/20/20 Testing 3 3/21/20 3/22/20 Update Website 3/23/20 3/23/20 Iteration 3 3/25/20 4/29/20 Diagrams 1 3/25/20 3/27/20 Implementation 1 3/27/20 4/4/20 Testing 1 4/4/20 4/5/20 Diagrams 2 4/6/20 4/8/20 Implementation 2 4/8/20 4/16/20	
Website 2/10/20 2/10/20 Traceability Matrix 2/10/20 2/10/20 Iteration 2 2/12/20 3/23/20 Diagrams 1 2/12/20 2/14/20 Implementation 1 2/14/20 2/22/20 Testing 1 2/22/20 2/23/20 Diagrams 2 2/24/20 2/26/20 Implementation 2 2/27/20 3/6/20 Testing 2 3/7/20 3/9/20 Diagrams 3 3/10/20 3/11/20 Implementation 3 3/12/20 3/20/20 Testing 3 3/21/20 3/22/20 Update Website 3/23/20 3/23/20 Iteration 3 3/25/20 4/29/20 Diagrams 1 3/25/20 3/27/20 Implementation 1 3/27/20 4/4/20 Testing 1 4/4/20 4/5/20 Diagrams 2 4/6/20 4/8/20 Implementation 2 4/8/20 4/16/20	
Traceability Matrix 2/10/20 2/10/20 Iteration 2 2/12/20 3/23/20 Diagrams 1 2/12/20 2/14/20 Implementation 1 2/14/20 2/22/20 Testing 1 2/22/20 2/23/20 Diagrams 2 2/24/20 2/26/20 Implementation 2 2/27/20 3/6/20 Testing 2 3/7/20 3/9/20 Diagrams 3 3/10/20 3/11/20 Implementation 3 3/12/20 3/20/20 Testing 3 3/21/20 3/22/20 Update Website 3/23/20 3/23/20 Iteration 3 3/25/20 4/29/20 Diagrams 1 3/25/20 4/29/20 Implementation 1 3/27/20 4/4/20 Testing 1 4/4/20 4/5/20 Diagrams 2 4/6/20 4/8/20 Implementation 2 4/8/20 4/16/20	
Iteration 2 2/12/20 3/23/20 Diagrams 1 2/12/20 2/14/20 Implementation 1 2/14/20 2/22/20 Testing 1 2/22/20 2/23/20 Diagrams 2 2/24/20 2/26/20 Implementation 2 2/27/20 3/6/20 Testing 2 3/7/20 3/9/20 Diagrams 3 3/10/20 3/11/20 Implementation 3 3/12/20 3/20/20 Testing 3 3/21/20 3/22/20 Update Website 3/23/20 3/23/20 Iteration 3 3/25/20 4/29/20 Diagrams 1 3/25/20 4/29/20 Implementation 1 3/27/20 4/4/20 Testing 1 4/4/20 4/5/20 Diagrams 2 4/6/20 4/8/20 Implementation 2 4/8/20 4/16/20	
Diagrams 1 2/12/20 2/14/20 Implementation 1 2/14/20 2/22/20 Testing 1 2/22/20 2/23/20 Diagrams 2 2/24/20 2/26/20 Implementation 2 2/27/20 3/6/20 Testing 2 3/7/20 3/9/20 Diagrams 3 3/10/20 3/11/20 Implementation 3 3/12/20 3/20/20 Testing 3 3/21/20 3/22/20 Update Website 3/23/20 3/23/20 Iteration 3 3/25/20 4/29/20 Diagrams 1 3/25/20 3/27/20 Implementation 1 3/27/20 4/4/20 Testing 1 4/4/20 4/5/20 Diagrams 2 4/6/20 4/8/20 Implementation 2 4/8/20 4/16/20	
Implementation 1 2/14/20 2/22/20 Testing 1 2/22/20 2/23/20 Diagrams 2 2/24/20 2/26/20 Implementation 2 2/27/20 3/6/20 Testing 2 3/7/20 3/9/20 Diagrams 3 3/10/20 3/11/20 Implementation 3 3/12/20 3/20/20 Testing 3 3/21/20 3/22/20 Update Website 3/23/20 3/23/20 Iteration 3 3/25/20 4/29/20 Diagrams 1 3/25/20 3/27/20 Implementation 1 3/27/20 4/4/20 Testing 1 4/4/20 4/5/20 Diagrams 2 4/6/20 4/8/20 Implementation 2 4/8/20 4/16/20	
Testing 1 2/22/20 2/23/20 Diagrams 2 2/24/20 2/26/20 Implementation 2 2/27/20 3/6/20 Testing 2 3/7/20 3/9/20 Diagrams 3 3/10/20 3/11/20 Implementation 3 3/12/20 3/20/20 Testing 3 3/21/20 3/22/20 Update Website 3/23/20 3/23/20 Iteration 3 3/25/20 4/29/20 Diagrams 1 3/25/20 3/27/20 Implementation 1 3/27/20 4/4/20 Testing 1 4/4/20 4/5/20 Diagrams 2 4/6/20 4/8/20 Implementation 2 4/8/20 4/16/20	
Diagrams 2 2/24/20 2/26/20 Implementation 2 2/27/20 3/6/20 Testing 2 3/7/20 3/9/20 Diagrams 3 3/10/20 3/11/20 Implementation 3 3/12/20 3/20/20 Testing 3 3/21/20 3/22/20 Update Website 3/23/20 3/23/20 Iteration 3 3/25/20 4/29/20 Diagrams 1 3/25/20 3/27/20 Implementation 1 3/27/20 4/4/20 Testing 1 4/4/20 4/5/20 Diagrams 2 4/6/20 4/8/20 Implementation 2 4/8/20 4/16/20	
Implementation 2 2/27/20 3/6/20 Testing 2 3/7/20 3/9/20 Diagrams 3 3/10/20 3/11/20 Implementation 3 3/12/20 3/20/20 Testing 3 3/21/20 3/22/20 Update Website 3/23/20 3/23/20 Iteration 3 3/25/20 4/29/20 Diagrams 1 3/25/20 3/27/20 Implementation 1 3/27/20 4/4/20 Testing 1 4/4/20 4/5/20 Diagrams 2 4/6/20 4/8/20 Implementation 2 4/8/20 4/16/20	
Testing 2 3/7/20 3/9/20 Diagrams 3 3/10/20 3/11/20 Implementation 3 3/12/20 3/20/20 Testing 3 3/21/20 3/22/20 Update Website 3/23/20 3/23/20 Iteration 3 3/25/20 4/29/20 Diagrams 1 3/25/20 3/27/20 Implementation 1 3/27/20 4/4/20 Testing 1 4/4/20 4/5/20 Diagrams 2 4/6/20 4/8/20 Implementation 2 4/8/20 4/16/20	
Diagrams 3 3/10/20 3/11/20 Implementation 3 3/12/20 3/20/20 Testing 3 3/21/20 3/22/20 Update Website 3/23/20 3/23/20 Iteration 3 3/25/20 4/29/20 Diagrams 1 3/25/20 3/27/20 Implementation 1 3/27/20 4/4/20 Testing 1 4/4/20 4/5/20 Diagrams 2 4/6/20 4/8/20 Implementation 2 4/8/20 4/16/20	
Implementation 3 3/12/20 3/20/20 Testing 3 3/21/20 3/22/20 Update Website 3/23/20 3/23/20 Iteration 3 3/25/20 4/29/20 Diagrams 1 3/25/20 3/27/20 Implementation 1 3/27/20 4/4/20 Testing 1 4/4/20 4/5/20 Diagrams 2 4/6/20 4/8/20 Implementation 2 4/8/20 4/16/20	
Testing 3 3/21/20 3/22/20 Update Website 3/23/20 3/23/20 Iteration 3 3/25/20 4/29/20 Diagrams 1 3/25/20 3/27/20 Implementation 1 3/27/20 4/4/20 Testing 1 4/4/20 4/5/20 Diagrams 2 4/6/20 4/8/20 Implementation 2 4/8/20 4/16/20	
Update Website 3/23/20 3/23/20 Iteration 3 3/25/20 4/29/20 Diagrams 1 3/25/20 3/27/20 Implementation 1 3/27/20 4/4/20 Testing 1 4/4/20 4/5/20 Diagrams 2 4/6/20 4/8/20 Implementation 2 4/8/20 4/16/20	
Iteration 3 3/25/20 4/29/20 Diagrams 1 3/25/20 3/27/20 Implementation 1 3/27/20 4/4/20 Testing 1 4/4/20 4/5/20 Diagrams 2 4/6/20 4/8/20 Implementation 2 4/8/20 4/16/20	
Diagrams 1 3/25/20 3/27/20 Implementation 1 3/27/20 4/4/20 Testing 1 4/4/20 4/5/20 Diagrams 2 4/6/20 4/8/20 Implementation 2 4/8/20 4/16/20	
Implementation 1 3/27/20 4/4/20 Testing 1 4/4/20 4/5/20 Diagrams 2 4/6/20 4/8/20 Implementation 2 4/8/20 4/16/20	
Testing 1 4/4/20 4/5/20 Diagrams 2 4/6/20 4/8/20 Implementation 2 4/8/20 4/16/20	
Diagrams 2 4/6/20 4/8/20 Implementation 2 4/8/20 4/16/20	
Implementation 2 4/8/20 4/16/20	
·	
Testing 2 4/17/20 4/18/20	
Diagrams 3 4/19/20 4/21/20	
Implementation 3 4/21/20 4/27/20	
Testing 3 4/28/20 4/29/20	
Update Website 4/29/20 4/29/20	

Resume Builder Feb 11, 2020

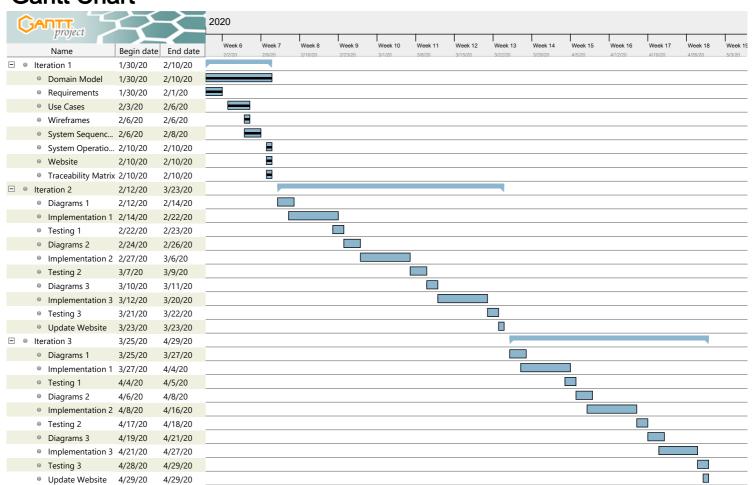
Resources

Name	Default role
Ashley Couch (Team Leader)	project manager
Matthew McCaskill	undefined
Jacob Curtis	undefined
Micah Schiewe	undefined
Brooklyn Stone	undefined

3

Gantt Chart

1 60 11, 2020



Resume Builder Feb 11, 2020

Resources Chart

GANTT.	92	2020													
Name	Default role	Week 6	Week 7	Week 8 2/16/20	Week 9 2/23/20	Week 10 3/1/20	Week 11 3/8/20	Week 12 3/15/20	Week 13 3/22/20	Week 14 3/29/20	Week 15 4/5/20	Week 16 4/12/20	Week 17 4/19/20	Week 18 4/26/20	Week 19 5/3/20
Ashley Couch (Team Leader)	project man	2/2/20	2/9/20	2/16/20	2/23/20	3/1/20	3/8/20	3/15/20	3/22/20	3/29/20	4/5/20	4/12/20	4/19/20	4/26/20	5/3/20
Matthew McCaskill	undefined														
Jacob Curtis	undefined														
Micah Schiewe	undefined														
Brooklyn Stone	undefined														

5

Time Sheet

Time Sheet										
Team Member	Total	Misc. Total	Meeting 1	Use Cases	Meeting 2	System Sequence Diagrams	Meeting 3	Operation Contracts	Meeting 4	Post-Presentation Fixe
Matthew McCaskill	19	1	1.5	1.5	5	1.5	3	0.5	2.5	2.5
Ashley Lu Couch	18.75	0	1.5	1.5	5	1.5	3	0.75	2.5	3
Micah Schiewe	20.5	2.75	1.5	1.5	5	2	3	1	2.5	1.25
Jacob Curtis	19.25	2.75	1.5	1.5	5	1.5	3	0.5	2.5	1
Brooklynn Stone	17	1	1.5	1.5	5	1.5	3	0.5	2.5	0.5
	Misc	cellaneous Entries					• N	//atthew McCaskill		
Team Member	Hours	De	scription				<u> </u>	shley Lu Couch		
Matthew McCaskill	1	Website								
Micah Schiewe	2	Java Tests/Prototyp	es					flicah Schiewe		
Micah Schiewe	0.75	Domain Model						acob Curtis		
Jacob Curtis	0.75	Domain Model					• B	Brooklynn Stone		
Jacob Curtis	1.5	Gantt Diagram								
Brooklynn Stone	1	Use Cases and Req	uirements							
Jacob Curtis	0.5	Requirements								
						Average Hours:	18.90			
						Standard Deviation:	1.26			

Point Redistribution

Point Redistribution

As a group,	, we decided t	hat we equally	/ distributed	our time	towards the	e tasks r	equired	in this
iteration ar	nd that no poi	nt redistribution	on is necessa	ary.				

Presentation

RESUMÉ REVIEW SOFTWARE

AGENDA

- Group + Project Intro
- Project Overview
 - Requirements
 - Domain Model
- Project Specifics
 - Use Cases
 - UC02 Request Review
 - UC05a/b Add/Remove Category
 - Wireframes
- Project Plan
 - Gantt Diagram
- Questions

GROUP + PROJECT INTRO

Relevant Persons:

- Ashley Lu Couch (Leader)
- Matthew McCaskill
- Micah Schiewe
- Jacob Curtis
- Brooklynn Stone

PROJECT OVERVIEW

Problem:

- Jobs require custom-tailored resumés
- Customizing resumés are a pain
- Reviewing resumés requires making an appointment

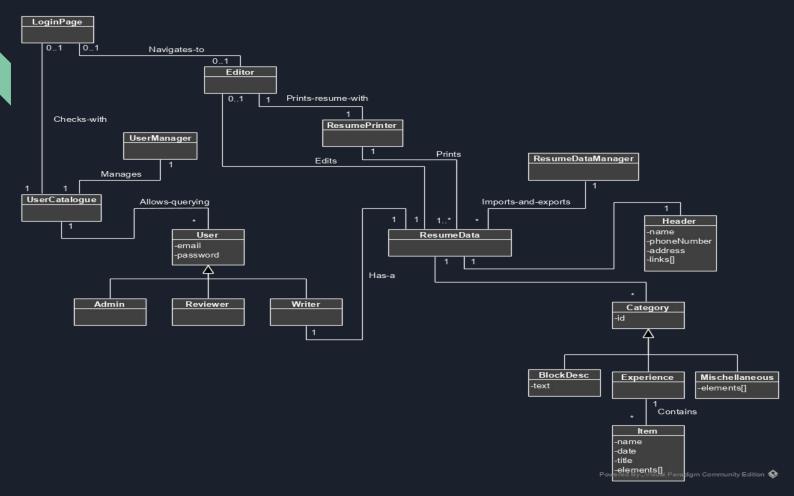
Solution:

 An application which handles customizing resumés and reviewing them in one place

PROJECT OVERVIEW

- Based off of the process of writing, revising, and reviewing a resumé from the Career Center
- Must accommodate:
 - Modify resume parts (R01)
 - Export Resume (R05)
 - Send review requests (R06)
 - Have comments (R07)
 - Toggle resume parts (R08)

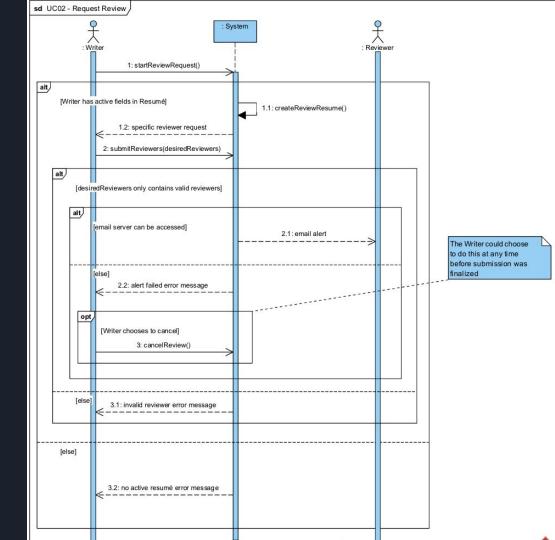
PROJECT OVERVIEW - DOMAIN MODEL



UC02 - Request Review

Stakeholder desires:

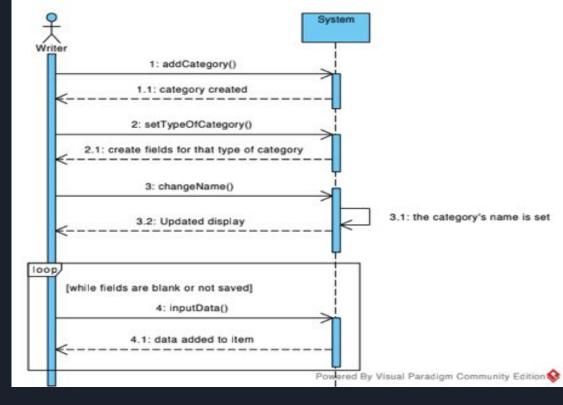
- Writer: Reliably notify specific reviewer his resumé is available to review.
- Reviewer: Be quickly notified of request to review resumé.
- Admin: Non-Reviewers are not contacted regarding a resumé review.



UC05a - Add Category

Stakeholder desires:

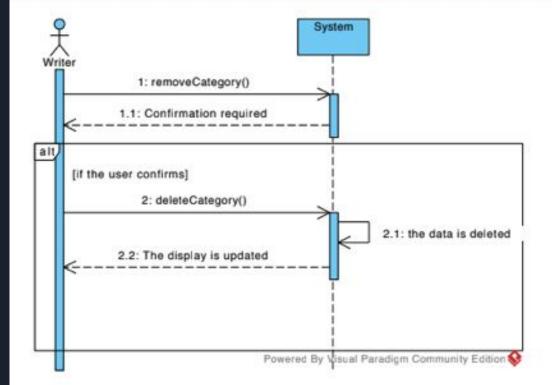
 Writer: Create new category they can add resume elements to.



UC05b - Remove Category

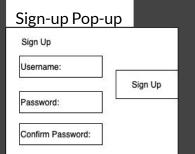
Stakeholder desires:

 Writer: Remove category they no longer want to be part of resume.



Wireframe

Login Page Login: Username: Password: Sign Up



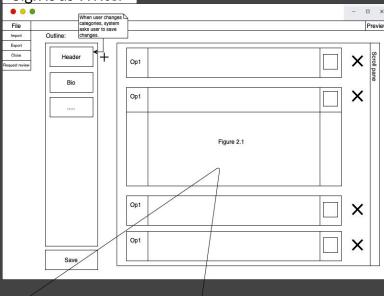
Sign is as Reviewer

Search by Name
Resume - User

Sign is as Admin

orginis as mannin							
Username	Role	Password					
\sim	\sim	\sim					
\sim	\sim	\sim					
\sim	\sim	\sim					
\sim	\sim	\sim					
\sim	\sim	\sim					
\sim	\sim	\sim					
\sim	\sim	\sim					

Sign is as Writer



Title Type Drop down (ex: job, experience, etc.)
Comments: Type Information:

PROJECT Plan

Gantt Diagram:

