Lab 3

COMP 354- Fall 23

Apekshaba Gohil Anam A. Shaikh Marzieh Adeli

apekshabagohil.it@gmail.com anams403@gmail.com marzieh.adeli@gmail.com

Quality Control

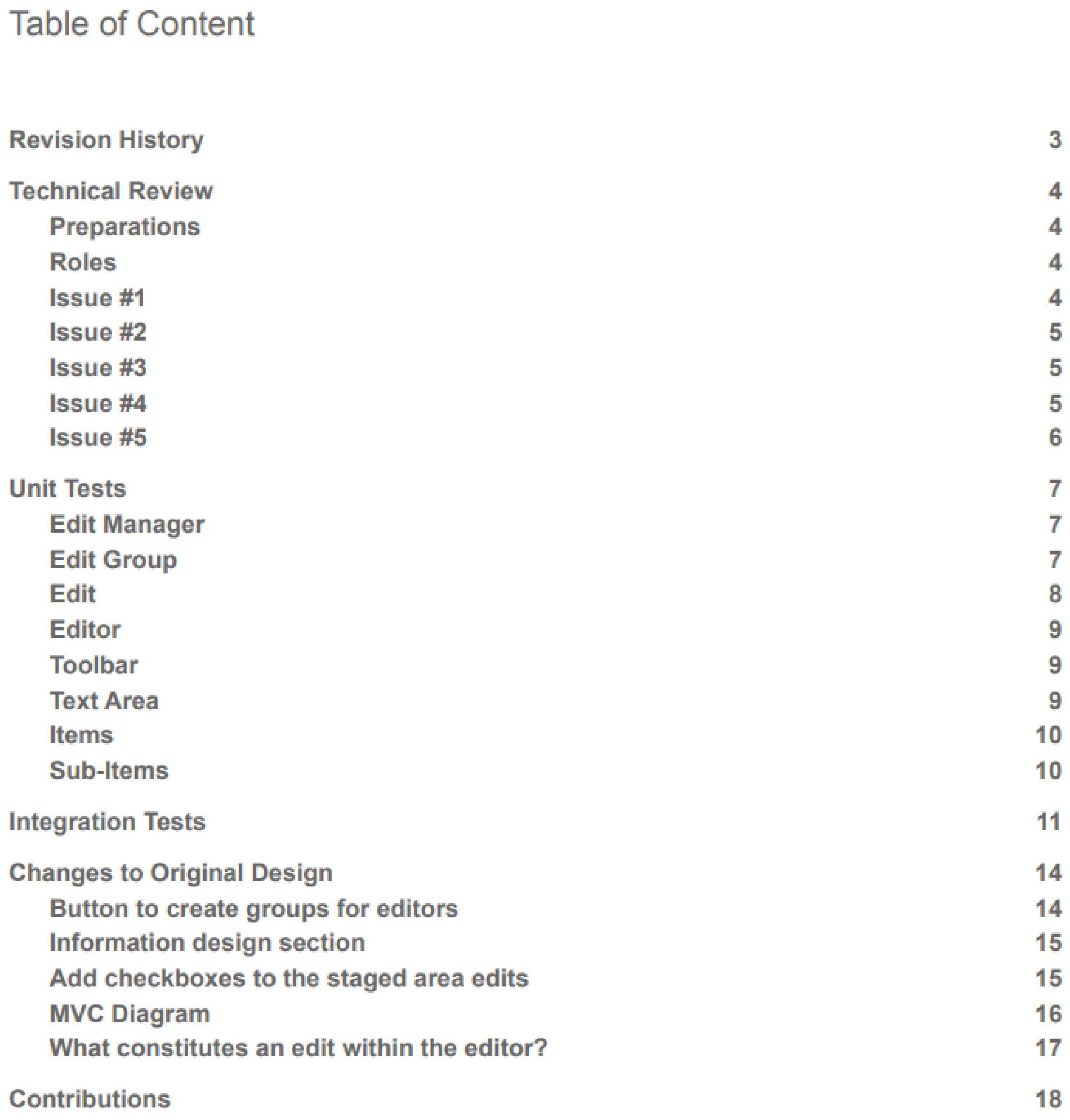
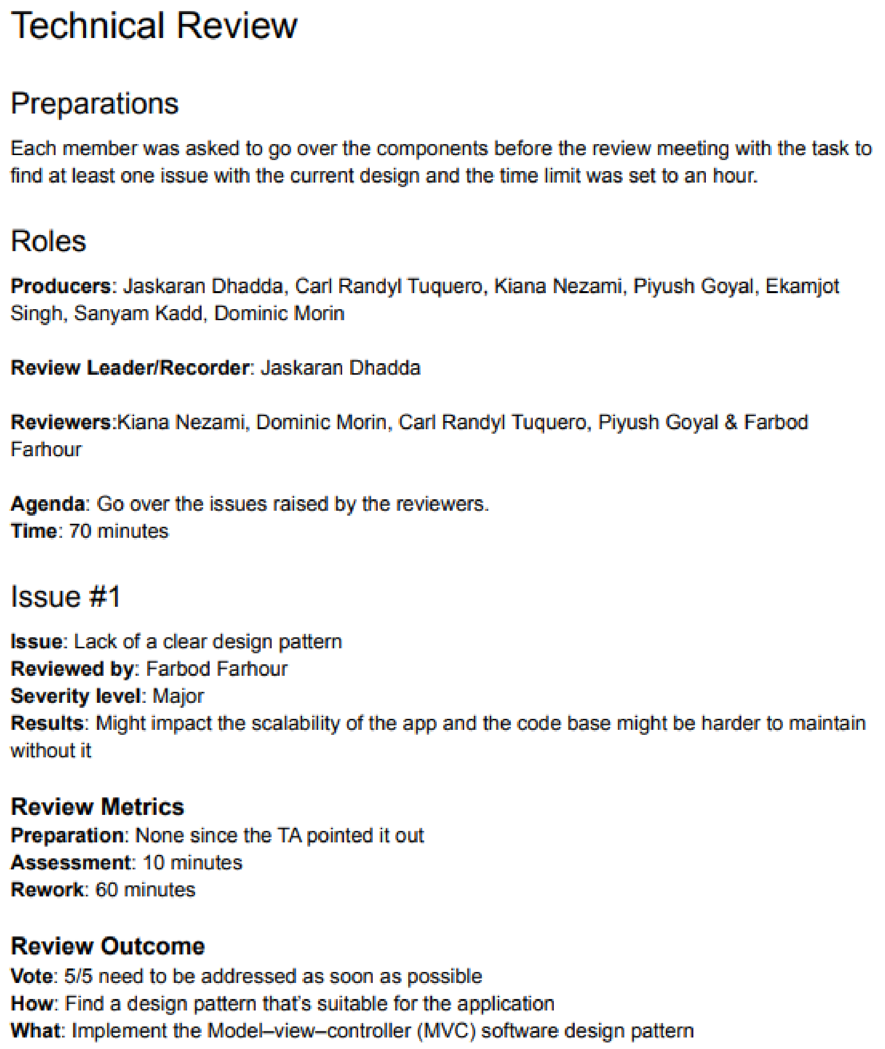


Table of Content

Technical Review



Technical Review

Preparations

Each member was asked to go over the components before the review meeting with the task to find at least one issue with the current design and the time limit was set to an hour.

Roles

Producers: Jaskaran Dhadda, Carl Randyl Tuquero, Kiana Nezami, Piyush Goyal, Ekamjot Singh, Sanyam Kadd, Dominic Morin

Review Leader/Recorder: Jaskaran Dhadda

Reviewers:Kiana Nezami, Dominic Morin, Carl Randyl Tuquero, Piyush Goyal & Farbod Farhour

Agenda: Go over the issues raised by the reviewers.
Time: 70 minutes

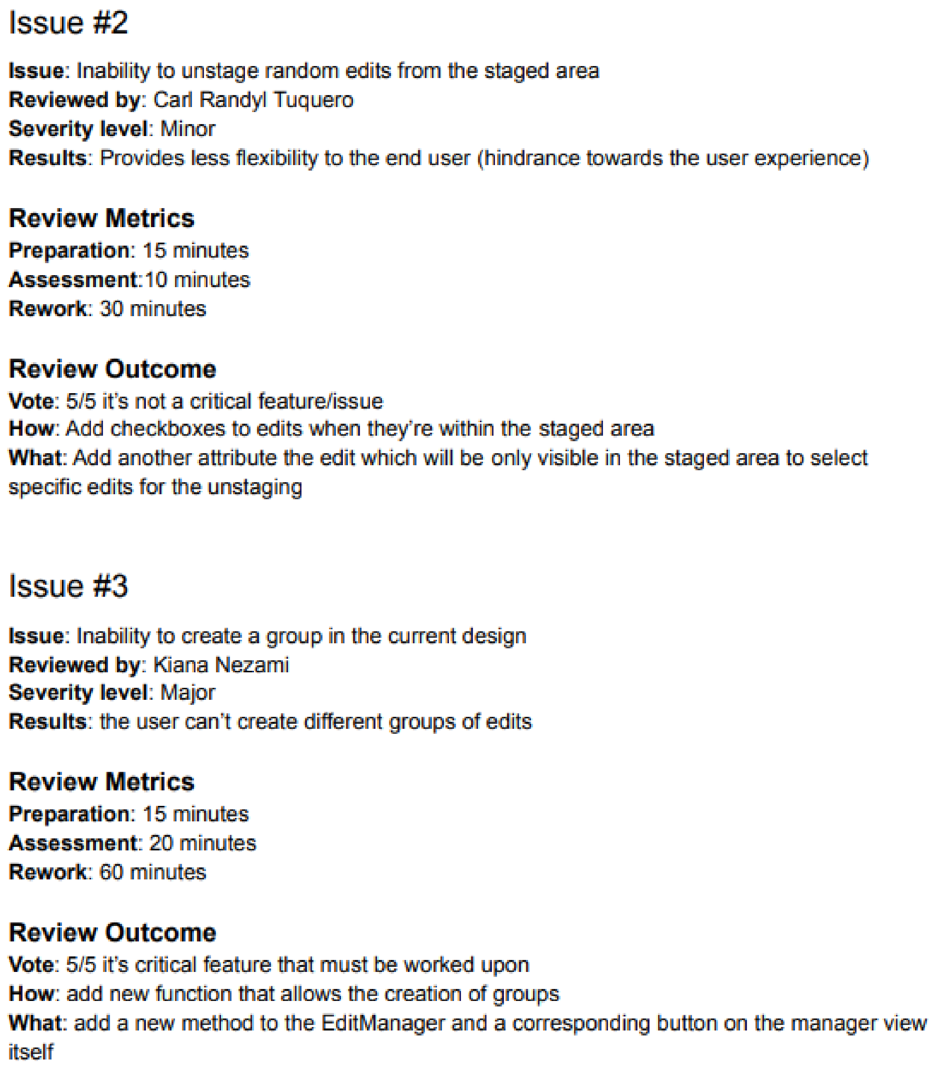
Issue #1

Issue: Lack of a clear design pattern
Reviewed by: Farbod Farhour
Severity level: Major
Results: Might impact the scalability of the app and the code base might be harder to maintain without it

Review Metrics
Preparation: None since the TA pointed it out
Assessment: 10 minutes
Rework: 60 minutes

Review Outcome
Vote: 5/5 need to be addressed as soon as possible
How: Find a design pattern that's suitable for the application
What: Implement the Model–view–controller (MVC) software design pattern

Technical Review



Issue #2

Issue: Inability to unstage random edits from the staged area
Reviewed by: Carl Randyl Tuquero
Severity level: Minor
Results: Provides less flexibility to the end user (hindrance towards the user experience)

Review Metrics

Preparation: 15 minutes
Assessment:10 minutes
Rework: 30 minutes

Review Outcome

Vote: 5/5 it's not a critical feature/issue
How: Add checkboxes to edits when they're within the staged area
What: Add another attribute the edit which will be only visible in the staged area to select specific edits for the unstaging

Issue #3

Issue: Inability to create a group in the current design
Reviewed by: Kiana Nezami
Severity level: Major
Results: the user can't create different groups of edits

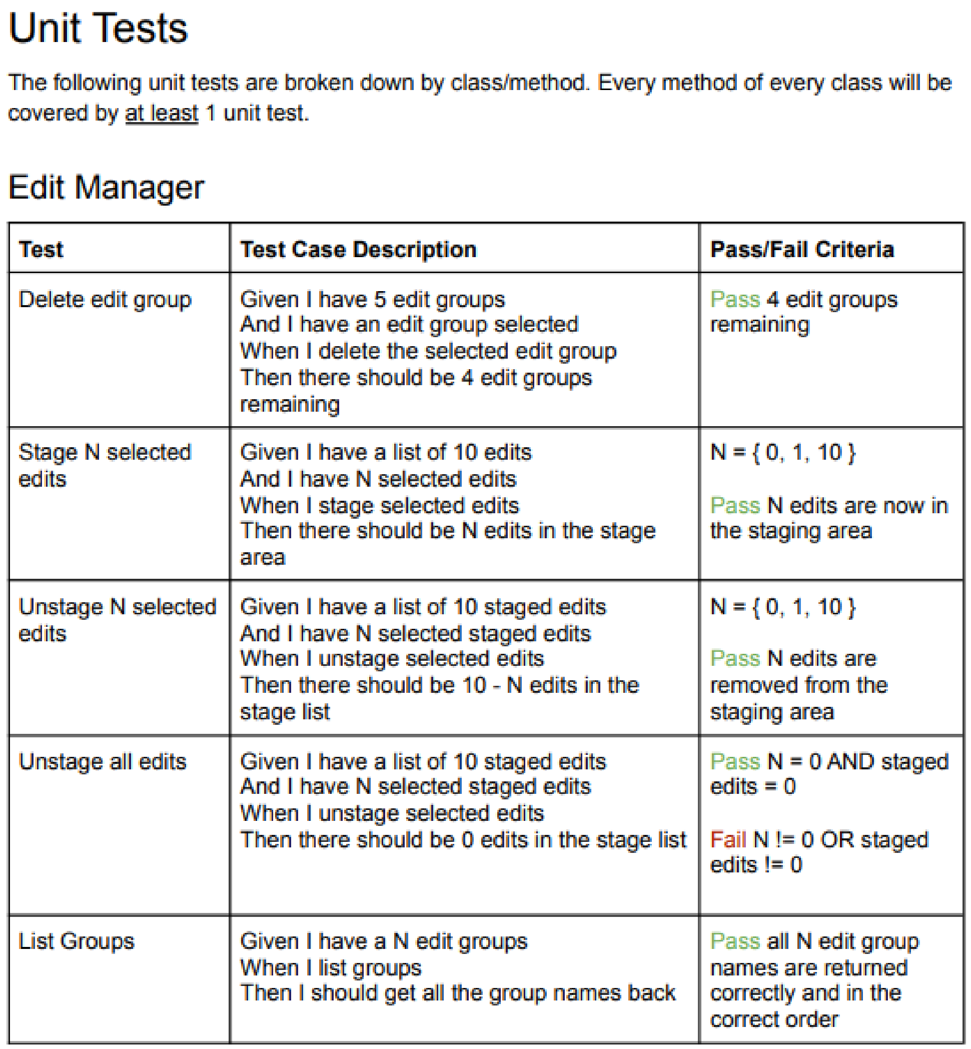
Review Metrics

Preparation: 15 minutes
Assessment: 20 minutes
Rework: 60 minutes

Review Outcome

Vote: 5/5 it's critical feature that must be worked upon
How: add new function that allows the creation of groups
What: add a new method to the EditManager and a corresponding button on the manager view itself

Unit Testing



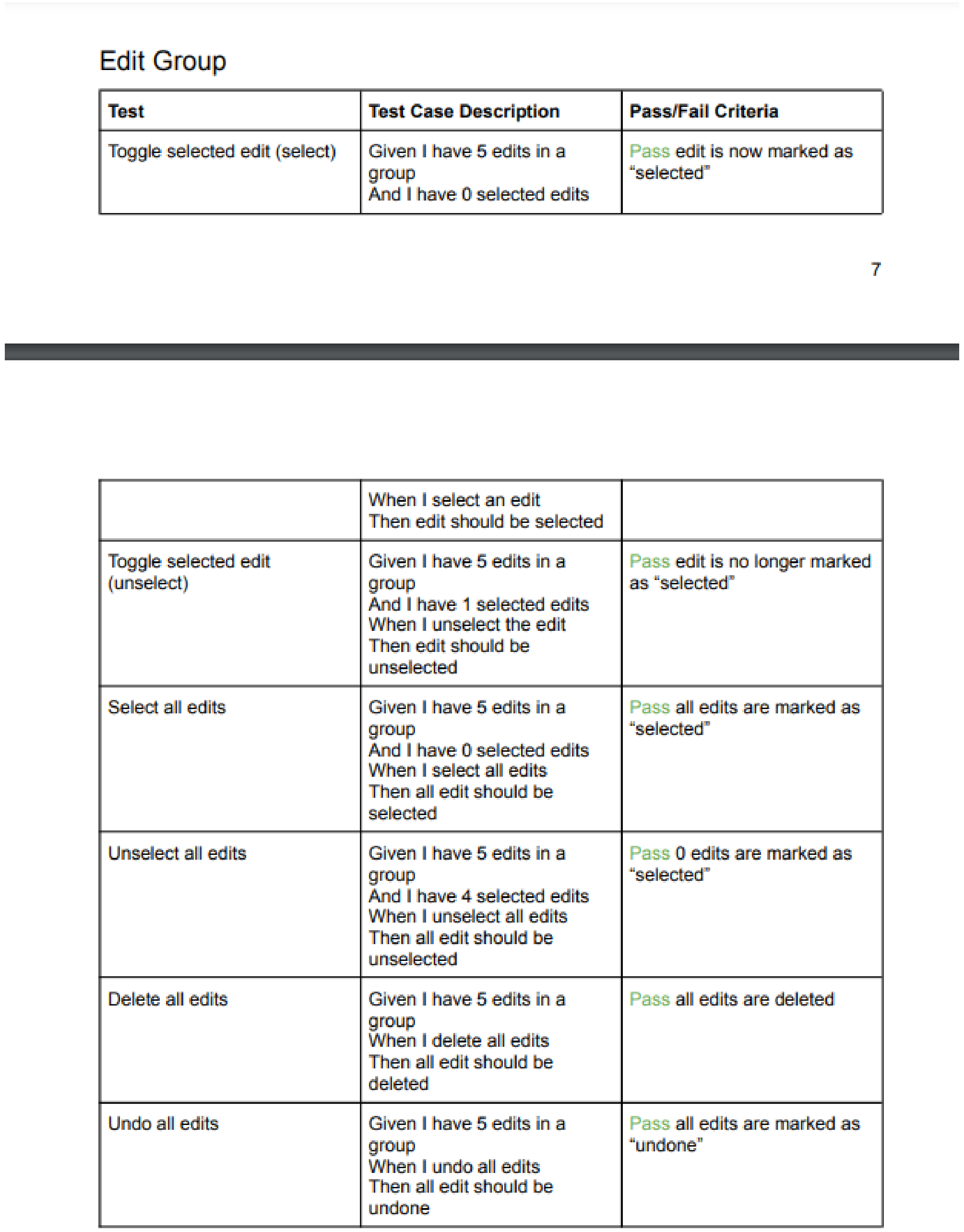
Unit Tests

The following unit tests are broken down by class/method. Every method of every class will be covered by at least 1 unit test.

Edit Manager

Test	Test Case Description	Pass/Fail Criteria
Delete edit group	Given I have 5 edit groups And I have an edit group selected When I delete the selected edit group Then there should be 4 edit groups remaining	Pass 4 edit groups remaining
Stage N selected edits	Given I have a list of 10 edits And I have N selected edits When I stage selected edits Then there should be N edits in the stage area	N = { 0, 1, 10 } Pass N edits are now in the staging area
Unstage N selected edits	Given I have a list of 10 staged edits And I have N selected staged edits When I unstage selected edits Then there should be 10 - N edits in the stage list	N = { 0, 1, 10 } Pass N edits are removed from the staging area
Unstage all edits	Given I have a list of 10 staged edits And I have N selected staged edits When I unstage selected edits Then there should be 0 edits in the stage list	Pass N = 0 AND staged edits = 0 Fail N != 0 OR staged edits != 0
List Groups	Given I have a N edit groups When I list groups Then I should get all the group names back	Pass all N edit group names are returned correctly and in the correct order

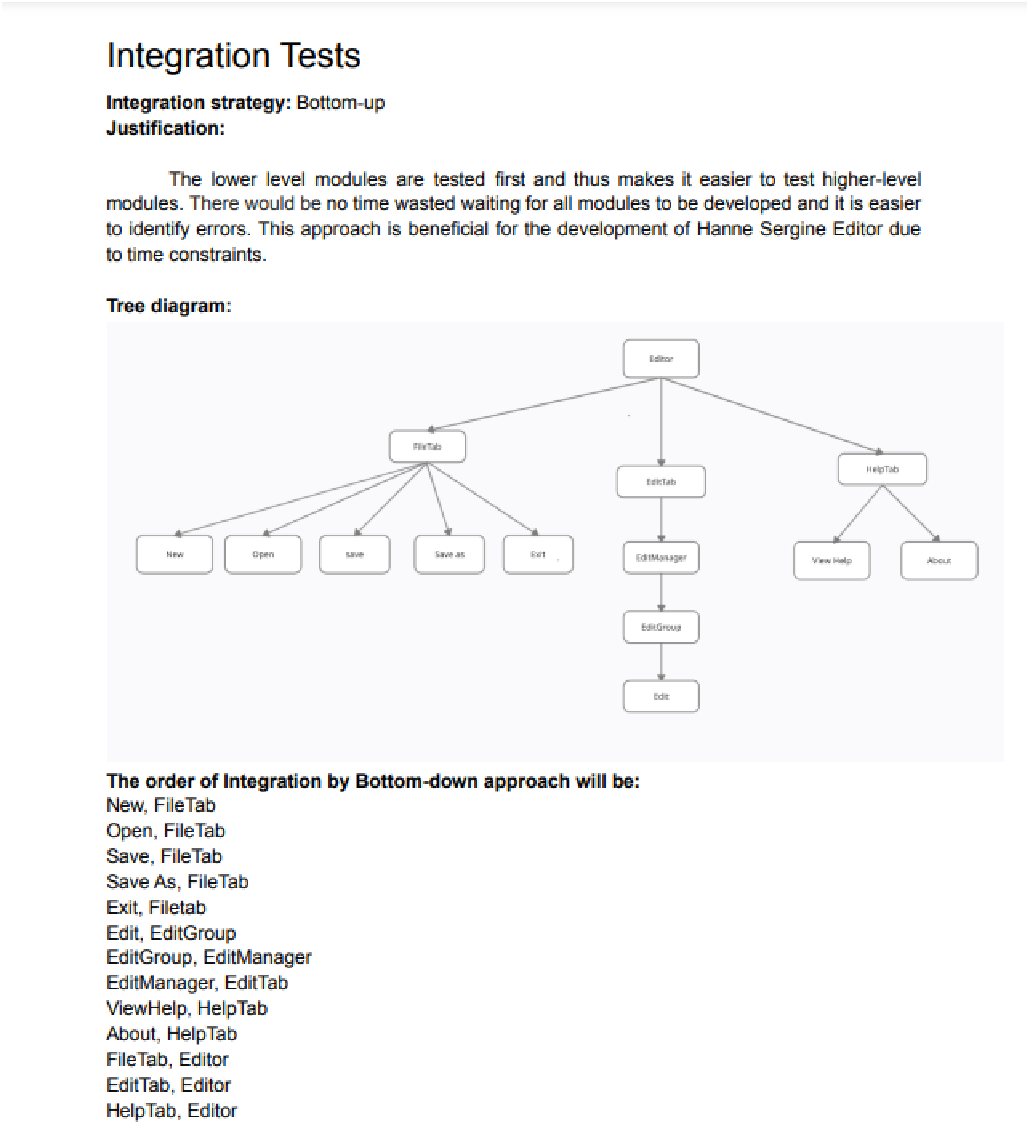
Unit Testing



Edit Group

Test	Test Case Description	Pass/Fail Criteria
Toggle selected edit (select)	Given I have 5 edits in a group And I have 0 selected edits When I select an edit Then edit should be selected	Pass edit is now marked as "selected"
Toggle selected edit (unselect)	Given I have 5 edits in a group And I have 1 selected edits When I unselect the edit Then edit should be unselected	Pass edit is no longer marked as "selected"
Select all edits	Given I have 5 edits in a group And I have 0 selected edits When I select all edits Then all edit should be selected	Pass all edits are marked as "selected"
Unselect all edits	Given I have 5 edits in a group And I have 4 selected edits When I unselect all edits Then all edit should be unselected	Pass 0 edits are marked as "selected"
Delete all edits	Given I have 5 edits in a group When I delete all edits Then all edit should be deleted	Pass all edits are deleted
Undo all edits	Given I have 5 edits in a group When I undo all edits Then all edit should be undone	Pass all edits are marked as "undone"

Integration Testing



Integration Tests

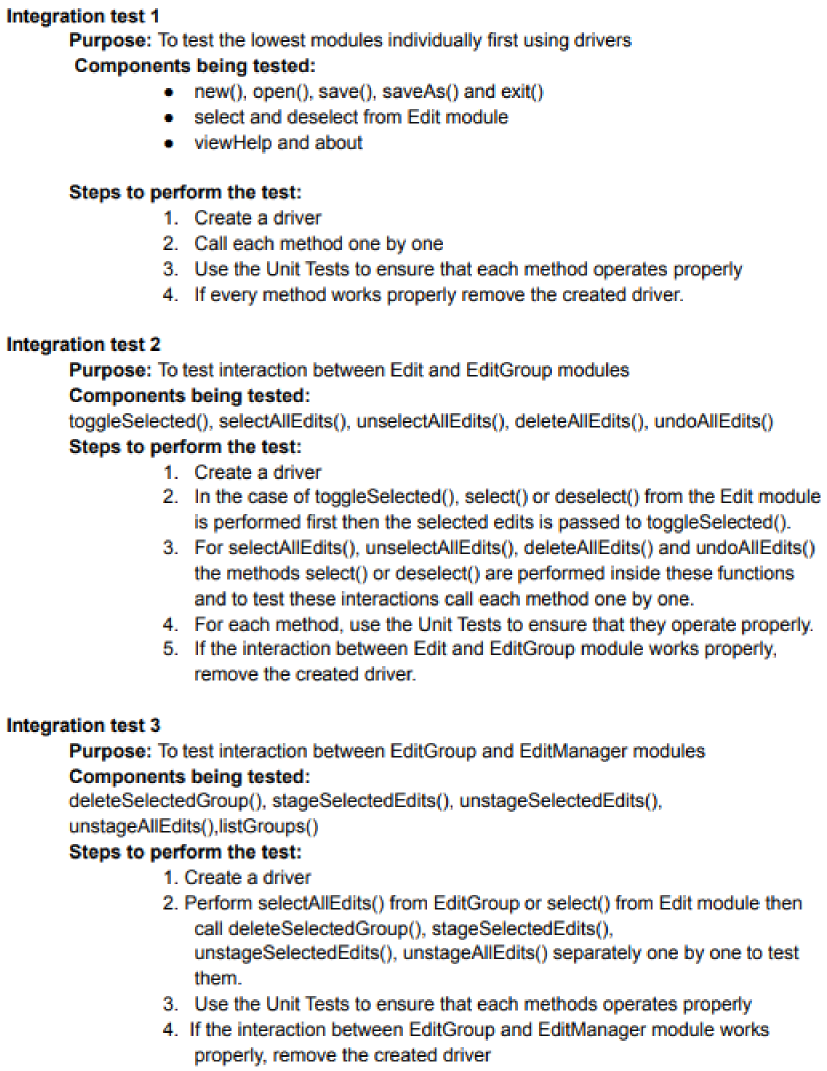
Integration strategy: Bottom-up
Justification:

The lower level modules are tested first and thus makes it easier to test higher-level modules. There would be no time wasted waiting for all modules to be developed and it is easier to identify errors. This approach is beneficial for the development of Hanne Sergine Editor due to time constraints.

Tree diagram:

The order of Integration by Bottom-down approach will be:
New, FileTab
Open, FileTab
Save, FileTab
Save As, FileTab
Exit, Filetab
Edit, EditGroup
EditGroup, EditManager
EditManager, EditTab
ViewHelp, HelpTab
About, HelpTab
FileTab, Editor
EditTab, Editor
HelpTab, Editor

Integration Testing



Integration test 1

Purpose: To test the lowest modules individually first using drivers

Components being tested:

- new(), open(), save(), saveAs() and exit()
- select and deselect from Edit module
- viewHelp and about

Steps to perform the test:

1. Create a driver
2. Call each method one by one
3. Use the Unit Tests to ensure that each method operates properly
4. If every method works properly remove the created driver.

Integration test 2

Purpose: To test interaction between Edit and EditGroup modules

Components being tested:

toggleSelected(), selectAllEdits(), unselectAllEdits(), deleteAllEdits(), undoAllEdits()

Steps to perform the test:

1. Create a driver
2. In the case of toggleSelected(), select() or deselect() from the Edit module is performed first then the selected edits is passed to toggleSelected().
3. For selectAllEdits(), unselectAllEdits(), deleteAllEdits() and undoAllEdits() the methods select() or deselect() are performed inside these functions and to test these interactions call each method one by one.
4. For each method, use the Unit Tests to ensure that they operate properly.
5. If the interaction between Edit and EditGroup module works properly, remove the created driver.

Integration test 3

Purpose: To test interaction between EditGroup and EditManager modules

Components being tested:

deleteSelectedGroup(), stageSelectedEdits(), unstageSelectedEdits(), unstageAllEdits(),listGroups()

Steps to perform the test:

1. Create a driver
2. Perform selectAllEdits() from EditGroup or select() from Edit module then call deleteSelectedGroup(), stageSelectedEdits(), unstageSelectedEdits(), unstageAllEdits() separately one by one to test them.
3. Use the Unit Tests to ensure that each methods operates properly
4. If the interaction between EditGroup and EditManager module works properly, remove the created driver

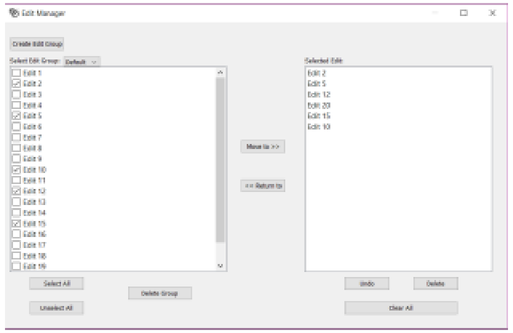
Changes to Original Design

Changes to Original Design

After the technical review in the first part, we came across a few cases that required changes in the original design.

Button to create groups for editors

In the last version, the user could not create different groups of edits thus, the newer version allows the user to do so.



Edit Manager

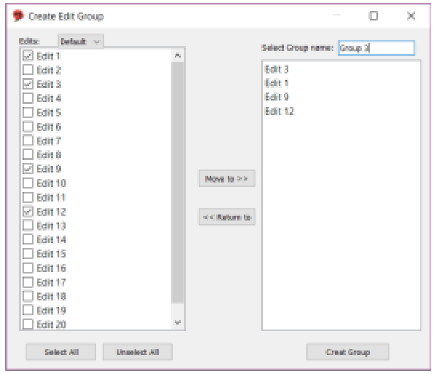
Select Edit Group: Default

☐ Edit 1
☑ Edit 2
☐ Edit 3
☐ Edit 4
☑ Edit 5
☐ Edit 6
☐ Edit 7
☐ Edit 8
☐ Edit 9
☑ Edit 10
☐ Edit 11
☑ Edit 12
☐ Edit 13
☐ Edit 14
☑ Edit 15
☐ Edit 16
☐ Edit 17
☐ Edit 18
☐ Edit 19

Selected Edit:

Edit 2
Edit 5
Edit 12
Edit 20
Edit 15
Edit 10

When the user clicks on the “Create Edit Group” button, a new page like below figure will be opened. In this page, a user can select the edits that he/she wants and move them to the next frame and place them in a group by selecting the group name.



Create Edit Group

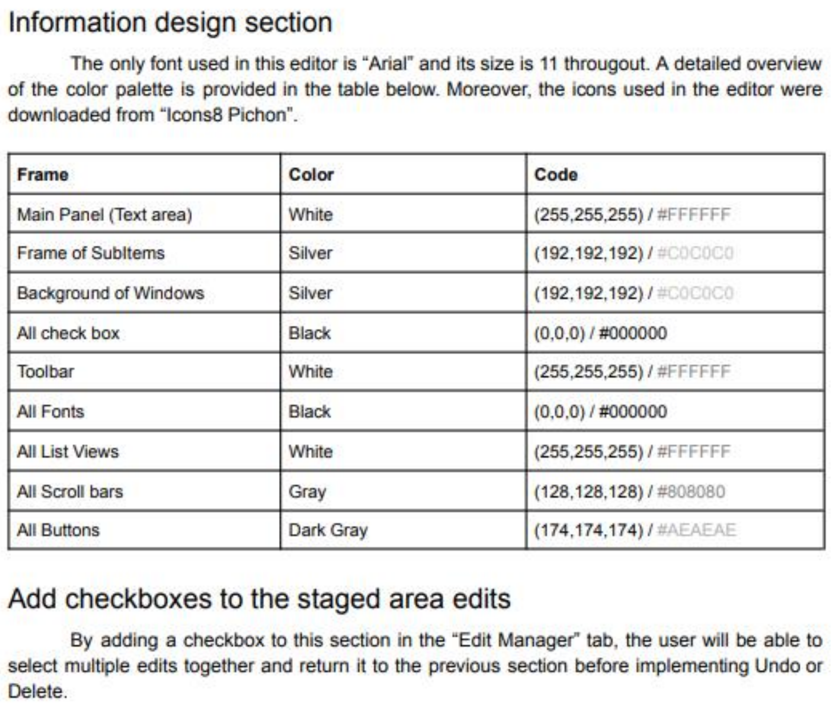
Edits: Default

☑ Edit 1
☐ Edit 2
☑ Edit 3
☐ Edit 4
☐ Edit 5
☐ Edit 6
☐ Edit 7
☐ Edit 8
☑ Edit 9
☐ Edit 10
☐ Edit 11
☑ Edit 12
☐ Edit 13
☐ Edit 14
☐ Edit 15
☐ Edit 16
☐ Edit 17
☐ Edit 18
☐ Edit 19
☐ Edit 20

Select Group name: Group 3

Edit 3
Edit 1
Edit 9
Edit 12

Changes to Original Design



Information design section

The only font used in this editor is “Arial” and its size is 11 througout. A detailed overview of the color palette is provided in the table below. Moreover, the icons used in the editor were downloaded from “Icons8 Pichon”.

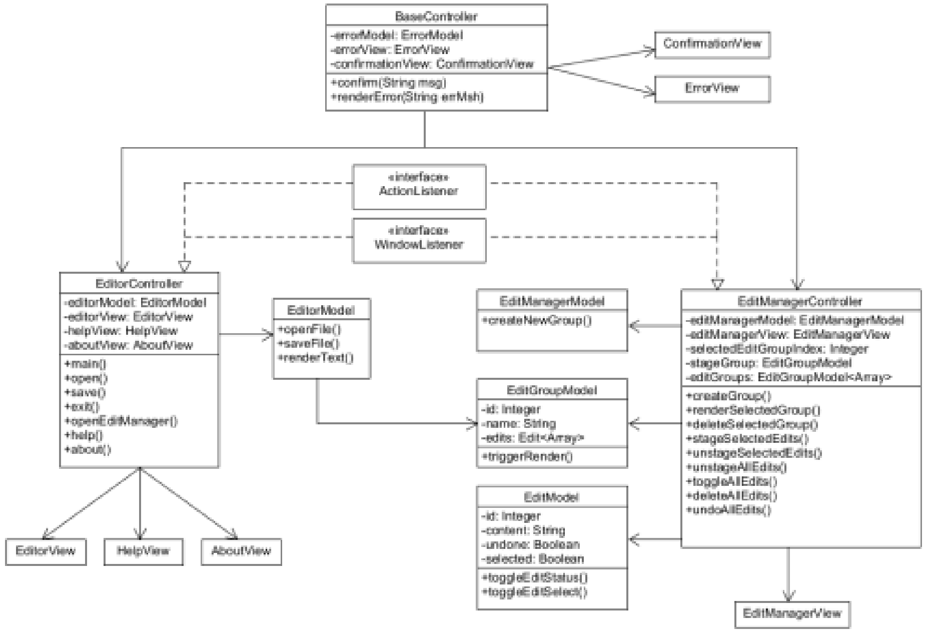
Frame	**Color**	**Code**
Main Panel (Text area)	White	(255,255,255) / #FFFFFF
Frame of SubItems	Silver	(192,192,192) / #C0C0C0
Background of Windows	Silver	(192,192,192) / #C0C0C0
All check box	Black	(0,0,0) / #000000
Toolbar	White	(255,255,255) / #FFFFFF
All Fonts	Black	(0,0,0) / #000000
All List Views	White	(255,255,255) / #FFFFFF
All Scroll bars	Gray	(128,128,128) / #808080
All Buttons	Dark Gray	(174,174,174) / #AEAEAE

Add checkboxes to the staged area edits

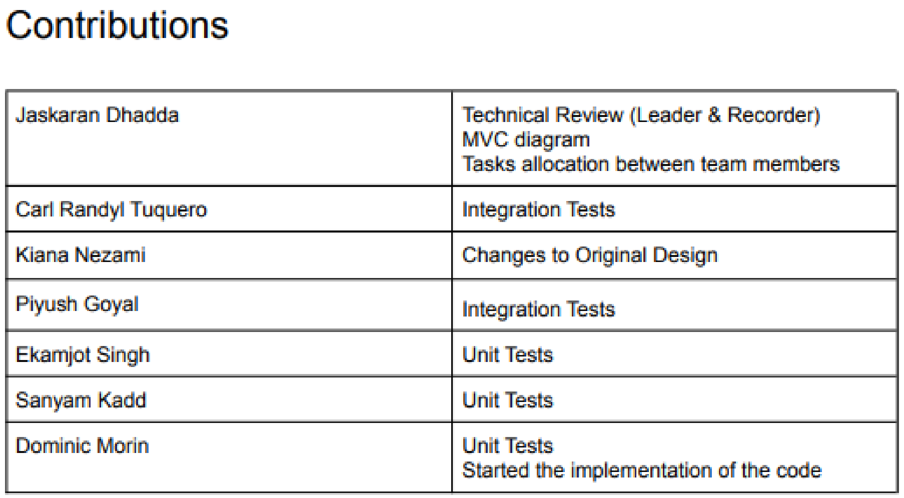
By adding a checkbox to this section in the “Edit Manager” tab, the user will be able to select multiple edits together and return it to the previous section before implementing Undo or Delete.

MVC Architecture

MVC Diagram



Contributions



Contributions

Jaskaran Dhadda	Technical Review (Leader & Recorder) MVC diagram Tasks allocation between team members
Carl Randyl Tuquero	Integration Tests
Kiana Nezami	Changes to Original Design
Piyush Goyal	Integration Tests
Ekamjot Singh	Unit Tests
Sanyam Kadd	Unit Tests
Dominic Morin	Unit Tests Started the implementation of the code