

Elaboration

Team Moebius-Qube

Amy Wieliczka, Kane Hadley, Matt Murphy, Gavin Templer

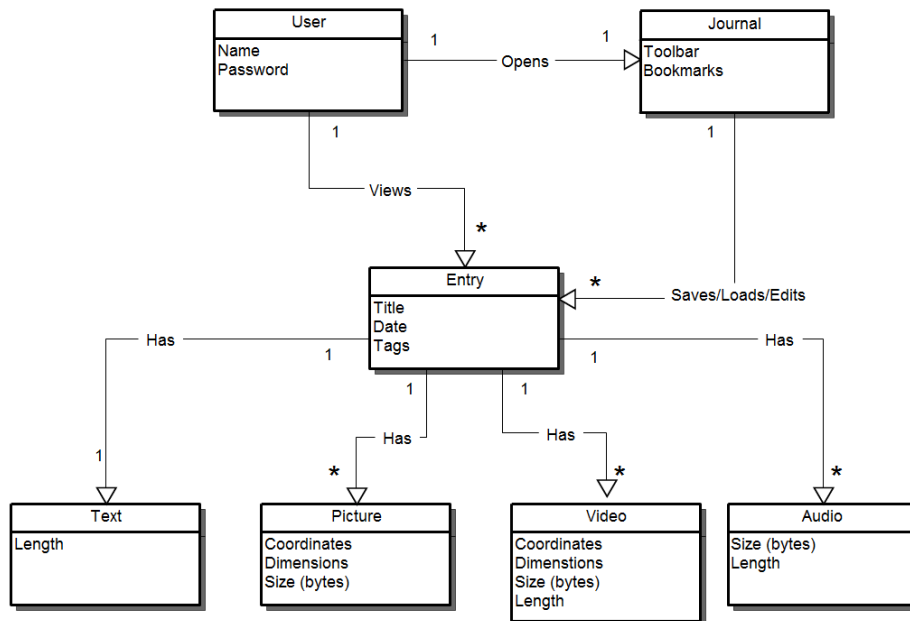
Project Status Report

This week we worked toward mitigating the risk of making sure our software works across the Mac, Windows, and Linux operating systems, by making sure all group members could run the QT development environment and the QT library runs cross platform. We primarily worked on our elaboration deliverables, but also flushed out our informal mock-ups to functional proof-of-concepts and began to build our UI and backend, focusing primarily on journal navigation.

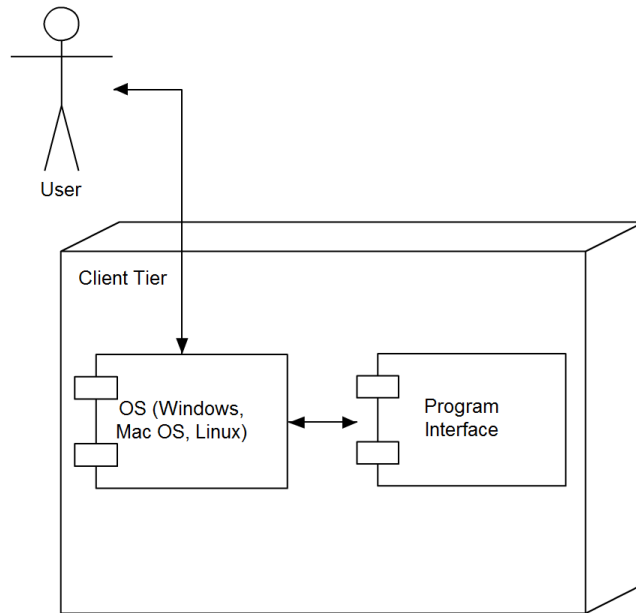
Contribution Summary

For our elaboration deliverables we separated the workload fairly evenly amongst everyone in the group. Matt put together the domain model diagram, deployment diagram, and worked with Amy on the system sequence diagrams. Amy worked primarily on the use cases as well as the functional proof-of-concepts. Kane put together our work breakdown structure and our revised schedule and did some work on our backend and the format we'd like to save our files in, while Gavin worked on our supplemental specifications.

Domain Model Diagram:



Deployment Diagram:



Supplemental Specification:

Must:

- Must be able to type and format text
- Must be able to link journal entries
- Must be able to tag entries
- Must be able to remove tags
- Must have a unique icon for all operations
- Must have help documentation for all operations
- Must be able to save and load entries
- Must be able to search for entries by title
- Must be able to search for entries by tags
- Must be able to view entries full screen
- Must be able to view multiple entries per page
- Must be able to upload media of industry standard file formats
- Must be able to upload entries to at least 1 specific blog site
- Must be able to delete entries

Should:

- Should be able to upload entries to a variety of blog sites
- Should have hotkeys for all options
- Should have a toolbar dedicated to text formatting
- Should be able to search documentation for keywords
- Icons should have the same theme
- Should have at least 1 alternative theme

Could:

- Could be able to upload media of less used file formats
- Could have numerous skins/themes
- Could have numerous entry viewing layouts

80% Use Cases

Use Case: Navigates journal entries

ID: UC1

Description: The author navigates through all journal entries for one or more entries matching author's specifications

Actors: author

Precondition:

- 1) The author has loaded the software
- 2) The author has set how many entries per page to view via UC8: Set Number of Entries to View per Page

Flow of events:

- 1) The author selects search bar, calendar button or tag list buttons
 - a. The calendar or list of tags is made viewable
- 2) The author provides search text, selects a date or series of dates, or a tag or series of tags
- 3) The author is prompted with a list of entry dates & titles of entries falling within parameters ordered by date and the number of entries to view per page displayed in journal viewer.

Postcondition:

- 1) List of entries containing search word available for browsing by their date and title

Use Case: Edit Entry

ID: UC2

Description: The author edits a journal entry.

Actors: author

Preconditions:

- 1) The author has loaded the software
- 2) The author selects a preexisting entry or a new entry to edit
- 3) Formatting toolbar becomes visible

Flow of events:

- 1) The author edits entry title
- 2) The author edits entry text
- 3) The author edits entry tags
- 4) The author links entry to another entry via UC3: Create Entry Link
- 5) The author adds media via UC4: Add Media

Postconditions:

- 1) Media is available for editing via formatting toolbar
- 2) Entry is available for upload via UC5: Entry Upload

3) Entry is available to delete via UC6: Entry Deletion

Use Case: Create Entry Link

ID: UC3

Description: The author creates a link to another journal entry.

Actors: author

Preconditions:

- 1) The author has selected a preexisting entry or new entry to edit
- 2) The author has selected a piece of text, image, video, or audio

Flow of events:

- 1) The author selects to add a link
- 2) The author is presented with a list of entries to link to
- 3) The author selects one
- 4) Link is successfully created or error message is displayed

Postconditions:

- 1) Previously selected text, image, video, or audio, when selected, will display linked entry below current entry.

Use Case: Add Media

ID: UC4

Description: The author adds media to a journal entry.

Actors: author

Preconditions:

- 1) The author has selected a preexisting entry or new entry to edit
- 2) Formatting toolbar is visible
- 3) Author selects Image, Video, or Audio buttons for adding media

Flow of events:

- 1) The author is prompted for media path
- 2) Author provides media path
- 3) Media is successfully placed or error message is displayed

Postconditions:

- 1) Media available to edit via formatting toolbar

Use Case: Entry Upload

ID: UC5

Description: The author seeks to upload an entry to a pre-existing weblog

Actors: author

Preconditions:

- 1) The author has selected a preexisting entry
- 2) The author has set a blog to upload to via UC9: Set Blog Preferences

Flow of Events:

- 1) The author selects upload entry
- 2) The author is prompted with text informing them if the upload was successful or not

Postconditions:

- 1) The upload entry button is no longer functional.

Use Case: Entry Deletion

ID: UC6

Description: The author seeks to delete an entry

Actors: author

Preconditions:

- 1) The author has selected a preexisting entry

Flow of events:

- 1) The author selects deletion button from toolbar

Postconditions:

- 1) The list of entries no longer contains that particular entry
- 2) The entries shown no longer show that particular entry

Use Case: Full Screen

ID: UC7

Description: The author views the journal at full screen.

Actors: author

Preconditions:

- 1) The author has loaded the software
- 2) The author has filtered a group of (1 or more) entries to view
- 3) The author has set how many entries per page to view via UC8: Set Number of Entries to View per Page

Flow of events:

- 1) The author selects full screen option
- 2) The journal viewer expands to fill full screen with the correct number of entries per page displayed

Postcondition:

- 1) Journal viewer has expanded to fill full screen
- 2) No software chrome visible

Use Case: Set Number of Entries to View per Page

ID: UC8

Description: The author sets the number of entries to view per page

Actors: author

Preconditions:

- 1) The author has loaded the software
- 2) The author has selected to edit their entry preferences

Flow of events:

- 1) The author is prompted with a series of choices with the current choice selected
- 2) The author selects their new choice
- 3) The author selects save preferences

Postconditions:

- 1) The new choice is selected

Use Case: Set Upload Blog Preferences

ID: UC9

Description: The author seeks to select a blog to upload their entries to.

Use Case: Set Theme Preferences

ID: UC10

Description: The author sets the stylistic theme of the journal entries

Actors: author

Preconditions:

- 1) The author has loaded the software
- 2) The author has selected to edit their theme preferences

Flow of events:

- 1) The author is prompted with a list of themes with the current theme highlighted
- 2) The author selects a theme
- 3) If the author selects preview theme
 - a. The current entries are shown in the new theme, though list of themes is still visible
- 4) The author selects save preferences

Postconditions:

- 1) The old theme is no longer highlighted
- 2) The new theme is highlighted

Use Case: Explicitly Load Journal

ID: UC11

Description: The author seeks to explicitly load a particular journal

Actors: author

Preconditions:

- 1) The author has loaded the software

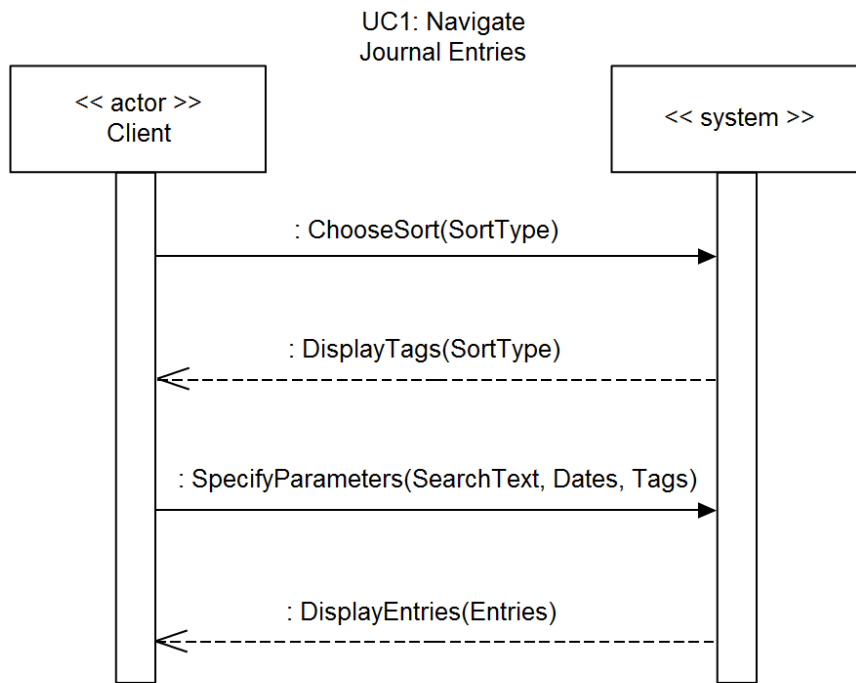
Flow of events:

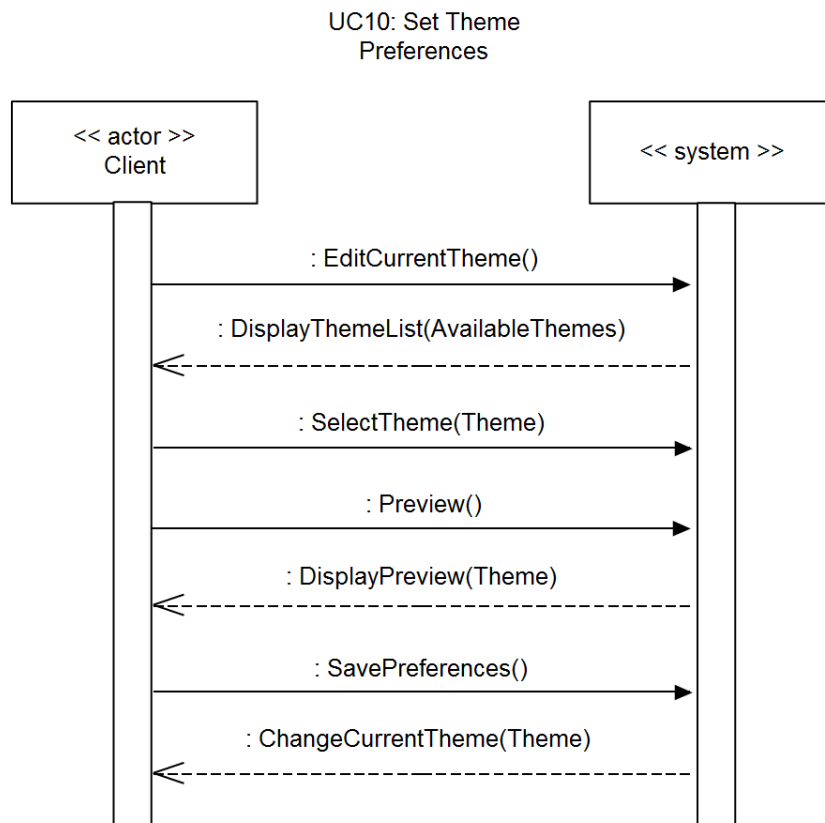
- 1) The author selects to load a journal
- 2) The author is prompted with their computer's contents
- 3) The author navigates to and selects journal to load
- 4) The author selects load journal

Postconditions:

- 1) The journal's entries are displayed in the entry navigation area
- 2) The journal's most recent few entries are displayed in entry display area.

System Sequence Diagram





Work Breakdown Structure

Create front end classes listed below

1. Menu Bar class to hold options and buttons
2. Editing Pane to hold different editable and focusable objects
3. Events activated by the user and the responses
 - Each Use Case Individually
 - Button events for the appropriate use cases.
 1. Tags button use case.
 2. Edit entry use case to add more text or other data.
 3. Create entry link use case.
 4. Add media use case to add pictures, video, and music.
 5. Delete entry use case to remove permanently.
 6. Full screen button use case for when the user wants to see only the editing pane.
 7. Number of entries to view button when the user wants to see more than one entry.
 8. Load journal button when the user wants to load a different set of entries from a different journaling book.
 - Mouse events for the appropriate use cases.

Create back end classes listed below

1. Entry class that represents an individual user created entry with accompanying insertions
 - Adding types of data to the entry (text, pictures, etc.).
 - Saving the entry to a file for later editing.
 - Loading an entry from a file to continue editing.
 - Modifying data in an entry (text) to the user's desired result.
2. Label class that represents the tags for each entry.
 - Adding data to the labels to either identify more entries or more labels.
 - Modifying data in the label to the most up to date and relevant need.
3. Picture class for any images the user wants to place on the page.
 - Adding data to the picture class (coordinates, image, wtc.).
 - Modifying data in the images (or maybe image modification).
4. Video class for any viewing media the user wants in the entry.
 - Adding data to the video object. It could be a reference, coordinates, etc.
 - Modifying data in the video. May be able to do some video editing at some point.
5. Music class for when the user wants to add audio data to the entry.
 - Adding data to the music.
 - Modifying the music in some form for others to hear and help understand the entry.
6. Search class for finding certain information really quickly.
 - Searching the text for an entry either in the direct entry itself or in the title.
7. File format(s) class(es) for storing the entries while not using them.
 - Create a specific way to save the class to a file for a saved entry on hard disk.
 - Create a way to load the class from a file to modify it.
8. Entry links to connect related entries.
 - Add Entry to the list of entries related for quick referencing or a reminder.

Miscellaneous

1. Icons for the buttons and clickables.
2. Art for the background and user interface for it to be appealing.
3. Deployment Diagram to show how the product will be used and where it might want to be used.
4. Domain Model Diagram to give an overview of the structure of the program.
5. Supplemental Specification for any extra considerations that might be useful.
6. System Sequence Diagrams to help guide our pseudo code when the time comes.

7. Use Cases to understand everything needed at a minimum to satisfy the users.
8. Work Breakdown Structure for keeping track of the remaining work.
9. Project Schedule to make sure the product will be completed on time.
10. Static Class Diagram to understand the relationships between all the classes in the product.
11. Code Review to find errors and keep code extendable and efficient.
12. Testing Documents to know what unit cases must succeed before we consider a part completed.
13. User Documentation to help users understand our product as well as provided basic use instructions.

Updated Project Schedule

- Week 1 (1/28/10): Learn about the course and design processes
- Week 2 (2/4/10): Learn about deliverables
- Week 3 (2/11/10): Create a group and decide upon an initial idea
- Week 4 (2/18/10): Plan out inception deliverables
- Week 5 (2/25/10): Design documentation Overview, Learn Java and XML, Submit inception deliverables
- Week 6 (3/4/10): **Iteration 1:** Second submission of inception deliverables
Continue design documentation
- Week 7 (3/18/10): Attempt to add to design documentation independently.
- Week 8 (3/25/10): **Iteration 2:** Elaboration reviewed and ends. Meeting to address initial problems and to foresee problems.
- Week 9 (4/1/10): **Iteration 3:** Construction 1 ends and Elaboration is submitted. Basic foundation of product is completed. Entry, a few buttons in GUI, some Events, and File format coded. Parts of Video, Music, and Pictures would be desired.
- Week 10 (4/8/10): Linking and Labeling are completed. Improve GUI. Add appropriate Events.
- Week 11 (4/15/10): **Iteration 4:** Construction 2 ends. Refine GUI to be more user friendly now that the product is functional. Finish events for buttons. First code review.
- Week 12 (4/22/10): Complete helpful extras such as search capabilities and anything requiring some form of computed intelligence. Begin write up of user documentation.
- Week 13 (4/29/10): **Iteration 5:** Construction 3 ends and construction deliverables are submitted. Integrate front end and back end. Second code review
- Week 14 (5/6/10): End of semester project completion. Optimize existing code and prepare presentation.
- Week 15 (5/13/10): Product completed and tested and presentation completed satisfactorily.