HiveText

Sprint 1 Retrospective

Team 25

George Albrecht, Arun Arjunakani, Bryan Battles, Alexander Geier, Paramesh Pradeep, Guangqi Sun

What Went Well?

User Story #1: As a user, I would like to edit plain-text files.

Task #	Description	Time (Hours)	Assigned Team Member
1	Integrate the ACE text editor into the client GUI (JavaScript).	10	Guangqi Sun
2	Implement “Open File” - displays a non-local file within the client and enables editing.	10	Bryan
3	Test that a user can insert/delete/highlight text anywhere in a file	.5	Guangqi Sun
	Total	20.5	

Discussion: ACE editor was integrated nicely, and provided some nice features that we didn’t expect when we decided to incorporate it. Open File works great, and menu bar functionality was eventually added.

User Story #2: As a user, I would like to interact with a desktop user interface

Task #	Description	Time (Hours)	Assigned Team Member
1	Create an active users list in the GUI.	5	Arun Arjunakani
2	Create a list for user edits in the GUI, with checkboxes.	5	Arun Arjunakani
3	Create a bar for tabs in the GUI	5	Arun Arjunakani
4	Add a toolbar (File, Edit, View, etc. dropdown menus)	4	Arun Arjunakani
5	Add menu options:	1	Arun Arjunakani

	-Save locally -SaveAs locally -Open locally		
6	Add barebones placeholder for chat window (with ability to minimize)	5	Arun Arjunakani
7	Create a user pane in the bottom right corner to contain the current logged in user’s username.	2	Bryan
8	Create an Icon for HiveText	2	Arun Arjunakani
9	Test for functionality in UI according to acceptance criteria.	1	Arun Arjunakani
	Total	32	

Discussion: A functional frontend was developed for the user to interact with. There are buttons for most things that the user might need and a menu bar with more detailed options.

User Story #3: As a user I would like a cross-platform application.

Task #	Description	Time (Hours)	Assigned Team Member
1	Incorporate Electron into the project	5	Bryan
2	Test client on different operating systems	1	Guangqi Sun
	Total	6	

Discussion: The application is now structured on electron, allowing it to be cross-platform by default. The team had to learn that a main process spawns new windowed processes, so in some circumstances using IPCs is necessary to communicate across windows.

User Story #4: As a user, I would like to have a personal user account.

Task #	Description	Time (Hours)	Assigned Team Member
1	Create a “userList” node in the database from root, along with a schema for “user”, including userID, username, password, and fileList fields.	1	Bryan
2	Create registration functionality that allows a	3	Bryan

	user to register an account and store their account information in the database.		
3	Create login functionality, that allows a user to login to their registered account and access all information stored in their user account on the database.	3	Bryan
4	Create logout functionality on the main GUI that allows a user to logout of their account once logged in.	2	Bryan
5	Create a “Forgot Password?” link with functionality to allow the user to recover their password through email.	4	Alex
6	Test that user can register an account, login as a registered user, and logout of their account according to the acceptance criteria	1	Bryan
	Total	14	

Discussion: The user can log in and log out. This allows them to see a list of all the files they have worked on. They can be ensured that someone cannot access their account or files without a password and also reset their password with their email if they forget it.

User Story #6: As a user, I would like to be able to edit files offline.

Task #	Description	Time (Hours)	Assigned Team Member
1	Allow an offline user still use the editor.	5	Alex
2	Allow offline users to open their machine’s local files.	5	Alex
3	Allow offline users to save changes on their machine locally.	10	Alex
4	Test for ability to edit files offline	1	Alex
	Total	20	

Discussion: Before a user signs in, they are still able to use the editor like any other editor. They can load files into the editor, edit them, and save them, all without an account. We wanted functionality to be available to users who are offline.

User Story #7: As a user, I would like to be able to upload a file to my user account.

Task #	Description	Time (Hours)	Assigned Team Member
1	Create an “open” option in the menu bar	1	Alex
2	Create functionality that allows the user to select a file from their local computer to open.	5	Alex
3	Create a “fileList” node in the database that will contain all files uploaded by users.	5	Bryan Battles
4	Save the file’s contents to the database under the “fileList” node and include a unique fileID, a fileName, and the userID of the creator.	10	Bryan Battles
5	Update the user’s fileList in the database to include the unique fileID of the file created.	5	Bryan
6	Show a list of the signed-in user’s files on the client’s GUI.	14	Bryan
7	Test functionality for uploading a file from local computer.	1	Paramesh
	Total	36	

Discussion: A user can upload a file to their user account by using the “open file” functionality while signed into their account. This functionality adds the user’s file to the firebase database with a unique fileID that contains their userID in it’s userList. This uploaded file will be able to be edited concurrently with other users after the next sprint.

User Story #8: As a user, I would like to be able to drag a file on top of the window to upload it.

Task #	Description	Time (Hours)	Assigned Team Member
1	Create functionality that detects dragged-and-dropped files onto the client’s GUI.	10	Paramesh
2	Determine if a dragged-and-dropped file should be opened in offline or online mode, and do so accordingly.	14	Paramesh
3	Test for ability to drag and drop files and open it in required mode	1	Paramesh

	Total	25	

Discussion: Instead of manually clicking the open file button, the user can simply drag and drop a file to open it in HiveText. This functionality uses the same process as a normal open file, but allows the user a small amount of extra flexibility in how they use the application.

User Story #9: As a user, I would like to be able to download a file to my local computer.

Task #	Description	Time (Hours)	Assigned Team Member
1	Create functionality that saves a local copy of the currently displayed text to a specified location.	6	Bryan
3	Test for functionality of save a file locally.	1	Bryan
	Total	8	

Discussion: Users can save a file locally. This allows a user to save a local copy of a file they opened locally or from the database. In either case, the functionality works the same.

User Story #10: As a user, I would like to be able to change the font size of the text.

Task #	Description	Time (Hours)	Assigned Team Member
1	Add a keybinding to ACE that recognizes CTRL and + as well as CTRL and - and increases/decreases the font size.	1	Paramesh
2	Add a “+” and a “-” button to the GUI that increases the font size of ACE.	1	Paramesh
3	Add a menu option to reset font size	.5	Paramesh
4	Test the client for support of font resizing using both methods.	.5	Paramesh
	Total	3	

Discussion: Users can now change the font size with menu options and shortcuts. They can also reset the font size. These features were found to be native to ACE, so not much time was spent on this user story.

User Story #11: As a user, I would like to have automatic code indenting.

Task #	Description	Time (Hours)	Assigned Team Member
1	Add automatic file type detection with syntax highlighting.	2.5	Alex
2	Test the client for support of auto-indenting for multiple different languages.	.5	Paramesh
	Total	3	

Discussion: The program will automatically identify the file type of the opened file and will automatically indent and highlight syntax based on the language. This feature was found to be native to ACE as long as we provided ACE the filetype open opening a file.

User Story #12: As a user, I would like to turn line numbers on and off.

Task #	Description	Time (Hours)	Assigned Team Member
1	Add a menu option that toggles line numbers in ACE.	1	Guangqi Sun
2	Test line number toggling in the client.	.5	Guangqi Sun
	Total	1.5	

Discussion: Users can now toggle the ability to see line numbers. This feature was found to be native to ACE, so not much time was spent on this user story.

User Story #13: As a user, I would like to be able to find and replace text.

Task #	Description	Time (Hours)	Assigned Team Member
1	Activate find & replace functionality through ACE.	.5	Guangqi Sun
2	Test find & replace functionality in client	.5	Guangqi Sun
	Total	1	

Discussion: Users can now find and replace text within a file. This feature was found to be native to ACE, so not much time was spent on this user story.

User Story #14: As a user, I would like to be able to cut/paste text via cursor or keyboard shortcuts.

Task #	Description	Time (Hours)	Assigned Team Member
1	Activate CTRL + C, CTRL + X, and CTRL + V functionality through ACE.	.5	Paramesh
2	Test copy, cut, and paste functionality in the client.	.5	Paramesh
	Total	1	

Discussion: Users can now copy and paste through the menu and with keyboard shortcuts. This feature was native to ACE, so not much time was spent on this user story.

User Story #15: As a user, I would like to be able to change my password.

Task #	Description	Time (Hours)	Assigned Team Member
1	Add “User Settings” pane to the GUI with an option to logout	2	Alex
2	Add functionality that logs a user out and returns to the login page	2	Bryan
3	Test for changing password	.5	Alex
	Total	4.5	

Discussion: Users can now change their settings on a page separate from the main GUI. This functionality allows the user to logout as well as change their user account details. This functionality was adapted for an edge case to also update the username fields of any file that belongs to the user if the user changed their username.

What Did Not Go Well?

User Story #5: As a user, I would like to sign in with my Google account as my user account.

Task #	Description	Time (Hours)	Assigned Team Member
1	On the login page, include a button that allows a user to sign-in with their Google account using Google Auth.	5	George
2	Create a pop-up that asks first-time Google account sign-ins to create a username for their account.	3	George
3	Save user data from new Google account sign-ins to the database.	3	George
4	Test for ability to register/sign-in as Google account.	1	George
	Total	12	

Discussion: This user story was not completed by the time we presented at our demo. This was due mainly in part to George not contributing a substantial amount to the project until the final Sunday of sprint 1. As of the end of sprint 1, functionality for registering an account through a google account is working, but signing in with your google account has not been implemented yet. This functionality will have to be moved to the beginning of sprint 2, where George should complete it as soon as possible.

How Should We Improve?

1. Next sprint we need to make sure that we are all working each week, and that we complete our tasks in a timely manner. Last sprint, not everyone worked throughout the entire sprint, and many tasks were not completed until the final week of the sprint.
2. Now that lectures have concluded, we should meet as a team during this open time-slot to work on the project as a team. This means we will have 3 days a week to meet face-to-face to establish goals, clarify implementation details, and assist each other with any problems encountered.