Meeting Minutes
CS 673 Team # 2

Week 6 Meeting (6/15 - 6/22)

Date and Time: 06/17/21 9:30PM - 10:21 PM
Place: Zoom Meeting
Meeting recording: 2021-06-18 21.30 MET CS 673 Team 2 meeting
Participants: Kayla Bayusik, Andrew Klimentyev, Francis Xavier Pulikotil, Zhaowei Gu, Alexander Dewhirst
Minutes taker: Zhaowei Gu
Timekeeper: Zhaowei Gu - 1hr
Purpose: We need to plan tasks for the final week.
Agenda:

- Main tasks:
    - Security - vault encryption
    - Finish all the testing
        - Manual
        - Automated
    - Entire team presentation
        - All member must join
        - Update the PPT
    - SDD - Classes and Methods
    - SPPP - update

Discussions:

- Writing model and test for the page and vault
- For security we are trying to plugin into the main branch making sure its not going to break
- The automatic test when you make a pull request is done
    - Write more test
- Make sure do a manual test as well
- Professor mention it's more important to be aware the security related concerns than actually implementing it
- Need to make time with TA and professor
    - Quick practice before actual live presentation
    - Monday, try to do the presentation
    - After 5PM EST is better
- We use a **composite design pattern**, when you render a blueprint it attaches it.
    - App.url map can see the rout
- Second meeting around sunday.

Key Decisions:

- Things to update/complete
    - Security
    - SDD
        - Class and method
        - Design pattern
    - SPPP
    - Progress report
        - contribution
    - Complete pivotal tracker

Action Items:

Complete everything

Week 5 Meeting (6/08 - 6/15)

Date and Time: 06/09/21 8:30 - 9:55 PM
Place: Zoom Meeting
Meeting recording: 2021-06-09 20.30 MET CS 673 Team 2 meeting
Participants: Kayla Bayusik, Andrew Klimentyev, Francis Xavier Pulikotil, Zhaowei Gu, Alexander Dewhirst, and Neha Abrol (facilitator)
Minutes taker: Kayla Bayusik
Timekeeper: Kayla Bayusik - 85 mins
Purpose: We need to address facilitator's comments from previous iteration submission
Agenda:

- In the next Iteration I would like to see **E2E tests** and incorporate the security recommendations made by the Professor.
- Also would like to see a little more **detail in the Security Arch and Design Patterns** next time and not just 1 FAT controller.
- Please keep **filling/updating Project Contribution** Tab
- For next Iteration would love to see **incorporation on some of the guidance by Professor on Passwords for Week 4 meeting on 06/2**
- Looking forward to seeing the **Security Recommendations** from Week 4 meeting in the next Iteration.
- I don't see controllers clearly written (app.py is too fat). Its recommended to have routes and business logic separated so code is easier to follow and also there is Separation of Concerns

Discussions:

- Manual testing of the entire application - write a few manual tests (in acceptance test format) as steps and go through them.
- Start on end-to-end testing, covering the entire flow not just single features/units - suggested use of Selenium
    - Change the name of our unit tests to e2e tests to remove confusion
    - Split large test file into smaller components

    - Create separate files for features and create folders with controllers to execute the business logic
- Expand security and design sections of SDD
- Reminder to keep notes or update project contribution tab of progress report spreadsheet
- Go over encryption algorithm design - clarify differences between vault key and encryption key
- Use CSRF token to prevent XSS:
    - How to enable CSRF protection in the Python / Flask app?
    - Should we include CSRF when we use flask-jwt-extended?

Key Decisions:

- Store user's password encrypted in session
- Deciding on individual member's tasks:
    - Kayla finish encryption/decryption and class
    - Zhaoewi write manual/e2e tests
    - Alexander work on blueprints and splitting app.py into smaller controllers
    - Francis write new vault item page
    - Andrew look in to CSRF

Action Items:

- Create vault item page
- Set up encryption class
- Separately defined manual, unit, and end-to-end tests
- Divide app.py controller into small components by business logic

Week 4 Meeting (6/01 - 6/08)

Date and Time: 06/02/21 8:30 - 9:40 PM
Place: Zoom Meeting
Meeting recording: 2021-06-02 20.30 MET CS 673 Team 2 meeting
Participants: Kayla Bayusik, Andrew Klimentyev, Francis Xavier Pulikotil, Yuting Zhang (Professor), and Neha Abrol (facilitator)
Minutes taker: Francis Xavier Pulikotil
Timekeeper: Francis Xavier Pulikotil - 70 mins
Purpose: Demo current state of project to Professor and facilitator, and discuss security, project management, and next steps.

Agenda:

- Demo current state of project; talk about security features; project management; ask for feedback from Professor and facilitator.
- Deliverables for this week
    - Quiz 2
    - Lab 3

Discussions:

- Neha reminded us that although Zhaowei is the QA lead, he's not responsible for

writing all the unit tests, all of us are. He is responsible for making sure that we are testing things appropriately.

- Neha mentioned that we should have stories put into PivotalTracker, assign them to people at the start of each sprint, and then work on those stories throughout the sprint.
- We talked about enabling email notifications in PivotalTracker so that we get notified when someone makes changes to the board, or changes issues, etc.
- We went into a very detailed discussion on security, hashing, encryption, etc. Professor jumped in and gave us a mini class on hashing and encryption related to security, as it applies to the real world. Highly recommended to watch the video recording to get all the details.
- We discussed our plan to move to multiple vaults (Alexander's PR on GitHub); concerns were increased complexity, implementation time.
- Professor talked about how stories get worked on, the various stages of the story's state, and how we can structure the way we work on stories.

Key Decisions:

- Put stories into PivotalTracker; assign to team members at the beginning of each sprint; work on the assigned tasks throughout the sprint.
- Based on input from the Professor, we have a new algorithm to encrypt the vault:
    - Encrypting the vault itself
        - Generate a cryptographically strong random number (X) (e.g. using secrets)
        - Encrypt the user's vault with X
        - Store the encrypted vault in the database
    - Encrypting X (the vault key)
        - User provides password (P)
        - Create a unique salt (S) from P (e.g. hash using bcrypt)
        - Store S in the database for that specific user.
        - Hash P with S to generate an encryption key (K) (e.g. using bcrypt or PBKDF2)
        - Use K to encrypt the vault key X
        - Store the encrypted vault key in the database
    - Decrypting the vault
        - User provides password (P)
        - Hash P with the user's salt (S) to regenerate the encryption key (K)
        - Use K to decrypt the vault key X
        - Use X to decrypt the user's vault
    - User changes password
        - User provides old password (P) and new password (P2)
        - Hash P with the user's salt (S) to regenerate the encryption key (K)
        - Use K to decrypt the vault key X
        - Hash P2 with S to generate a new encryption key (K2)
        - Use K2 to encrypt the vault key X
        - Store the new encrypted vault key in the database

        - Now the next time the user wishes to access their vault, they will need to use P2
    - Other Notes
        - We encrypt the user's vault using X and not directly using K so that we don't need to re-encrypt the entire vault whenever the user changes their password. We only need to re-encrypt X.
- Professor discussed XSS and CSRF exploits; we need to research how to prevent these.
    - To prevent CSRF exploits we can use a CSRF token with every request/response. See: https://stackoverflow.com/a/33829607
- We can go ahead with the multiple vaults design (Alexander's PR on GitHub), but we all should try to review the changes as a team.

Action Items:

- Create stories on PivotalTracker; assign to team members.
- Implement new security algorithms and features as discussed.
- Continue working on other features of the project such as CRUD operations for vault items, password generation UI, etc.

Week 3 Meeting 2 (5/25 - 6/01)

Date and Time: 05/30/21 8:30 - 9:40PM
Place: Zoom Meeting
Meeting recording: 2021-05-30 20.30 MET CS 673 Team 2 meeting
Participants: Kayla Bayusik, Zhaowei Gu, Alexander Dewhirst, Francis Xavier Pulikotil, Andrew Klimentyev
Minutes taker: Andrew Klimentyev
Timekeeper: Andrew Klimentyev
Purpose: Discuss current state of application and next steps.

Agenda:

- Walkthrough and Demo of Hashing and Vault UI
    - Password is stored within session cookies
    - Use session id to ensure user is logged in
- Deliverables for this week
    - Presentation
        - Product demo
        - Cover DB design and UI

Discussions:

- Vault passwords can be encrypted from the password stored in the session cookie as the key.
- Create unit tests that use the user class to modify the database
- Expand the user class to include changing password and vault entries
- Update some user stories before exporting to fit into the expected format
- Add a setup script to the root of the repository

- Run tests before committing code

Key Decisions:

- Continue iterating on our current tasks in Pivotal Tracker.
- Keep default color scheme from flask

Action Items:

- Create buttons for generating password vault entries
- Encrypt passwords stored in vault
- Prepare presentation
- Update SDD and SPP
- Update stories in pivotal

Week 3 Meeting 1 (5/25 - 6/01)

Date and Time: 05/26/21 8:30 - 10:00 PM
Place: Zoom Meeting
Meeting recording: 2021-05-26 20.30 MET CS 673 Team 2 meeting
Participants: Kayla Bayusik, Zhaowei Gu, Alexander Dewhirst, Francis Xavier Pulikotil, Andrew Klimentyev, and Neha Abrol (facilitator)
Minutes taker: Alexander Dewhirst
Timekeeper: Alexander Dewhirst - 75 minutes
Purpose: Discuss current state of application and next steps.

Agenda:

- Excellent Demo and Walkthrough Flask/SQLite3 App
    - Setup Python and activate virtual environment
    - Install Flask
    - Setup environment variables
    - Initialize Database
    - Start the server
- Present wireframes
    - Login
    - Register
    - Vault
    - New Entry

Discussions:

- Go over requirements for SDD documents. The first two sections are required. We should work on each section to the best of our abilities
- Presentation this week will be Kayla and Zhaowei

Key Decisions:

- Continue iterating on our current tasks in Pivotal Tracker.

Action Items:

- Work on an icebox object.
- Use ms-python.python VS Code extension for linting

- Alexander - SDD document
- Kayla, Zhaowei - Presentation

Week 2 Meeting (5/18 - 5/25)

Date and Time: 05/19/21 8:30 - 10:00 PM
Place: Zoom Meeting
Meeting recording: 2021-05-19 20.30 MET CS 673 Team 2 meeting
Participants: Kayla Bayusik, Zhaowei Gu, Alexander Dewhirst, Francis Xavier Pulikotil, Andrew Klimentyev, and Neha Abrol (facilitator)
Minutes taker: Zhaowei Gu
Timekeeper: Zhaowei Gu
Purpose: Create and assign tasks; start working on project

Agenda:
- Create tasks in Pivotal Tracker
- Assign some tasks to team members
- Figure out what else is required for this sprint
- Kickoff project implementation

Discussions:
- Kayla start some security for us to work on
- Start and setup
    - POC & TOC
- Epic - for user stories
    - Quality Assurance
        - All the work required to test and make sure the application works as required.
    - Password Vault
        - This is all the work required for a user to view their passwords, generate and store new passwords, and modify existing ones.
    - Login/Registration
        - All the work required for a user to be able to log into the application, or register for an account if they don't have one. This would also handle sessions, logging out, encryption of credentials, etc.
    - UI Mockups
        - Design of the user interface, will contain stories for creating the design of the UI. This should include wireframes, data flow, navigation, and semi-detailed page design.
    - Proof of concepts / Dev work
        - This epic is for creating small, rough proof of concepts which can be used as reference and to vet the various libraries we want to use.
- List of high level stories (prevent duplication):
- Icebox task

    - Python unittest setup for automated/manual testing
        - Setup unit tests for automatically running tests when code is committed, or for the programmer to run periodically.
    - Create wireframes for the pages
        - Create wireframes and design the pages, navigation, flow of the application.
    - Find out how to manage encryption and storage of user data
        - Steps required for encryption and storage of username/password, password vault, etc.
    - Cryptography
        - Find out how pycrypto works, and create a class which can be used in the project.
    - SQLite
        - Setting up SQLite, maybe provide a sample, build required classes for use in the project.
- Check with the professor if you also need to unittest the front end.

Key Decisions:

- Team member roles
    - Alexander Dewhirst - Create wireframes for the pages, SQLite
    - Zhaowei Gu - Python unittest setup for automated/manual testing
    - Andrew Klimentyev - Cryptography
    - Kayla Bayusik - Create wireframes for the pages, Find out how to manage encryption and storage of user data
    - Francis Xavier Pulikotil - Create sample Python/Flask application poc, help other members with technical work.

Action Items:

- Work on an icebox object.

Week 1 Meeting (5/11 - 5/18)

Date and Time: 05/16/21 7:00 - 8:20 PM
Place: Zoom Meeting
Participants: Kayla Bayusik, Zhaowei Gu, Alexander Dewhirst, Francis Xavier Pulikotil, Andrew Klimentyev, and Neha Abrol (facilitator)
Minutes taker: Kayla Bayusik
Timekeeper: Zhaowei Gu
Purpose: Set project and work on proposal

Agenda:

- Choose project
- Select technologies
- Requirements / features
- Next steps

Discussions:

- Choose project
    - PassMan - a password manager
- Select technologies
    - Discussing web app vs desktop app
        - Web app
        - Developed local, hosting considered at the end of the project
        - Must consider security for ultimately expanding beyond local hosting
    - Python backend, HTML5/Javascript front end, Ajax, SQLite
    - Python Flask framework
    - Javascript framework-less
        - Looking for confirmation from professor
    - Unit test for QA
- Requirements / features
    - Essential
        - Sections / pages: login, main
        - Login allows for user entering password
        - Main allows for clicking on passwords to view, edit, delete
        - Buttons to add new password to manager and exit / log out
        - Not storing password plain text - Python library for encryption
        - User registration
    - Desirable
        - Copy directly to clipboard without showing password
        - Dividing passwords into tags / categories
        - Import / export passwords - to encrypted file
        - Bootstrap styling
        - Share ?
    - Terminology:
        - User is a person who logs into the site
        - Accounts would be that user's logins for other sources
- Next steps
    - Fill out SPPP document
    - Fill out Progress Report spreadsheet
    - Go over documents Monday 5/17 evening
    - All team members finish lab 1, merge lab 1 branch to main
    - Make brief project proposal video

Key Decisions:

- Team member roles
    - Team Leader - Francis Xavier Pulikotil
    - Requirement Leader - Francis Xavier Pulikotil
    - Design and Implementation Leader - Alexander Dewhirst
    - QA Leader - Zhaowei Gu
    - Configuration Leader - Andrew Klimentyev
    - Security Leader - Kayla Bayusik

- Project idea itself
    - Password manager
    - “PassMan”
    - General scope and requirements of PassMan

Action Items:

- Finishing up the SPPP by Monday evening
- Finishing up the team Progress Report by Monday evening
- Video introduction