Sprint 2 Planning Document

Team 16 - xpac

Adit Kumar, Amol Moses Jha, Anunai Ishan, Parth Shelgaonkar

Sprint Overview

For the second sprint, we plan on implementing installation of composite packages having dependencies, for example, the popular text editor *vim*. This would include building up on the metadata class, which furnishes information about a package, that we implemented in the previous sprint. Furthermore, we aim to utilize the metadata class in order to generate a graph for the universe of packages and topological sorting in order to generate subgraphs , representing the dependency trees for their respective packages. The client-side will generate the aforementioned subgraph based on packages that have already been installed and request the server for any additional packages necessary.

Scrum Master: Adit Kumar

Meeting Plan: Sunday (5:30 pm to 7:30 pm)

Risks and Challenges: A significant portion of this sprint consists of tasks that are demanding in nature, such as generating subgraphs of the packages requiring installation on client-side. This will be crucial in efficiently installing composite packages, i. e., packages having dependencies. The client should be able to build this subgraph based on the information provided by the metadata class, chiefly the packages that are already installed, and install any additional packages necessary by requesting them from the server.

The server will also integrate the functionality to include the metadata class in order to be able to add new packages to the repository. This will allow the server to include new versions of packages and remove older versions from the repository using auxiliary scripts.

Current Sprint Detail

We plan to achieve the following user stories during the course of this sprint:

User Story #1:

As a user, I would like xpac to able to install composite packages.

Task Details:

#	Task Description	Estimated Time	Team	Owner
1	Integrate metadata classes with both the server and client.	**7 hrs**	**Client & Server**	**Amol & Parth**
2	Implement graph representation of packages on client-side.	**15 hrs**	**Client**	**Adit & Anunai**
3	Topologically sort the given subgraph.	**10 hrs**	**Client**	**Amol & Parth**
4	Build dependency trees based on the result of topological sorting of the sub-graph.	**15 hrs**	**Client**	**Adit & Anunai**

Acceptance Criteria:

➢ Given that xpac is able to handle packages with multiple dependencies both on the server and client side.

➢ Given that xpac is able to generate dependency trees using the metadata and perform topological sort to generate a subgraph while installing a composite package.

User Story #2:

As a user I would like xpac to utilize compression and decompression functionalities in order to efficiently manage my packages.

Task Details:

#	Task Description	Estimated Time	Team	Owner
1	Set up functions to compress and decompress metadata employing common compression algorithms.	**16 hrs**	**Server & Client**	**Amol & Parth**

Acceptance Criteria:

➢ The server is able to compress the metadata of the universe of packages.

➢ The client is able to decompress this metadata before storage and use locally.

User Story #3:
As a user, I would like to ensure that the packages served by xpac are not corrupt or malicious.

Task Details:

#	Task Description	Estimated Time	Team	Owner
1	Implement checksum verification of files fetched, utilizing the MD5 algorithm on client-side.	**12 hrs**	**Client**	**Amol & Parth**
2	Implement the hashing functionality for MD5 on server-side.	**12 hrs**	**Server**	**Amol & Parth**

Acceptance Criteria:
➢ The client is able to perform validation on the integrity of file fetched from the server by comparing the hash generated using the MD5 algorithm on client-side to the hash on the server-side.
➢ The server is able to store the hash in the metadata of the new package being introduced into the universe of packages.

User Story #4:
As a user, I wish to be able to list all the packages that have been installed by xpac.

Task Details:

#	Task Description	Estimated Time	Team	Owner
1	Set up client to store and list all the packages available.	**7 hrs**	**Client**	**Amol & Parth**

Acceptance Criteria:
➢ The client is able to maintain a list of packages that have already been installed.
➢ The client is able to check against this list in order to prevent the user from trying to install a package previously installed.

User Story #5:
As a user, I would like xpac to network with developers and provide a platform for users to submit valuable feedback and critiques about xpac.

Task Details:

#	Task Description	Estimated Time	Team	Owner
1	Set up an IRC server for the users to be able to connect to the developer community.	**7 hrs**	**Website**	**Adit & Anunai**
2	Link the feedback form where users can submit valuable feedback and bug reports to the website	**5 hrs**	**Website**	**Adit & Anunai**

Acceptance Criteria
➢ Given that the user is able to join the chat room on the IRC channel and connect with developers.
➢ Given that the developers are able to view reported bugs and feedback from the users.

User Story #6:
As a user, I would like xpac to be a stable and robust piece of software.

Task Details:

#	Task Description	Estimated Time	Team	Owner
1	Set up unit testing for server.	**7 hrs**	**Testing**	**Adit & Anunai**
2	Set up unit testing for client.	**7 hrs**	**Testing**	**Adit & Anunai**

Acceptance Criteria:
➢ Given that the above user stories are completed, rigorous testing and quality control should be performed on each component of xpac and its functionalities to ensure stability and robustness.

Remaining Backlog (12/25 Stories Complete)

1. ~~As a user, I would like to have the ability to connect to the server which hosts repositories.~~
2. ~~As a user, I would like the server to establish multiple connections with clients, concurrently.~~
3. ~~As a user, I wish to be able to send and receive files to and from the repository.~~
4. ~~As a user, I would like xpac to able to efficiently manage packages.~~
5. As a user, I would like to have the ability to know the packages hosted by the server.
6. As a user, I would like to have the ability to know the versions available of a particular package.
7. ~~As a user, I would like to have the ability to install composite packages.~~
8. ~~As a user, I would like to have the ability to list the packages already installed using xpac.~~
9. As a user, I would like to have the ability to update the software database.
10. As a user, I would like to view if there are any updates available for the packages installed.
11. As a user, I would like to have the ability to update packages.
12. As a user, I would like to have the ability to do a full system-wide update.
13. As a user, I would like to have the ability to delete packages.
14. As a user, I would like to have the ability to delete orphans and unneeded packages.
15. As a user, I would like to have the ability to compile from source from packages if needed.
16. As a user, I would like to have the ability to choose between a stable version, or a current bleeding-edge version package for installation.
17. As a user, I would like to be able to automate package management using xpac.
18. As a user, I would like to be able to tweak and customize xpac for a more personalized experience.
19. As a user, I would like to have the ability to use xpac to manage packages on different Unix based operating systems.
20. ~~As a user, I would like to have access to proper software documentation, like man pages for xpac.~~
21. ~~As a user, I would like xpac to have an online presence to be able to to connect with the community of contributors and other users.~~
22. ~~As a user, I would like xpac to have a way to connect to the developers in order to submit valuable feedback and necessary critiques and comments about xpac.~~
23. ~~As a user, I would like xpac's website to responsive and elegant.~~
24. ~~As a user, I would like xpac to be a stable and robust piece of software.~~
25. ~~As a user, I would like to have xpac to be easy to use.~~