Gitlet Design Document

Project Overview

Gitlet is a lightweight version control system inspired by Git. It supports basic features like initializing repositories, staging files, creating commits, switching branches, and restoring files. This implementation focuses on providing core functionality while maintaining a modular structure.

Features

- **Initialization**: Create a new Gitlet repository.
- **Staging Files**: Stage files for commits.
- **Committing Changes**: Record the current state of the project.
- **Branching**: Create and switch between branches.
- **File Restoration**: Restore specific files from commits.
- **Commit Logs**: Display history of commits.
- **Global Logs**: Display all commits in the repository.
- **Finding Commits**: Locate commits by message.
- **Status**: View repository status, including staged files and branches.
- **Reset**: Reset the working directory to a specific commit.

Class Design

1. Main Class

Purpose: Entry point for Gitlet. It parses commands and delegates tasks to the Repository class.

Methods:

- main(String[] args): Routes commands to appropriate methods in the Repository class.

---

2. Repository Class

Purpose: Core logic of Gitlet, handling commands and repository-level operations.

Instance Variables:

- CWD: Current working directory.
- GITLET_DIR: Directory for Gitlet metadata.

- STAGING_DIR, COMMITS_DIR, BLOBS_DIR: Subdirectories for staging, commits, and file blobs.
- HEAD_FILE, CURRENT_BRANCH_FILE: Track the current commit and branch.
- BRANCHES_DIR: Directory for branch metadata.

Methods:

- **init()**: Initializes a new Gitlet repository.
- **add(String fileName)**: Stages a file for commit.
- **commit(String message)**: Saves a snapshot of the current project state.
- **rm(String filename)**: Removes a file from staging or marks it for deletion.
- **restore(String... args)**: Restores a file from a specific commit.
- **log()**: Prints the commit history of the current branch.
- **globallog()**: Displays all commits in the repository.
- **find(String message)**: Finds commits by their message.
- **status()**: Displays repository status, including branches, staged files, and untracked files.
- **branch(String branchName)**: Creates a new branch.
- **switchbranches(String branchName)**: Switches to a different branch.
- **rmBranch(String branchName)**: Deletes a branch.
- **reset(String commitId)**: Resets the working directory to a specific commit.

Helper Methods:

- **getHeadcommitID()**: Retrieves the current head commit ID.
- **getCommitbyID(String commitID)**: Retrieves a Commit object by its ID.
- **loadstagingArea() / saveStagingArea()**: Manages the staging area.

---

3. Commit Class

Purpose: Represents a commit in the repository.

Instance Variables:

- message: Commit message.
- parent: ID of the parent commit.
- fileSnapshots: Mapping of file names to their blob hashes.

- secondParent: Optional second parent for merge commits.
- timestamp: Date and time of the commit.

Methods:

- **getMessage()**: Returns the commit message.
- **getTimestamp()**: Returns the commit's timestamp.
- **getParent()**: Returns the parent commit ID.
- **getSecondParent()**: Returns the second parent ID (if any).
- **addFileSnapshot(String fileName, String fileSha1)**: Associates a file with its blob hash.

---

4. Utils Class

Purpose: Provides utility methods for file operations, SHA-1 hashing, and serialization.

Methods:

- **sha1(Object... vals)**: Generates a SHA-1 hash from inputs.
- **readContents(File file) / writeContents(File file, Object... contents)**: Reads/writes file contents.
- **readObject(File file, Class<T> expectedClass) / writeObject(File file, Serializable obj)**: Serializes/deserializes objects.
- **plainFilenamesIn(File dir)**: Lists files in a directory.
- **restrictedDelete(File file)**: Deletes a file if it exists in a Gitlet-controlled directory.

Workflow Description

Initialization

- **Command**: init
- **Behavior**: Creates a .gitlet directory and initializes metadata (e.g., HEAD, branches).

Adding Files

- **Command**: add [filename]
- **Behavior**: Stages a file for commit by saving its contents as a blob and recording its hash in the staging area.

Committing Changes

- **Command**: commit [message]

- **Behavior**: Creates a new commit object linking staged changes to the current commit.

Branch Management

- **Commands**: branch [branchName], rm-branch [branchName], switch [branchName]
- **Behavior**: Creates, deletes, or switches between branches.

Restoration

- **Command**: restore [fileName]
- **Behavior**: Restores a file to its state in a specified commit.