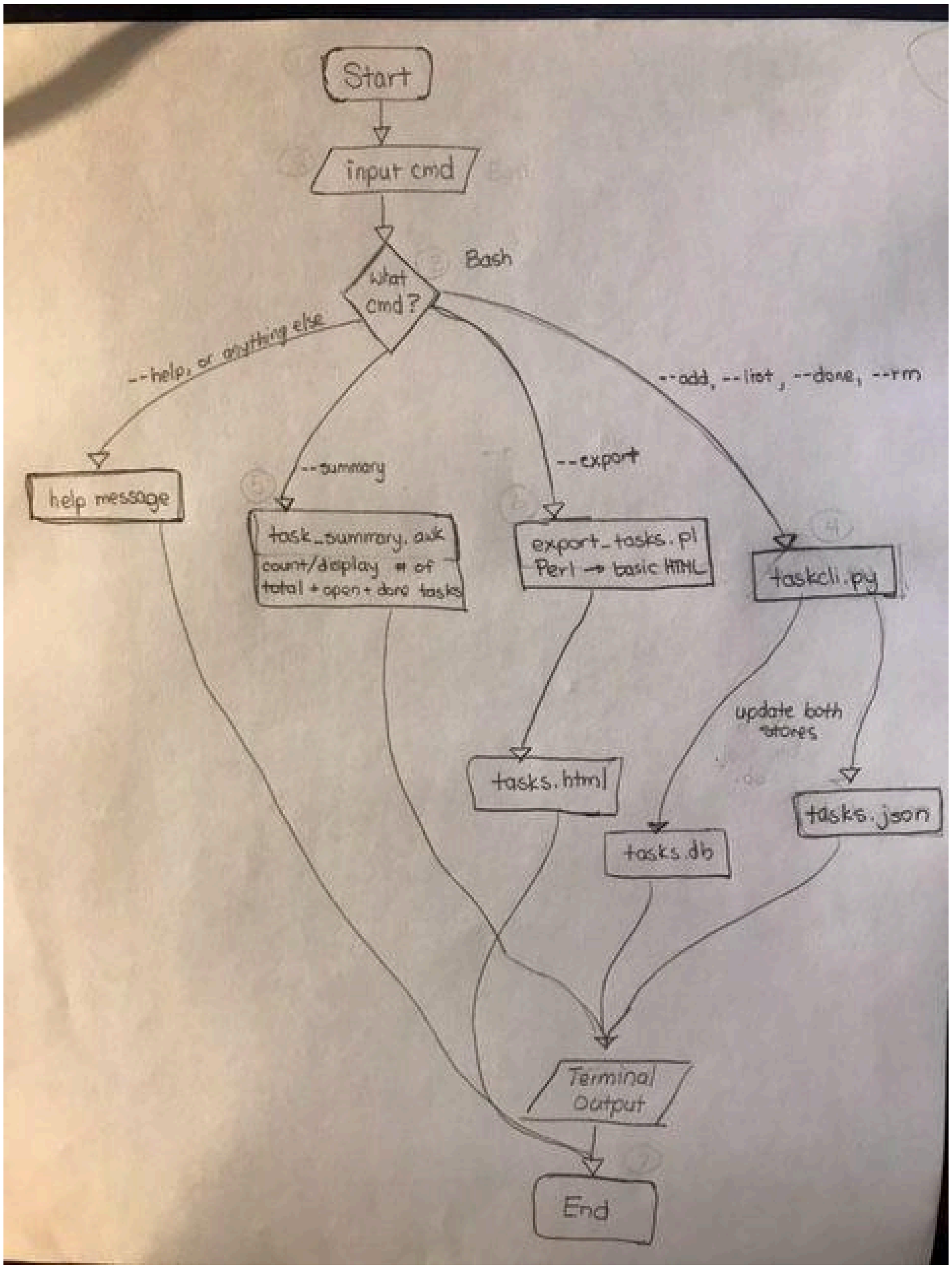
TaskCLI – Team members: Aleksei Ivanov, Darren Schell

Problem / Goal: Manage a to-do list from the command line

Flowchart



Algorithm

Step 1: Start the program and read the command-line arguments

Step 2: Determine which flag is passed:

- If --add "task", parse and save the task to a local storage file, and a record to the database. If it is the first task entered, initialize a new table for the data.

- If --list, read the file and display the tasks in formatted output

- If --done ID, locate the task with that ID and mark it as completed in both stores

- If --remove ID, locate the task with that ID and delete it from both stores

- If --help or any other command entered, show help & correct usage message

- If --export, pass .json file to Perl script and create a basic HTML file that shows the tasks

- If --summary, pipe .json file into AWK to create a basic summary of the tasks

Step 3: Perform the action (add, list, done, rm, etc)

Step 4: Save the updated task list and database (if modified)

Step 5: Display confirmation or results to the user

Step 6: Exit program

Pseudocode

# TaskCLI - Pseudocode

Bash main driver

```
IF (taskcli directory does not exist) then
        Create directory taskcli
        Move to directory taskcli

Command = the first command line argument
Arguments = the remaining arguments

Case command in --add | --list | --rm | --done)   call pythonCore(command, arguments)
                --export)                         call perlScript()
                --summary)                        call awkSummary()
                --help|*)                         display help message
```

```
End Case
****************  End Bash main driver  *************************************************


pythonCore(command, arguments)

IF file does not exist:
        CREATE file with empty task list []
ELSE
        Load taks from .json file

IF database does not exist:
        CREATE new database

Open sqlite3 database

IF (command == “--add”) then
        Determine the next available task_id
        CREATE task object with id, description, status=False
        APPEND task to tasks list
        SAVE tasks to file
        PRINT "Task added!"
        INSERT new row into database

 ELIF (command == “--list”) then
        FOR each task in tasks:
        PRINT id, status (✓ or ✗), description

 ELIF (command == “--done”) then
        FIND task with given id
        SET task["done"] = True
        SAVE tasks to file
        UPDATE corresponding (use id) database row
        PRINT "Task marked as complete!"

 ELIF (command == “--remove”) then
        REMOVE task with given id from list
        SAVE tasks to file
        DELETE corresponding (use id) database row
        PRINT "Task removed!"

 ELSE:
        PRINT help message or "No valid command"
```

```
Close File
Close database

*****************  END pythonCore() **********************************************************
// handles command “--export”
perlScript()
    Load tasks.json into a taskArray
    Create HTML table structure
    FOR each task in taskArray
        SET status = ‘✔’
        IF task.done == TRUE
            SET status = “✘”
        ADD a table row holding task_id, status, task description
    OUTPUT results in tasks.html

*****************  END perlScript() **********************************************************
// handles command “--summary”
awkSummary()
    Read .json file line by line
    taskCount = # of entries
    doneTasks = # of entries whose “done” flag is TRUE
    openTasks = taskCount - doneTasks
    Print the summary for the user: Total / Done / Open


*****************  END awkSummary() **********************************************************
```