EECS 448 – Software Engineering I Project 04 System Documentation

Group 08:

Huzaifa Zahid

Mir Shazil Faisal

Junyi Zhao

Zhenzhou Wang

Story Point Analysis of Previous EECS Projects

1	2	3	5	8	13
Lab 1 -Simple Printing Exercise	Lab 6-Array reversal, file I/O	Lab 3-Lidear Sensor	Lab 10-Intro to OOP	Lab 11- DMV Class	Lab 7 -Recursion& Backtracking
Lab 4 - Fibonacci Numbers, ASCII Conversion	Lab 1- Command-line argument file manipulation	Lab6 – Recursion exercises	Lab4 – Elevator(Queues&Stacks)	Lab9,Lab10- Binary Search Tree	EECS 448-project1
Homework Assignment 1	Lab8 – Time Complexities	Homework Assignment 2	Homework Assignment 3	Lab5-Browser Tracking (List)	EECS 448-Project2
	Homework Assignment 4	Lab 6-Unified Modeling Language		EECS 448-Project3	Homework Assignment 5
	Homework Assignment 5	Homework Assignment 6	Lab 7-Software Testing	Lab 8-Web Language Intro	EECS 448-Project4

Legend:

	EECS 168
	EECS 268
	EECS 368
	EECS 388
	EECS 448

Estimate of Hours/Story Point:

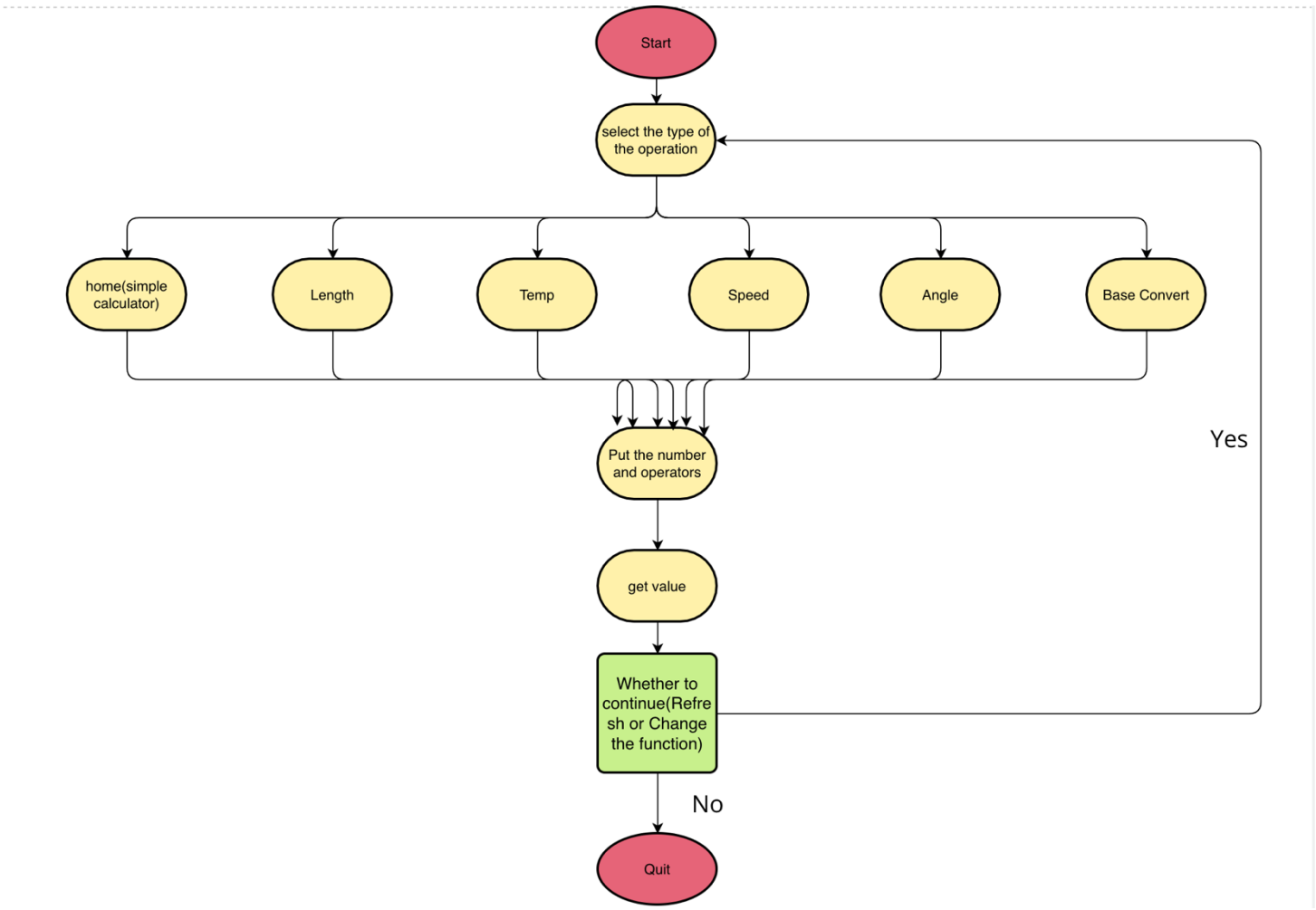
1 hour/story point.

We placed the project4 under 13 story points because we thought although project 4 is an upgrade from project 3 and adds some functions. But the added functions are much more complicated than project3, including some algorithms.

Introduction:

Our project 4 is a Scientific calculator program implemented in JavaScript, HTML and CSS. In project3, we create a simple calculator performs addition, subtraction, multiplication, and division. In the project 4, we create a scientific calculator based on the project 3 that can calculate slightly more complex trigonometric and calculus equations, this includes the conversion of trigonometric functions, as well as the conversion of international units such as length and temperature and speed and angle.

Project 4 Scientific Calculator Flow Chart



defect tracking tool

Date reported	Apr 23rd	Apr 23rd
Who reported it	Junyi Zhao	Mir, Shazil Faisal
Brief description of defect	sin cos and tan function will throw out the error if no brackets are added	When trigonometric functions add something, it will throw out error.
Date fixed	Apr 23rd	
Who fixed it		
Description of how it was fixed	The defects do not need to be fixed, just add the brackets for trigonometric functions	

Code Integrated:

In Project 4, we used JavaScript, HTML and CSS, where the HTML files is responsible for the layout, the CSS files for the interface design and the JavaScript files for the logic, and modularity is evident in our project, especially in the corresponding .JS files for the different functions. For example, Home, length, temperature, speed, angle, and base convert are seven modules. So, the team division of labor can become very clear. Before integration, you just need to design different modules and give them to different people, and then put the different code together for testing.
The Integration Strategy we used in the Project 4 should be the Sandwich Integration. The Sandwich Integration has several features. It is combines two strategies of topdown and bottom-up which maximizing their strengths and minimizing their weaknesses. The sandwich integration artition modules into two groups (basically layout and logical). Implement and integrate logic artifacts top down(in the JavaScript files, which is the decision making flow and generally situated close to root) and Implement and integrate operational artifacts bottom up (in the HTML files,which is Performing actual operations and Generally found in lower levels). Last but not the least, test interfaces between logic artifacts and operational artifacts (both HTML and JavaScript files).
Our Project 4 is designed to meet these features from design to distribution to testing. And the reason we do the Sandwich Integration because it has three strengths, which is more easier fault isolation, major design faults show up early and the operational artifacts are adequately tested.

Deployment Plan:

- What are the steps required to deploy your project?
- Who is the potential market?
- What will it cost to deploy it?
- Example costs:
    - app store costs
    - costs to get your game on XBoxLive
    - costs to print disks for distribution
    - costs to buy domains
    - costs to attend conventions and conference and set up booths

Maintenance Plan:

- How much it will cost to maintain your product for the next year?
- Example costs:
    - Costs for hiring developers
    - Monthly or annual fees for servers or domain names
    - Monthly or annual fees for your distribution platform (e.g. app store, XBox live)

Code Review:

- Undertake, as part of a team activity, an inspection of a medium-size code segment using either a Walkthrough or Inspection as described in the lecture on Software Verification
- Document the following:
    - List of faults detected
    - Description of each fault
    - Who was assigned to fix each fault
    - How the fault was fixed