COSC310 Assignment #2

Project Group #8

Liam Harbec 66453853
Michael Ogden 89534952
Yidu Guo 45006160
Riley Clark 45546512
Tim Mammadov 62199559

**All detailed documentations at full size can also be found under the "documentations" folder in Github Repository (https://github.com/COSC310-Project-Group-8/AssignmentTwo), figures included in this document are merely meant as an overview.

Presentation Link (https://www.youtube.com/watch?v=qOvMm_9t9D4)

Project Description

Our goal is to make a Gym Helpline agent. A user may have a variety of questions about a gym and we hope the agent may act as a supplement to a "Frequently Asked Questions" page. Information provided by the agent can be one of several simple topics, such as, available trainers, equipment provided, opening and closing hours and other services provided by the gym. This will allow the gym to provide answers to potential clients without having one or more human agents on call at times.

Repository

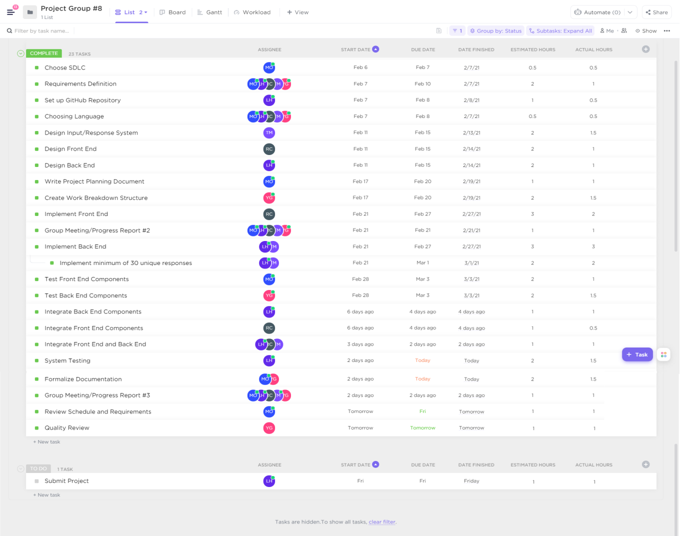
Software Development Cycle

We have decided to use the plan-driven Waterfall model for this project. If we had more time to build a more advanced chat-bot by introducing new features, then we would most likely use an agile strategy focusing around scrums. Since we are desiring to make the most complete version we can in the time allotted as well as the fact that we have a good idea of what our chat bot's functionality will be before beginning development, we decided having a thorough plan and schedule ahead of development would be more productive and ensure we have the best final product.

Software Development Cycle Phases

1. Requirements Definition:
    a. Deciding context of use
    b. Deciding on areas of response
    c. Defining general interface
2. Software and System Design:
    a. Designing Back End
        i. Dialogue Structure
        ii. Input and Response
        iii. (At least) 30 Responses
    b. Designing Front End
        i. Designing interface for input and output
        ii. Designing triggers for response/responses
3. Implementation and Unit Testing:
    a. Implement Front End
    b. Implement Back End
    c. Testing of backend components
    d. Testing of library of inputs/outputs
4. Integration and System Testing:
    a. Integrate backend components
    b. Integrate Front and Back end
    c. Test System
5. Operation and Maintenance
    a. Review Schedule and Requirements
    b. Quality Review
    c. Post-Review Activities
        i. (If necessary/if time allows) Design and Implement missing functions
        ii. Otherwise, take lessons and finish.

Work Breakdown Structure



Project Group #8
1 List

List 2 · Board · Gantt · Workload · + View · Automate (0) · Share

Filter by task name... · 1 · Group by: Status · Subtasks: Expand All · Me · Show

COMPLETE 23 TASKS

	ASSIGNEE	START DATE	DUE DATE	DATE FINISHED	ESTIMATED HOURS	ACTUAL HOURS
Choose SDLC	MO	Feb 6	Feb 7	2/7/21	0.5	0.5
Requirements Definition	MO LH RC TM YG	Feb 7	Feb 10	2/7/21	2	1
Set up GitHub Repository	LH	Feb 7	Feb 8	2/8/21	1	0.5
Choosing Language	MO LH RC TM YG	Feb 7	Feb 8	2/7/21	0.5	0.5
Design Input/Response System	TM	Feb 11	Feb 15	2/13/21	2	1.5
Design Front End	RC	Feb 11	Feb 15	2/14/21	2	1
Design Back End	LH	Feb 11	Feb 15	2/14/21	2	1
Write Project Planning Document	MO	Feb 17	Feb 20	2/19/21	1	1
Create Work Breakdown Structure	YG	Feb 17	Feb 20	2/19/21	2	1.5
Implement Front End	RC	Feb 21	Feb 27	2/27/21	3	2
Group Meeting/Progress Report #2	MO LH RC TM YG	Feb 21	Feb 21	2/21/21	1	1
Implement Back End	LH TM	Feb 21	Feb 27	2/27/21	3	3
Implement minimum of 30 unique responses	LH TM	Feb 21	Mar 1	3/1/21	2	2
Test Front End Components	MO	Feb 28	Mar 3	3/3/21	2	1
Test Back End Components	YG	Feb 28	Mar 3	3/3/21	2	1.5
Integrate Back End Components	LH	6 days ago	4 days ago	4 days ago	1	1
Integrate Front End Components	RC	6 days ago	4 days ago	4 days ago	1	0.5
Integrate Front End and Back End	LH RC TM	3 days ago	2 days ago	2 days ago	1	1
System Testing	LH	2 days ago	Today	Today	2	1.5
Formalize Documentation	MO YG	2 days ago	Today	Today	2	1.5
Group Meeting/Progress Report #3	MO LH RC TM YG	2 days ago	2 days ago	2 days ago	1	1
Review Schedule and Requirements	MO	Tomorrow	Fri	Tomorrow	1	1
Quality Review	YG	Tomorrow	Tomorrow	Tomorrow	1	1

+ New task

TO DO 1 TASK

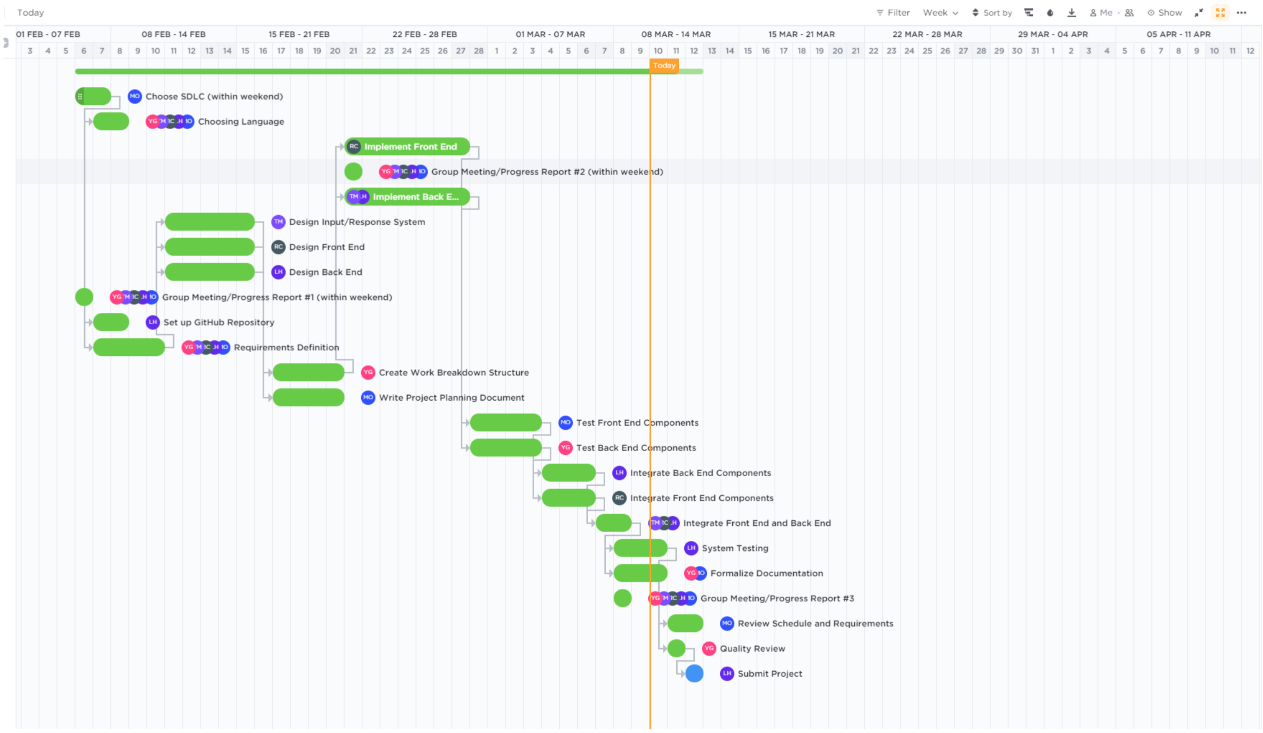
	ASSIGNEE	START DATE	DUE DATE	DATE FINISHED	ESTIMATED HOURS	ACTUAL HOURS
Submit Project	LH	Fri	Fri	Friday	1	1

+ New task

+ Task

Tasks are hidden.To show all tasks, clear filter.

Gantt Chart



Gantt Chart and WBS Source

Role Assignments

(Refer to Gantt Chart and WBS Source)

Hour Estimation and Actuals

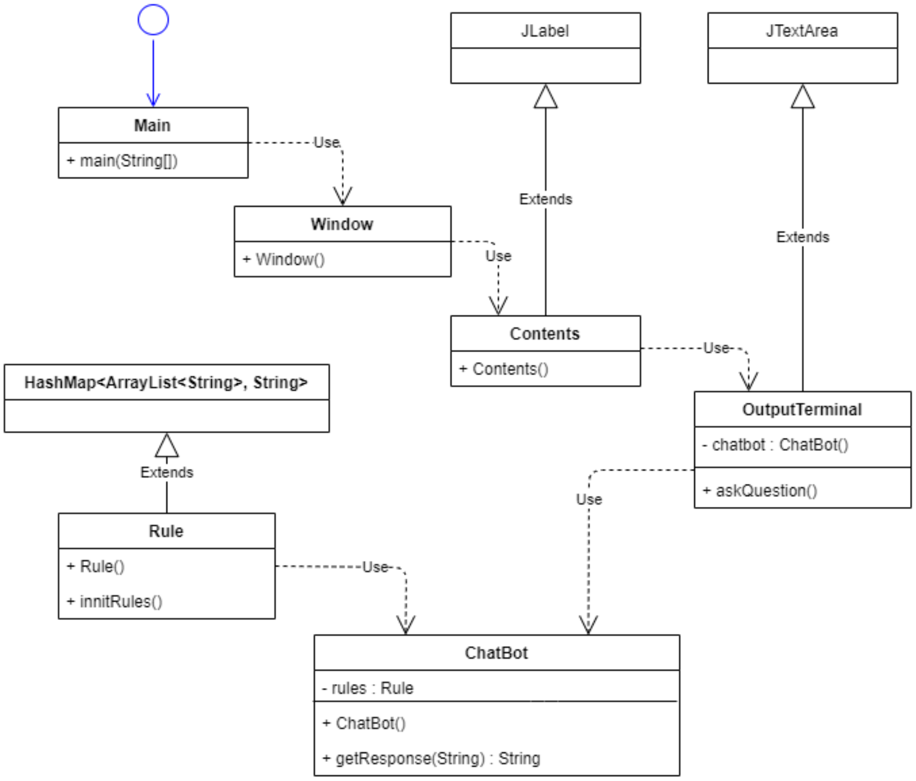
(Refer to Gantt Chart)

Project Meetings

(Refer to Gantt Chart**)**

System Documentations

Class Hierarchy



Class Description

In general, our system can be divided into a UI component (Window, Contents, OutputTerminal) and core ChatBot functionality (ChatBot, Rule).

Main class:

- Our entry point, containing void main(String[] args) method. It creates an instance of a Window class with dimensions 800x800.

Window class:

- Window class sets up a JFrame object to keep all our UI components.
- It also instantiates the Contents class and adds it to frame.

Contents class:

- Sets up everything displayed on the screen, including input/output windows, button, scrollbar, etc.
- Extends JLabel to perform Swing library functionality.

OutputTerminal class:

- Makes up the output window containing all “human” questions and “bot” responses.
- Uses ChatBot to generate the responses.

ChatBot class:

- Generates responses according to keywords registered in rules.
- Matches the user input with a pattern of common keywords.
- Uses a default response if no keywords found.

Rule class:

- A hash map that is used by chatbot to access all preset keywords
- It preloads the keywords from a file “word.txt”
- Each list of keywords (key) corresponds to one response (value)

System Limitations

1. The chatbot only have fixed answers for each topic, different question containing different keyword of the same topic is likely to result in the same response from the chatbox
2. Enter button activatible only by clicking it, some user may attempt in using “enter” key on keyboard
3. No learning function, chatbox can only respond to pre-determined keywords, and can only respond with fixed responses. It cannot automatically detect and add additional keywords and responses

4. Chatbot often will prompt users to visit other external resources such as the gym website, however chatbot does not contain a function to automatically redirect users to desired external resources.
5. Chatbot does not keep records of individual user’s questions, returning users will not be able to see records of previous chatbot conversation.