# Date-bot Project Report

Eric Scherban, Zachary Maludzinski, James Boonstra, Pardeep Rathore, Swakhar Poddar

# Table of Contents

1 Introduction

2 SDLC

  2.1 Rational

  2.2 Overview

3 Graphs & Analytics

  3.1 WBS Graph

  3.2 WBS Sheet

  3.3 Gantt Chart

  3.4 Sizing

4 Product Limitations

5 Sample Output

6 Conclusions and Reflections

# 1 Introduction

For Assignment 2 our team created a program which replicates the dialogue between individuals on a first date. As such, the agent has certain characteristics which alter the nature of its response based on user input. For example, the agent will be able to distinguish between positive and negative statements and respond in kind. The project will start by relying on the console for input and output; however, upon completing the more technical aspects of the project (and if time permits), a simple GUI will be implemented for the program. Our team will rely on the Waterfall SDLC as it is straightforward and suited for a project with a relatively long timeline.

Github repository: https://github.com/Zmalski/310Project

# 2 SDLC

## 2.1 Rationale

We chose the Waterfall model as our SDLC. Waterfall worked really well for this project as it was straightforward, therefore suitable for a team whose members are relatively new to planning projects by using a specific framework. Waterfall was very beneficial because we had a month to complete the task with only five group members, so one step was required to be completed before we could move on to the next one, allowing us to easily coordinate our time and resources. However, this was also a challenge when a group member fell behind and it prohibited the rest of the group from proceeding.

## 2.2 Overview

1. Concept Exploration
    a. Choose Product
        i. Determine characteristics
        ii. Determine theme
    b. Determine Functions
        i. User interaction with Bot
        ii. Determine product features
    c. Project Vision
        i. Determine learning objectives
        ii. Desired audience.
2. Requirements
    a. Choose libraries
        i. Analyze libraries and their functionality
    b. Create data structure of common phrases and responses
        i. Determine ideal data structure
    c. Determine scope of suitable input

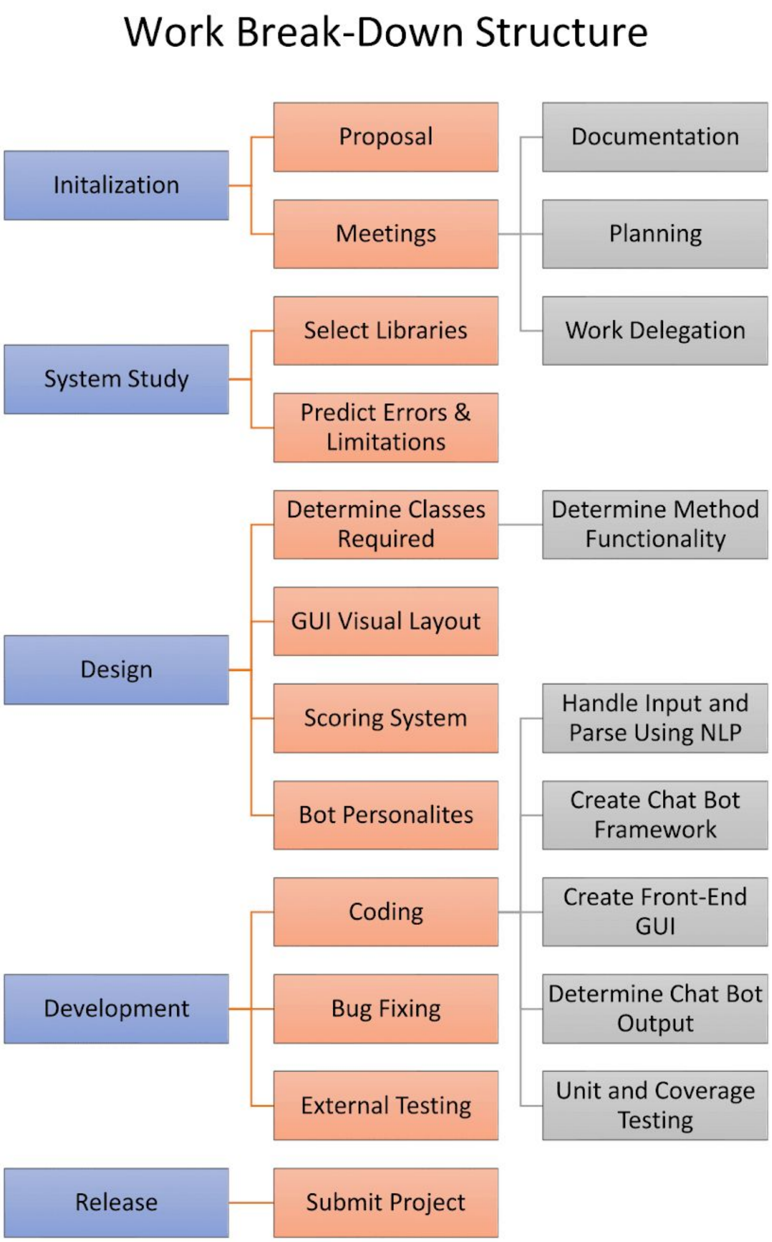
3. Design
    a. Determine classes and objects and how they interact
        i. Assign classes to group members
    b. Determine appearance and feel of GUI
        i. Choose colour scheme
        ii. Select either JavaFX or Swing
    c. Determine specifics of Bot responses

        i. Attitude
        ii. Interests
        iii. Scoring mechanism

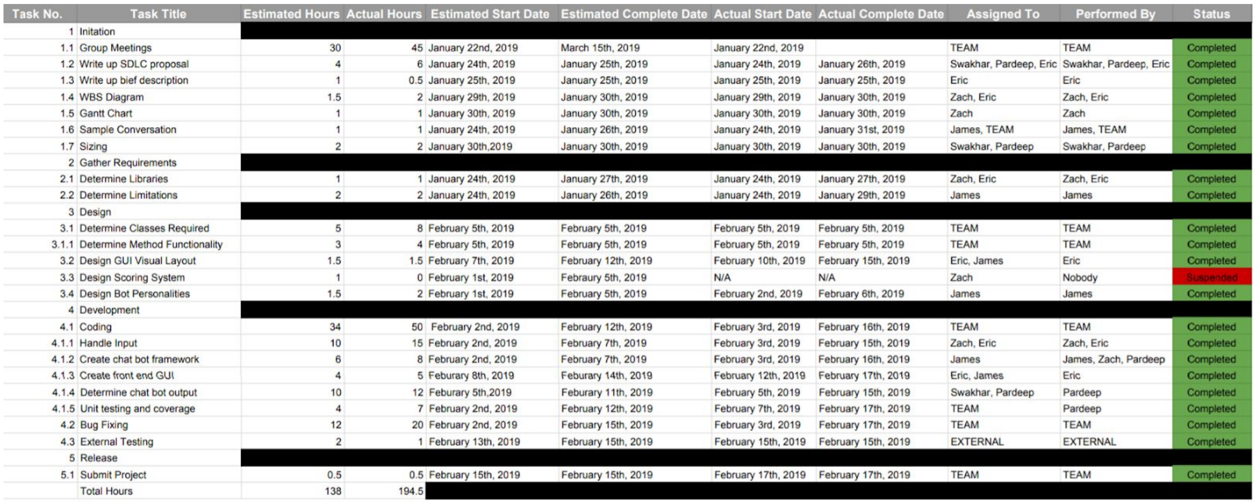
4. Implementation:
    a. Coding
        i. Create an agent which can hold a realistic conversation with a user for at least 30 “exchanges”
        ii. Design the agent so that it is capable of identifying various questions and responses from the user
        iii. Implement scoring algorithm to rate user’s date performance
        iv. Determine whether or not the user is successful on the date using scoring algorithm
        v. Frequent commenting and documentation of methods
    b. (Optional) GUI coding using visual programming techniques
    c. Synchronize code using a coding repository
        i. Frequent collaboration
5. Operations & Support
    a. Finalize documentation
        i. Create walkthrough for usage of software
    b. Consult with user to ensure functioning product
        i. Provide any required technical assistance
    c. Log reported errors
6. Maintenance
    a. Testing
        i. Unit/Coverage Testing
        ii. Input validation
        iii. External testing
    b. Debugging
        i. Update code and documentation
    c. Continue internal troubleshooting
7. Retirement
    a. Submit assignment

# 3 Graphs and Analytics

WBS Graph

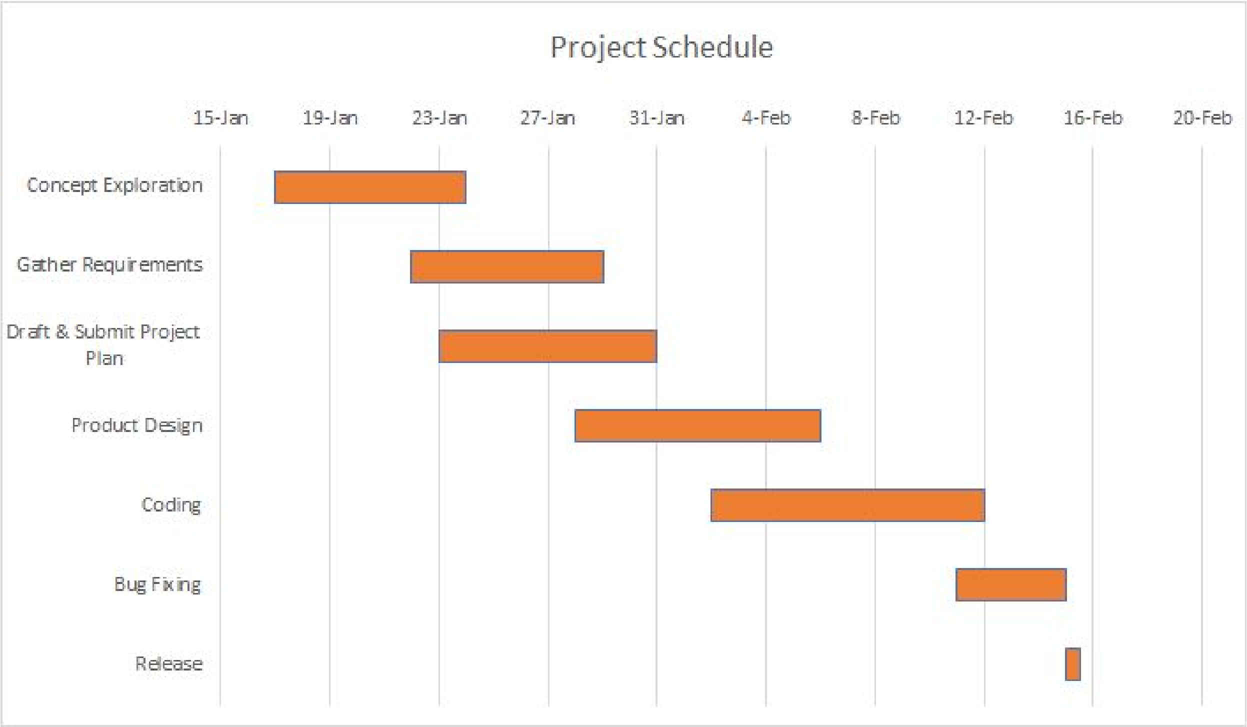


## WBS Sheet – Includes Role Assignments & Hours



| Task No. | Task Title | Estimated Hours | Actual Hours | Estimated Start Date | Estimated Complete Date | Actual Start Date | Actual Complete Date | Assigned To | Performed By | Status |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Initation | | | | | | | | | |
| 1.1 | Group Meetings | 30 | 45 | January 22nd, 2019 | March 15th, 2019 | January 22nd, 2019 | | TEAM | TEAM | Completed |
| 1.2 | Write up SDLC proposal | 4 | 6 | January 24th, 2019 | January 25th, 2019 | January 24th, 2019 | January 26th, 2019 | Swakhar, Pardeep, Eric | Swakhar, Pardeep, Eric | Completed |
| 1.3 | Write up bief description | 1 | 0.5 | January 25th, 2019 | January 25th, 2019 | January 25th, 2019 | January 25th, 2019 | Eric | Eric | Completed |
| 1.4 | WBS Diagram | 1.5 | 2 | January 29th, 2019 | January 30th, 2019 | January 29th, 2019 | January 30th, 2019 | Zach, Eric | Zach, Eric | Completed |
| 1.5 | Gantt Chart | 1 | 1 | January 30th, 2019 | January 30th, 2019 | January 30th, 2019 | January 30th, 2019 | Zach | Zach | Completed |
| 1.6 | Sample Conversation | 1 | 1 | January 24th, 2019 | January 26th, 2019 | January 24th, 2019 | January 31st, 2019 | James, TEAM | James, TEAM | Completed |
| 1.7 | Sizing | 2 | 2 | January 30th,2019 | January 30th, 2019 | January 30th, 2019 | January 30th, 2019 | Swakhar, Pardeep | Swakhar, Pardeep | Completed |
| 2 | Gather Requirements | | | | | | | | | |
| 2.1 | Determine Libraries | 1 | 1 | January 24th, 2019 | January 27th, 2019 | January 24th, 2019 | January 27th, 2019 | Zach, Eric | Zach, Eric | Completed |
| 2.2 | Determine Limitations | 2 | 2 | January 24th, 2019 | January 26th, 2019 | January 24th, 2019 | January 29th, 2019 | James | James | Completed |
| 3 | Design | | | | | | | | | |
| 3.1 | Determine Classes Required | 5 | 8 | February 5th, 2019 | February 5th, 2019 | February 5th, 2019 | February 5th, 2019 | TEAM | TEAM | Completed |
| 3.1.1 | Determine Method Functionality | 3 | 4 | February 5th, 2019 | February 5th, 2019 | February 5th, 2019 | February 5th, 2019 | TEAM | TEAM | Completed |
| 3.2 | Design GUI Visual Layout | 1.5 | 1.5 | February 7th, 2019 | February 12th, 2019 | February 10th, 2019 | February 15th, 2019 | Eric, James | Eric | Completed |
| 3.3 | Design Scoring System | 1 | 0 | February 1st, 2019 | Febraury 5th, 2019 | N/A | N/A | Zach | Nobody | Suspended |
| 3.4 | Design Bot Personalities | 1.5 | 2 | February 1st, 2019 | February 5th, 2019 | February 2nd, 2019 | February 6th, 2019 | James | James | Completed |
| 4 | Development | | | | | | | | | |
| 4.1 | Coding | 34 | 50 | February 2nd, 2019 | February 12th, 2019 | February 3rd, 2019 | February 16th, 2019 | TEAM | TEAM | Completed |
| 4.1.1 | Handle Input | 10 | 15 | February 2nd, 2019 | February 7th, 2019 | February 3rd, 2019 | February 15th, 2019 | Zach, Eric | Zach, Eric | Completed |
| 4.1.2 | Create chat bot framework | 6 | 8 | February 2nd, 2019 | February 7th, 2019 | February 3rd, 2019 | February 16th, 2019 | James | James, Zach, Pardeep | Completed |
| 4.1.3 | Create front end GUI | 4 | 5 | Feburary 8th, 2019 | Feburary 14th, 2019 | February 12th, 2019 | February 17th, 2019 | Eric, James | Eric | Completed |
| 4.1.4 | Determine chat bot output | 10 | 12 | Feburary 5th,2019 | Feburary 11th, 2019 | February 5th, 2019 | February 15th, 2019 | Swakhar, Pardeep | Pardeep | Completed |
| 4.1.5 | Unit testing and coverage | 4 | 7 | February 2nd, 2019 | February 12th, 2019 | February 7th, 2019 | February 17th, 2019 | TEAM | Pardeep | Completed |
| 4.2 | Bug Fixing | 12 | 20 | February 2nd, 2019 | February 15th, 2019 | February 3rd, 2019 | February 17th, 2019 | TEAM | TEAM | Completed |
| 4.3 | External Testing | 2 | 1 | February 13th, 2019 | February 15th, 2019 | February 15th, 2019 | February 15th, 2019 | EXTERNAL | EXTERNAL | Completed |
| 5 | Release | | | | | | | | | |
| 5.1 | Submit Project | 0.5 | 0.5 | February 15th, 2019 | February 15th, 2019 | February 17th, 2019 | February 17th, 2019 | TEAM | TEAM | Completed |
| | Total Hours | 138 | 194.5 | | | | | | | |

## Gantt Chart



Project Sizing

As per the sizing calculation, our expected lines of code for the project was 1688 to 2745 lines. Upon completion, our chatbot contains about 1800 lines of code. As such, our estimation was consistent with the actual lines of code required to complete the project.

# 4 Product Limitations

Limitations

- Can’t handle spelling errors
- Has trouble with synonyms
- Limited Vocabulary
- Limited Interests/Things to talk about
- Queries need to be entered with keywords
- Responses are limited.
- Very base layer on a wide span of interests.
- Unable to “dive” into any interests

These are all things that can be improved upon in Assignment 3.

# 5 Sample Output

The following is an example of an ideal conversation with our dating bot. Of course, conversations will vary depending on the user.

1 . **Chatbot:** You are on a blind date. Would you like to date a man or a woman.

User: woman

**Chatbot:** You are now on a date with a girl named Kiera

2. **Chatbot**: Hi! I'm Kiera, what's your name?

User: My name is James

3. **Chatbot**: It's nice to meet you James. How are you?

User: I'm fantastic

4. **Chatbot**: That's good to hear.

User: How are you?

5. **Chatbot**: I'm doing well, thanks.

User: What do you do for work?

6. **Chatbot**: I am a chef

User: That's cool

7. **Chatbot**: What do you do for a living?

User: I am an astronaut

8. **Chatbot**: An astronaut? That's cool!

User: Thanks

9. **Chatbot**: Seen any good movies lately?

User: I like toy story and the road to el dorado

10. **Chatbot**: Oh I haven't heard of those movies

User: What movies do you like?

11. **Chatbot**: I like cars

User: That's a good movie

12. **Chatbot**: What genres of music do you listen to?

User: I listen to rap, reggae, hip hop, and techno

13. **Chatbot**: I don't like rap sorry.

User: What music do you like?

14. **Chatbot**: I like instrumental, classical, and jazz

User: I love Frank Sinatra.

15. **Chatbot**: What countries do you like?

User: I like peru, japan, and china

16. **Chatbot**: I like china too!

User: That's cool

17. **Chatbot**: What are your hobbies

User: I like winemaking, beekeeping, and building canoes

18: **Chatbot**: I like beekeeping too!

User: What are your hobbies?

19: **Chatbot**: I like beekeeping, auto repair, and magic tricks

User: That's neat

20. **Chatbot**: What are your favourite sports?

User: I like hockey, curling, and skating

21. **Chatbot**: I like hockey too! I don't like curling sorry

User: It's a fun sport. What sports do you like?

22. **Chatbot**: I like hockey, surfing, and wakeboarding

User: How old are you?.

23. **Chatbot**: I am 22 years old

User: I am 23 years old

24. **Chatbot**: Lets play rock paper scissors! Pick rock, paper, or scissors.

User: rock

25. **Chatbot**: I picked rock. We tied, that was so fun!

User: F*** you

26. **Chatbot:** Watch your language

User: What the hell! I wanted to win

27. **Chatbot**: You kiss your mother with that mouth?

User: You're a fool!

28. **Chatbot:** Don't call me names!

User: I'm sorry

29. **Chatbot:** I'm sorry I'm out of things to talk about! This has been fun.

User: Goodbye

30. **Chatbot**: Bye!

# 6 Conclusions and Reflections

We originally planned to implement an XML based markup language known as AIML. However, we decided against implementing it as it was determined to have too great of a learning curve given the time frame. Since we decided on the waterfall as our SDLC, we were forced to return to the system requirement phase and rethink our options. We ended up going with a manual approach towards handling user input and determining our bots output. While this ended up being much less flexible than an AIML based chatbot, it fits well within our group's experience levels. We do recognize that the upfront learning curve may have saved us time in the long run, however.

While Waterfall was well suited for our project, it did require a lot of backtracking if any changes at a later point are needed to be made. As such, we would consider using a different development life cycle in the future, using this experience we have gained to weigh our options accordingly.