**Description of the Project:**

Develop an interactive conversational travel agent that responds to user questions using Java. The user can ask the chatbot any question that is related to the trip such as weather of the destination, cost of the vacation, and transportation for the trip, etc.

Repository URL: <https://github.com/allantsai123/COSC310project>

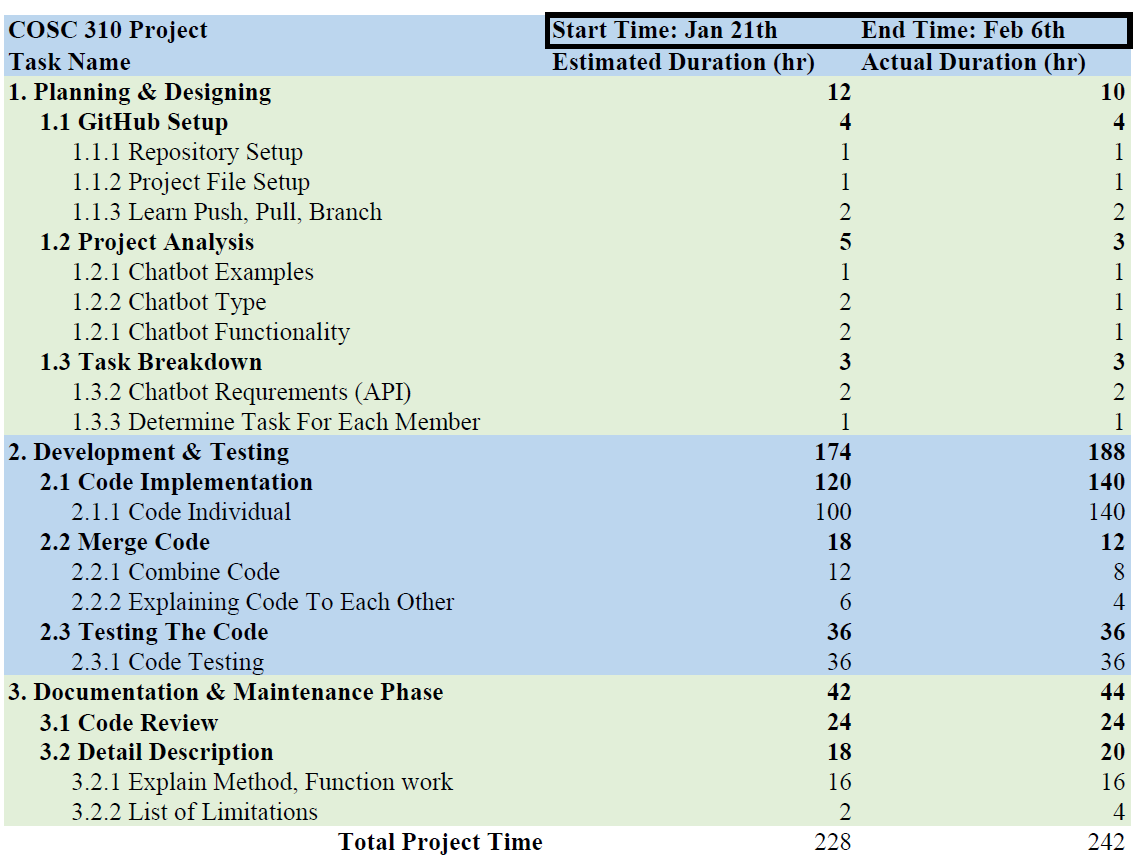
**SDLC: Scrum combined with Incremental**

We choose scrum combine with incremental because we can separate the whole project into small tasks (divide and conquer approach), and each member can work on one part of the project. We also want to prioritize the tasks so we know which functions need to be implemented as soon as possible. Since there will be lots of rapidly changing during the design and development phase, we want to have lots of testing and meetings so we can exchange ideas, update the status and address the issues.

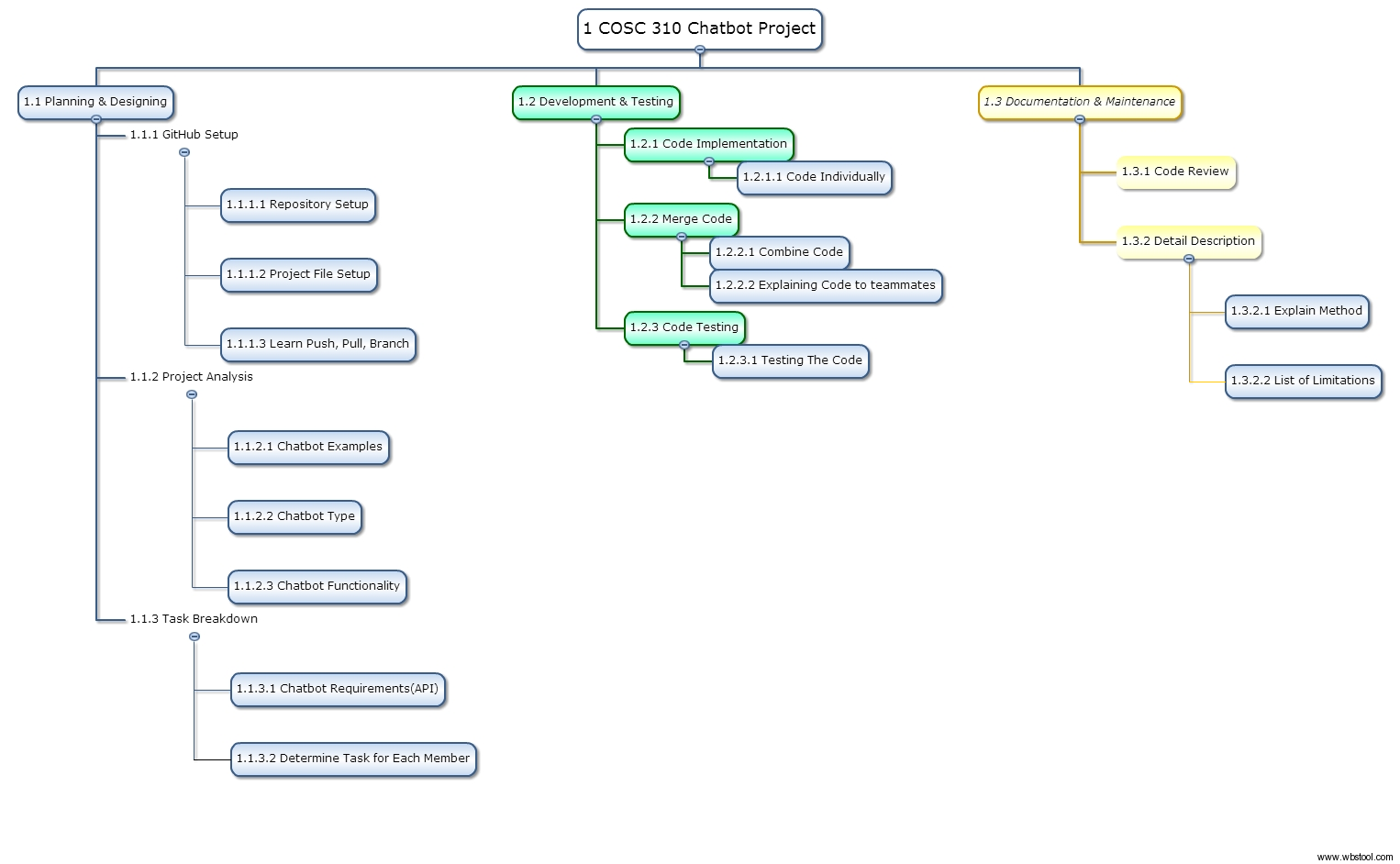
**SDLC Phases:**

1. Planning and Design Phase
   1. Set up the GitHub
      1. Repository and project file set up
      2. Learn push, pull, branch
   2. To know and learn what the project is and how to implement it.
      1. Find some example of Chabot.
      2. Figure out what type of Chabot we are making.
   3. Discuss how to separate the project into different tasks
      1. Functionalities/Sizing
      2. DB? API?
      3. Determine who is doing which task.
2. Development and Testing Phase
   1. Code implementation
      1. Each member code on their own time.
   2. Merging the code
      1. Combine the code from each member
      2. Understanding the code from other
   3. Testing the code
      1. Unit tests
      2. Fixing the code
3. Documentation and Maintenance Phase
   1. Document how the project work in a detail way
      1. Explaining how each method, function, or class works.
      2. List of limitations
   2. More functions?
      1. Discuss the time left and implement more features?

**WBS in Table Form:**



**WBS in Tree Graph:**

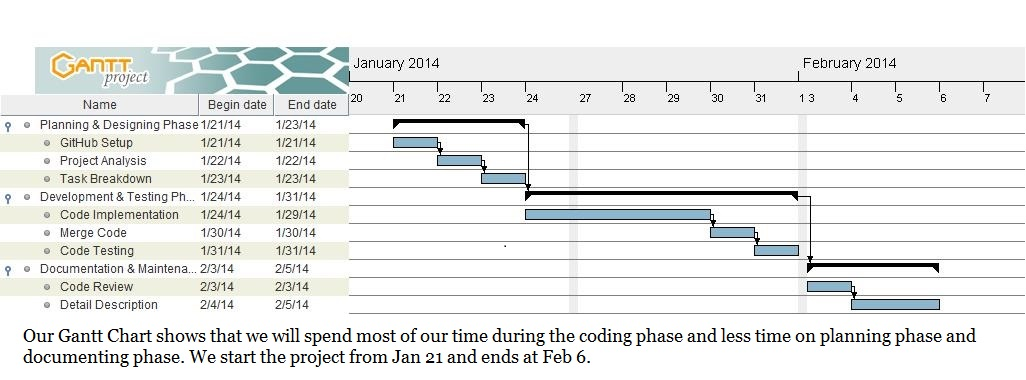


We choose Democratic Team Approach for team organization, where everyone takes a part of the project and work on their own. We use this strategy because each teammate has the same role and it is much more productive than other approaches. This method also let us practices egoless programming, where we are less likely to criticize each other (make decisions by consensus) and more motivation to find errors which improve our project.

In Planning and Designing phase, we estimate 12 hours of working time. However we spent 10 hours on this phase, which is pretty close to our estimated time.

In phase 2, Development and Testing, our WBS chart shows that we spent most of the time in this phase, and we end up have to spend more time on code implementation. Our actual time on merging the code is less than the estimate time is because we utilized the pull and push function in GitHub.

In Documentation and Maintenance phase, our actual time is very close to the estimated time. And we spend a little more time on the documenting the methods, and functions.

**Gantt Chart:**

**Actual Timeline:**

**Project Starts:** Jan 21th

**Project Ends:** Feb 7th

**Project starts:**

**Phase 1:**

Jan 21th: Discuss on how to deal with assignment 2 after class

Jan 22th: Set up the GitHub repository and add project folder

Jan 25th: Research on the topic

Jan 27th: Meet at the library; discuss the topic and what functions should be implemented. (basic structure of the code)

**Phase 2:**

Jan 28-31th: Implement the code individually.

Jan 29th: Meet at the library; discuss and revise the code

Jan 31-Feb 3rd: finish up the coding part of the project.

**Phase 3:**

Feb 3rd: Meet at the library; discuss what is done and what is left.

Feb 4th: Documentation.

Feb 5-7th: Meet at the library; finishing up the project. (Presentation)

**Limitations of the Program:**

1. It can only handle conversations for traveling guild, weather, transportation, distances between two cities, etc.
2. The chatbot can only understand certain words (knowledge limitations), sentences (words) need to be very specific. Destinations have to be specific. There can be many places having the same name.
3. Does not answer the question “why”.
4. Only work in English.
5. …

**Future Implementations:**

Because of the time constraint and the lack of knowledge we got, blah blah blah